Week 5 Final Assignment—E-Commerce Website Test Plan

Shaun Hoadley

CST313: Software Testing

Michael Hayden

07/05/2021

Software Requirements Specification

for

E-Commerce Website Test Plan

Version 1.0 Release 1.0 Waiting for Approval

Prepared by Shaun Hoadley

CSC313: Software Testing

07/05/2021

Table of Contents

Table of Contents iii

Revision History iii

1. Introduction 4

1.1 Purpose **4**

1.2 Document Conventions **4**

1.3 Intended Audience and Reading Suggestions **4**

1.4 Product Scope **4**

2. Overall Description 4

2.1 Product Perspective 4

2.2 Product Functions 4

2.3 User Classes and Characteristics 5

2.4 Operating Environment 5

2.5 Design and Implementation Constraints 5

2.6 User Documentation 5

2.7 Assumptions and Dependencies 5

3. External Interface Requirements 6

3.1 User Interfaces 6

3.2 Hardware Interfaces 6

3.3 Software Interfaces 6

3.4 Communications Interfaces 6

4. System Features 6

4.1 New User Account 6

4.2 Existing Account Login 7

4.3 User Searches Items 7

4.4 User Views Item Details 8

4.5 User Add Item to Cart 8

4.6 User Cart View 9

4.7 User Can Checkout and Finalize 10

5. Other Nonfunctional Requirements 10

5.1 Performance Requirements 10

5.2 Safety Requirements 10

5.3 Security Requirements 11

5.4 Software Quality Attributes 11

5.5 Business Rules 11

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this SRS (software requirements specification) is to outline the requirements of software build version 1.0, document release 0.6, of the e-commerce website. The e-commerce website has the expressed purpose of allowing customers to quickly locate and purchase items from a range of options. Release 0.6 of this document refers to the e-commerce website version 1.0.

## Document Conventions

This document will be broken into 5 sections. These sections will have headers for Introduction, Overall Description, External Interface Requirements, System Features, and Other Non-Functional Requirements. Sub-headers will provide the details of the specifics for each of those sections.

## Intended Audience and Reading Suggestions

This SRS has an intended audience of software developers, architects, designers, project managers, software testers, and any other stakeholders involved this project and its execution. To gain full understanding for the purpose and requirements for the e-commerce website, it is suggested that all audience members read the document read the document in its entirety.

## Product Scope

The e-commerce website allows users to find the products they want quickly and easily. The user will be able to search for an item, select from a list of search suggestions, narrow the search by selecting different filters items, view the details of the items, add items to a cart, continue shopping for more items, and checkout when they are ready. The development of this e-commerce website should increase growth and profitability, which is the objective of the business.

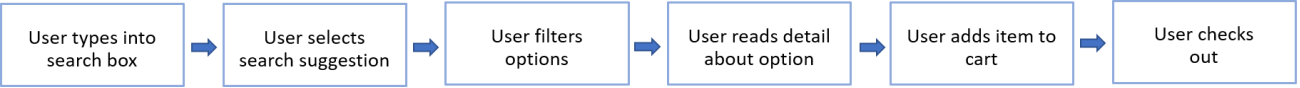
# Overall Description

## Product Perspective

This E-commerce website will be a new build and a self-contained project.

## Product Functions

Users should be able to enter a search for items, predictive typing should present suggested groups or item types, use filters to narrow the search, view the details of resulting items, add items to the cart, continue to shop for other items, and check out when they have completed shopping. Different item types will have different filters available that represent aspects of their item type (grouping items of similar type).



## User Classes and Characteristics

For the E-commerce website, the user classes consist of customers, vendors, and administrators. Customer class users will need to be able to search, select, and purchase items. Vendor class users will need to be able to add, remove, and update their inventory lists. In addition to all previous functions above, administrator class users will also need to be able to assist the other user classes with login issues, retrieve confirmed orders by an order ID number, and manage site maintenance.

## Operating Environment

In addition to cross-platform compatibility, the E-commerce website should exhibit mobile responsiveness. The E-commerce website should function properly on Windows, Mac, Linux, Chromium, IOS, and Android operating systems, and work in all of the major browsers (Edge, Chrome, Firefox, Safari, etc.).

## Design and Implementation Constraints

The project timeframe deadline for the E-commerce website is six months. The website is to be designed to work with relational databases and the user account sensitive data must be encrypted. As the site is to be maintained by the client, it’s design and build must exhibit modifiability, maintainability, and testability support, and comply with ADA accessibility guidelines (ADA, 2007).

## User Documentation

The E-commerce website will include support documentation for users needing troubleshooting assistance and user account creation guidance. There is also to be a chat-bot that users may request needed assistance.

## Assumptions and Dependencies

The E-commerce website has a deadline of six months. The budget for the project is fifty-thousand dollars. The project team may not consist of more than eight people, including the designers, architects, database administrators, developers, testers, and project managers.

# External Interface Requirements

## User Interfaces

For consistency, every page of the E-commerce must have the same header, footer, and navbar. There needs to be a back arrow button located in the same place of each page so users may return to their previous page as well as a home button that is clearly unctio. Font Awesome should be the sole source for any icons use on the website. Users of the website should be able to TAB through pages and all images should contain alt-tags. All fonts and colorsused in the website should be selected for visual appeal, usability, and readability. The page layout should be consistent between all pages and advertisements should be positioned so they do not draw attention away from the main content of the page. Any forms on the website should contain clear feedback for filling them out and correcting any errors that are made.

## Hardware Interfaces

All major devices, be it computer, tablet, or phone to be able to access the E-commerce website. HTTP and HTTPS will handle site requests and TCP for resource transfers between web servers and clients. Screen reader support should also be included in the E-commerce website design.

## Software Interfaces

The E-commerce website should contain APIs for working with relational databases, payment acceptance, and single signin options for Facebook, Google, and Twitter. The site should function on IOS, Android, Windows, Mac, Linux, and Chromium.

## Communications Interfaces

Encryption will be used for securing all PII (personally identifiable information). Users will be able to opt in for newsletters and special deals and receive emails to confirm orders and shipping.

# System Features

## New User Account Creation

* + 1. Description and Priority
* Priority: High
* Description: Users should be able to create an account
* Benefit: 9

4.1.2 Stimulus/Response Sequences

* User prompted to login or create account
* User clicks/selects register new account
* User fills out form data
* User receives confirmation or instructions for correcting errors

4.1.3 Functional Requirements

* REQ-1: Website presents buttons for login and registering on homepage
* REQ-2: Website presents registration form if register button selected
* REQ-3: Form data validated, sanitized, processed, and submitted to database when submit button selected
* REQ-4: Instructions provided if data entered fails to meet requirements
* REQ-5: Success message presented after account successfully created

## Existing Account Login

* + 1. Description and Priority
* Priority: High
* Description: Users should be able to login to account
* Benefit: 9

4.1.2 Stimulus/Response Sequences

* User prompted to login or create account
* User clicks/selects login
* User enters login form data
* User clicks/selects submit
* User provided confirmation that they have been logged in or clear instructions for correcting their input

4.2.3 Functional Requirements

* REQ-1: Website presents buttons for login and registering on homepage
* REQ-2: Website presents login form if login button selected
* REQ-3: Form data validated, sanitized, processed, and submitted to database when submit button selected
* REQ-4: Instructions provided if data entered fails to meet requirements
* REQ-5: Data is compared to database information
* REQ-6: Success message presented after account successful login

## User Searches Items

* + 1. Description and Priority
* Priority: High
* Description: Users can search for items. This is part of the website main unctionality
* Benefit: 8

4.3.2 Stimulus/Response Sequences

* User types into the search bar
* User is provided with search suggestions
* User selects a search suggestion
* User is presented with a list of search results
* User selects filters to narrow search results

4.3.3 Functional Requirements

* + REQ-1: Website should have a search bar
  + REQ-2: Website should dynamically present user with search suggestions
  + REQ-3: Website should return search results when user selects suggestion
  + REQ-4: Website should dynamically generate filters depending on the category that the item falls into
  + REQ-5: Website should update search results when filters are selected

## User Views Item Details

* + 1. Description and Priority
* Priority: High
* Description: Users read item details. Users will be reluctant to buy if they do not have a description.
* Benefit: 7

4.4.2 Stimulus/Response Sequences

* User selects item from search results list
* User views item details

4.4.3 Functional Requirements

* + REQ-1: Website provides link to item descriptions
  + REQ-2: Website redirects to page populated with selected item

## User Add Item to Cart

* + 1. Description and Priority
* Priority: High
* Description: Users can add item to virtual cart. This is part of the main website functionality
* Benefit: 8

4.5.2 Stimulus/Response Sequences

* User clicks/selects add to cart
* User presented option to checkout or continue shopping
* User redirected to search page if continue shopping selected
* User redirected to checkout page if checkout selected

4.5.3 Functional Requirements

* + REQ-1: Item detail page should have an “add to cart” button
  + REQ-2: Website should dynamically update cart icon with the correct number of items in cart
  + REQ-3: Website should give option to continue shopping or go to checkout
  + REQ-4: Website should redirect to search page if continue shopping selected
  + REQ-5: Website should redirect to checkout page if checkout selected

## User Cart View

* + 1. Description and Priority
* Priority: Medium
* Description: Users can view cart prior to finalizing purchases.
* Benefit: 7

4.6.2 Stimulus/Response Sequences

* User clicks/selects cart icon
* User can view items in cart
* User can edit or delete items in cart

4.6.3 Functional Requirements

* + REQ-1: Website should have a cart icon in top right of each page that links to cart view page
  + REQ-2: User should see items added in a list in cart view page with short descriptions
  + REQ-3: User can delete items from cart, edit quantity of items to purchase, and price of items
  + REQ-4: Website should show projected cost total for items in cart
  + REQ-5: Website should update after any changes made to cart list
  + REQ-6: There will be a button to go to checkout in cart view page

## User Can Checkout and Finalize Purchase

4.3.1 Description and Priority

* Priority: High
* Description: Users can finalize the purchase at checkout page. This is part of the website main unctionality
* Benefit: 9

4.3.2 Stimulus/Response Sequences

* User clicks/selects cart icon
* User clicks/selects checkout from cart page
* Website presents checkout form to user
* User fills out checkout form
* User selects submit on checkout form
* Website gives confirmation of success or clear instructions to fix errors

4.3.3 Functional Requirements

* + REQ-1: Website should have a checkout button on cart view page
  + REQ-2: Website should redirect user to checkout page
  + REQ-3: Website should present checkout form to user
  + REQ-4: Website validates, sanitizes, processes, and updates databases when user selects submit on checkout form
  + REQ-5: Website indicates successful or clear instructions to correct errors

# Other Nonfunctional Requirements

## Performance Requirements

Speed and reliability are key to having a user return in the future. New accounts should take no longer than a minute to fill out and submit. After searching for an item, the results list should take no more than three seconds to display to the user. After selecting checkout, it should not take longer than two minutes for the user to complete the checkout form and receive a confirmation message.

## Safety Requirements

A session should timeout after five minutes of inactivity without confirmation from the user to continue. Before confirming an order, the E-commerce site should verify the user to be over the age of fourteen, all PII should be protected, and passwords and financial information needs to be encrypted.

## Security Requirements

Before a user gets access to an account, the E-commerce site must verify the account credentials. All of a users PII needs to be protected and secure. Passwords and financial data must be encrypted and transactions handled through secure processing methods.

## Software Quality Attributes

The E-commerce website needs to be reliable and fast to prevent interruptions that may deter users from returning. The site should be easy to navigate and use. Additionally, it should follow the guidelines for accessibility outlined by the ADA (2007). The E-commerce website needs to be scalable to support business and userbase growth. Upon completion of acceptance by the client, they will be responsible for maintenance, so the site needs to be modifiable, maintainable, and testable to allow new features to be added.

## Business Rules

Administrators need to be able to assit users with resetting their account credentials. They need to be able to search for orders using order ID numbers for both current and past orders for reporting purposes.

**Testing Levels and UML Models**

**Introduction**

Throughout the development lifecycle, it is vital to test software to ensure the client receives a quality product. Effectively performing tests can save on costs, reduce the development time, improve the quality of code, improve maintainability of the product, locate and correct faults earlier, and ensure the product meets the client’s requirements. According to Spillner et al., the four levels of testing are component testing, integration testing, systems testing, and acceptance testing (2014).

**Component Testing**

Component testing for the E-commerce site consists with testing each of the site’s functions independently. Component testing is done to ensure that each function meets the specifications and works properly. In theory, the most critical defects, outside of interaction between components, should be located and corrected during component testing.

**Integration Testing**

After all of the components have passed the component testing, the next level of testing for the E-commerce site is integration testing. Integration testing is done to ensure that the components interact with each other properly and there are no problems with the interfaces between them. Integration testing is performed in isolation using fabricated data where the results that should occur are already known. Integration testing is also where the interfaces, whether terminal or GUI (graphical user interface), are tested if they meet the requirements of the client, are intuitive, and are responsive.

**Systems Testing**

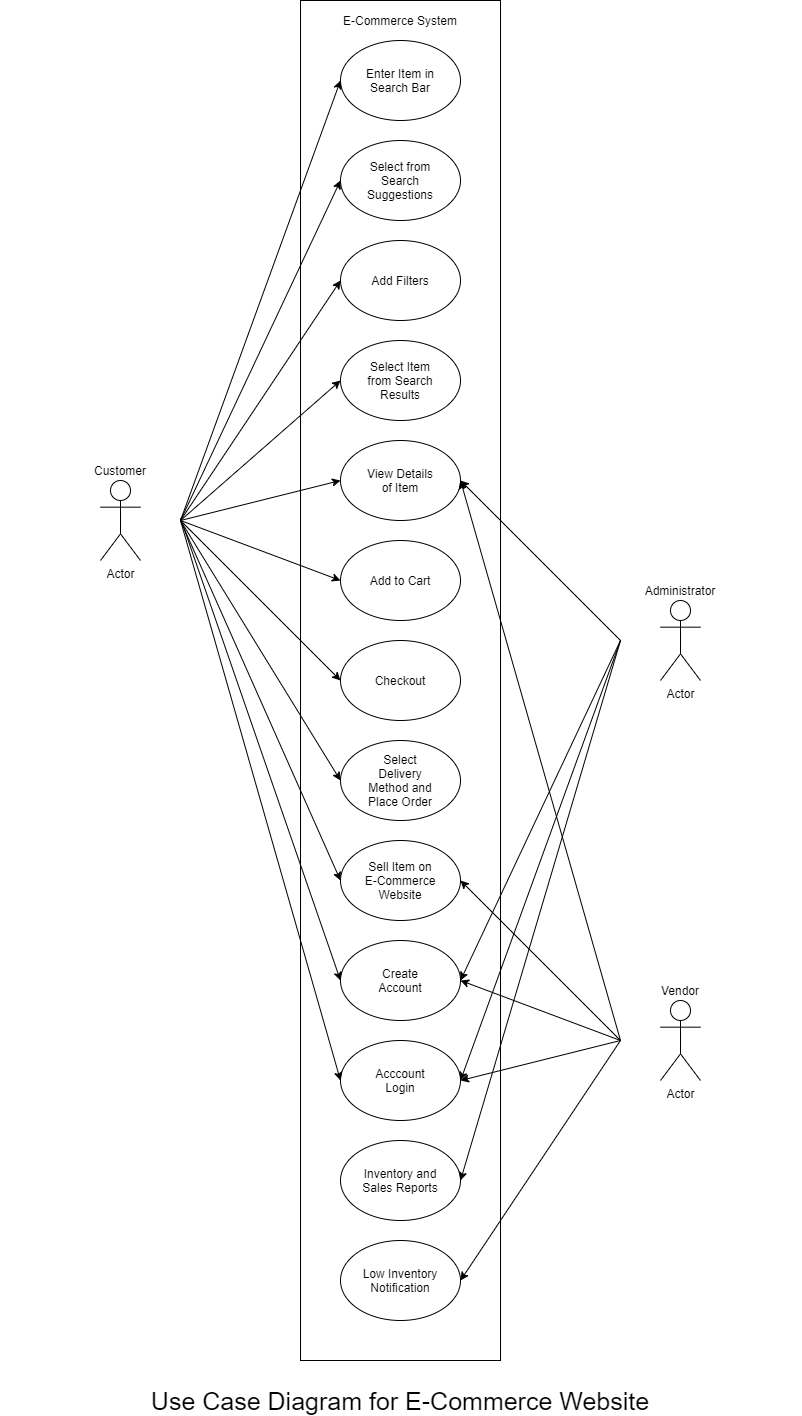
Systems testing is often done with the aid of IT and business professionals. Systems testing makes sure that the product, as a whole, functions properly in the environment that it will reside in and will not crash the system after it is implemented. This is the last level of testing prior to bringing the client in to examine the E-commerce site.

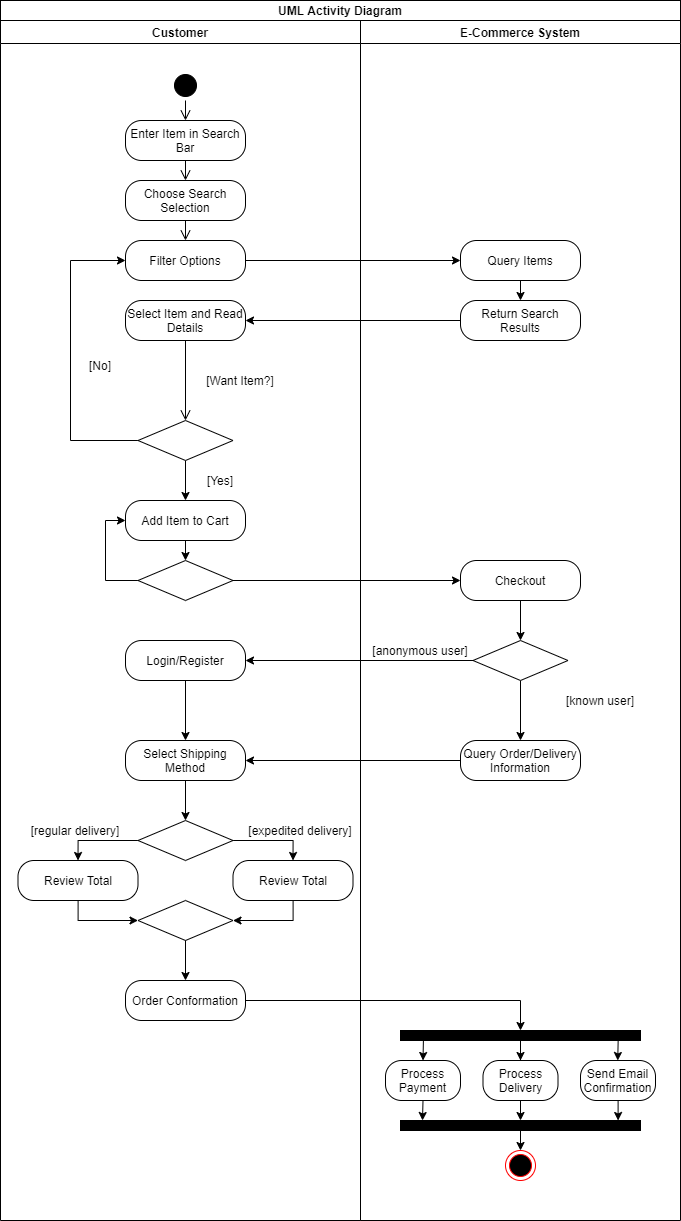
**Acceptance Testing**

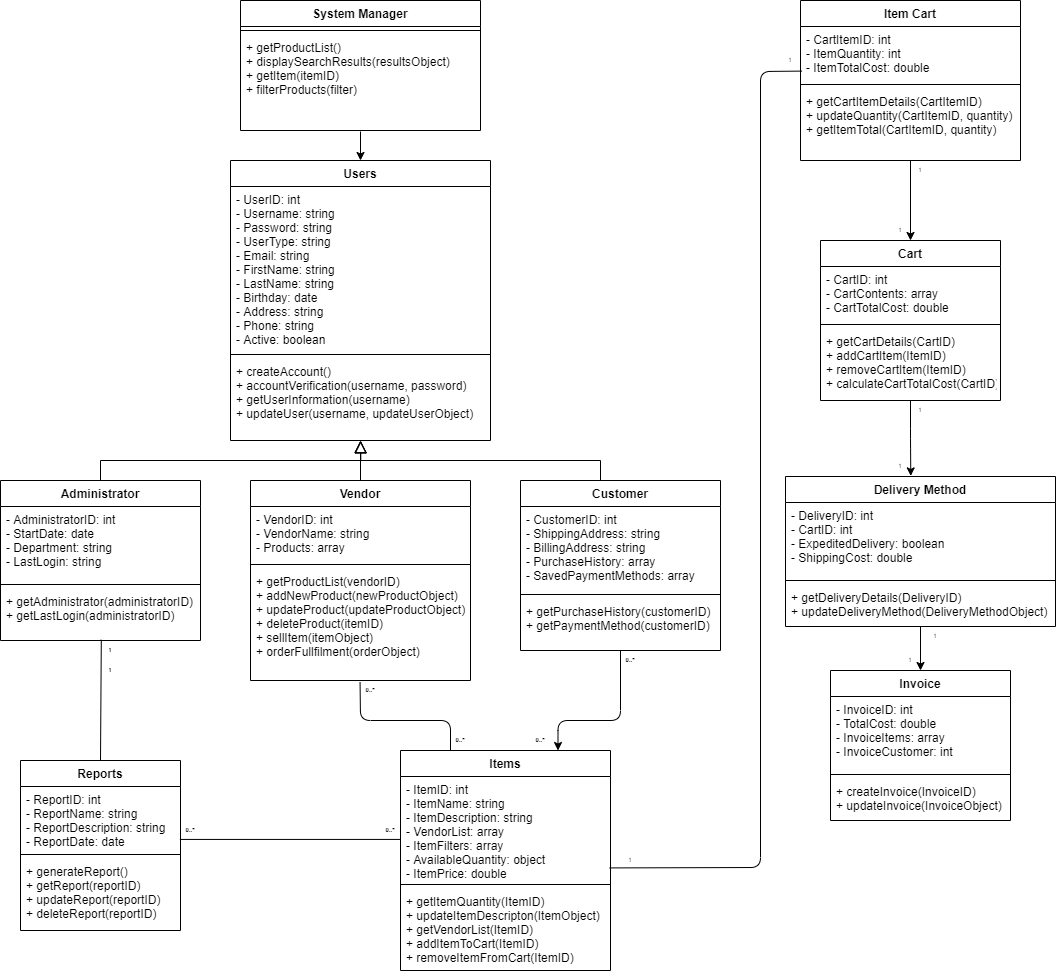
The final level of testing of the E-commerce site is acceptance testing. Acceptance testing is performed by the client to determine if the product meets their needs and specifications. The client puts it through it’s paces and decides to either accept the product or discuss what, if any, changes need to be made before they will take possession and the project is considered completed.

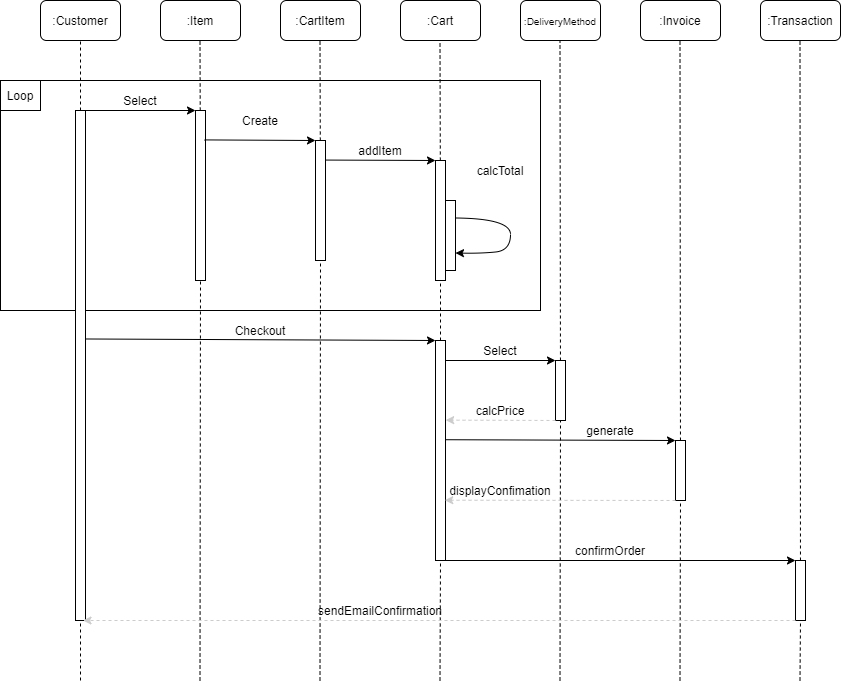
**UML Diagrams**

UML diagrams are used to visualize how a system functions and how data should flow. They are used to aid developers and testers in designing the components and test cases. Following are a use-case diagram, activity diagram, class diagram, and sequence diagram for the E-commerce site.



****

****

****

**Testing Strategies**

**Overview of Review Stategies**

**Walkthroughs**

First, here is an overview of each of the review strategies, walkthrough, inspection, technical review, and informal review. The walkthrough, according to Spillner, Linz, and Schaefer, is an informal review strategy used for finding “defects, ambiguities, and problems in written documents” (2014). The goal of a walkthrough is to educate the audience about the project and is generally lead by the author of the documents. “The main objectives of a walkthrough are mutual learning, development of an understanding of the review object, and error detection.” (Spillner, Liz, & Shaefer, 2014). There is usually no time limit in the meeting, little preparation is required, and questions from the audience is spontaneous.

**Inspections**

Inspections are a formal review with rules, protocols, checklists, and clearly defined roles for the audience (Spillner, Liz, & Shaefer, 2014). In planning an inspection, the primary focus is defining the objectives of the meeting. Facilitating the meeting is a moderator who is tasked with following a clear agenda and keeping the meeting moving smoothly and on point. Spillner, Linz, and Schaefer state the goals of an inspection as “finding unclear items and possible defects, measuring review object quality, and improving the quality of the inspection process and the development process” (2014). Below is a sample checklist template for an inspection (Fox, 1999).

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

**Technical Reviews**

For technical reviews, the author is not usually present and a recorder is present to take notes and consolidate feedback from the reviewers. The recorder also generates the final results of the meeting. The goal for technical reviews is ensure that the test objects meet the specifications of the project (Spillner, Liz, & Shaefer, 2014). Prior to the meeting, the reviewers do considerable preparation going over the test objects and organize their feedback concerning it. The reviewers should have a certain level of technical expertise with the parts they are reviewing in order to have a successful meeting. During the meeting, discussions take place about the feedback the reviewers prepared, proposals for alternatives, and errors and defects. Below are some templates from Carnegie Melon for technical reviews (1991).

Text

Description automatically generated

Text, letter

Description automatically generated

Text, letter

Description automatically generated

**Informal Reviews**

Informal reviews, as described by Spillner, Linz, and Shaefer, are less formal technical reviews initiated by the author. The author selects the reviewers and gives them the time to present their feedback (2014). Because no real meeting is scheduled, there are no templates involved and results are not generally recorded. As these reviews mostly involve sending messages amongst the reviewers, little effort is required and costs are low.

**Recommended Review Strategies**

As for the E-commerce website, all of review strategies are recommended except inspections. While inspections would be beneficial, they require significant amounts of time, coordination, and expertise. The other three review types should provide sufficient methods for verifying requirements, readability, maintainability, locating defects, and improving code quality at a reduced cost of time and money.

**Static Testing Strategies**

**Tools**

For the E-commerce project static testing, the tools that will be needed are an IDE or text editor, a compiler, and a linter. According to Spillner et al., these tools help developers and testers check interface consistency, identify syntax errors, locate dead code, cross-reference program elements, detect undeclared variables, and check data types (2014). Additionally, these tools will aid in debugging and help ensure the code is readable and consistent.

**Conventions and Standards**

"Compliance to conventions and standards can also be checked with tools. For example, tools can be used to check if a program follows programming regulations and standards." (Spillner, Linz, & Shaefer, 2014). There are tools within IDEs and linters that should be used for implementing rules that enforce proper conventions and standards for coding. Additionally, there are style guides available for the various programming languages like the Google Java Style Guide (Google, n.d.).

**Dataflow Analysis**

The purpose of dataflow analysis on the E-commerce site is to reveal defects in the code. This type of analysis, according to Spillner et al, "checks the usage of every single variable", including variables that are defined, referenced, or undefined (2014). Dataflow analysis is also used for revealing dataflow anomalies. Spillner et al states “An anomaly is an inconsistency that can lead to failure but does not necessarily do so. An anomaly may be flagged as a risk” (2014).

**Control Flow Analysis**

Control flow analysis is also used to locate defects by using graphs to analyize program execution represented from a sequence of statements. Control flow diagrams are used for visualizing changes in the execution of the program caused in the logic of conditionals and loops. “Due to the clarity of the control flow graph, the sequences through the program can easily be understood and possible anomalies can be detected.” (Spillner et al, 2014). While anomalies may not cause a program failure, addressing them helps ensure compliance and standards are maintained. Following are some example diagrams from GeeksForGeeks (n.d.).

Background pattern

Description automatically generated with medium confidence

**Dynamic Testing Strategies**

**Blackbox Testing**

Aimed at verifying test objects, blackbox testing includes a group of techniques that ensures the test objects meet the specifications and that specified inputs result in expected outputs. Using blackbox testing techniques on the E-commerce site is crucial because, as stated by Spillner et al, the goal of these strategies is “the verification of the functionality of the test object. It is indisputable that the highest priority is that the software work correctly. Thus, black box techniques should always be applied.” (2014). For the E-commerce site, the following blackbox techniques should be used, equivalence class partitioning, boundary value analysis, state transition testing, and use-case-based testing.

**Whitebox Testing**

Whitebox testing techniques will be used for the E-commerce site because they are aimed at testing the source code directly. These testing techniques are used to evaluate branching logic, conditionals, and loops. The whitebox testing techniques that should be utilized are statement testing, decision/branch testing, and the testing of conditions.

**Experience-Based Testing**

Experience-based (or intuitive-based) testing should be used in conjunction with whitebox and blackbox testing techniques. These tests take advantage of the experience and skill of developers and testers to generate tests that might have been overlooked in the systematic testing.

**Test Management Strategy**

**Test Management Strategy Overview**

For the E-commerce website, we need to develop a test management strategy. The test management strategy will cover test teams, exit criteria, test estimated effort, test and risk, incident reporting, defect classification, and configuration management. Having a good test management strategy will aid in keeping costs and time in line.

**Test Teams**

The consideration of the test teams and the approach to each level of tests are vitally important in achieving efficient development of the E-commerce website. Spillner et al. state, developers should not test their own code, if possible, “because there is a tendency to be blind to our own errors, it is much more efficient to let different people perform testing and development and to organize testing as independently as possible from development.” (2014). Because of this reason, ideally designated testers on the development team should do component testing, a deignated team (ie. Business or IT) on the project should do integration testing, and a designated team of specialists for systems testing. “Especially in system testing, it is often necessary to extend the test team by adding IT specialists, at least temporarily, to perform work for the test team”, allowing the system to be tested and viewed from multiple perspectives (Spillner, Linz, & Schaefer, 2014).

**Test Roles**

To conduct testing of the E-Commerce, the following roles should be assigned: test manager, test designer, test automator, test administrator, and tester (Spillner, Linz, & Schaefer, 2014). The test manager should have experience with personnel management, project management, quality management, and software testing. The test designer should have a test methods and specifications, testing, and software engineering in their skillset. Testing, scripting, programming, test tools, and automation experience is what is desired in a test automator. The test administrator needs a skillset that includes the setup and support of test environments, networking, and system administration. Testers need to be able to follow procedures, execute tests, report failures, and use test objects and testing tools (Spillner, Linz, & Schaefer, 2014).

**Exit Criteria**

Every test case should have specific exit criteria to guide testers in determining when a test is considered to reach completion. The importance of exit criteria is “They prevent tests from ending too early, for example, because of time pressure or because of resource shortages” (Spillner, Linz, & Schaefer, 2014). This exit criteria helps in determining when tests should be started or stopped for the E-commerce site. “To make a right decision to stop testing is an arduous resolution, the pre-defined exit criteria can help simplify this process. It is a very important step where all test processes get stopped and this decision is either made by the tester or the whole team together” (Nidagundi & Novickis, 2016), and by identifying them in the beginning, the testers and developers can rely on the criteria to determine when to stop the tests.

**Test Estimated Effort**

Consideration for both the cost of testing and the costs of undetected defects is vital when planning software testing. Initiating the test effort estimation in the planning phase is the test managers responsibility to ensure the proper assignment and distribution of resources. To provide the most reliable test effort estimation, the E-commerce website test manager will use as a base the estimations on former or similar projects of the same scale. Spillner et al states, “task-driven test effort estimation tends to underestimate the testing effort. Estimating based on experience data of similar projects or typical values usually leads to better results” (2014).

**Test and Risk**

To ensure that the most significant defects are revealed as early as possible, the E-commerce website will implement a risk-based prioritization. “Risk based prioritization of the tests ensures that risky product parts are tested more intensively and earlier than parts with lower risk. Severe problems (causing much corrective work or serious delays) are found as early as possible.” (Spillner, Linz, & Schaefer, 2014). This approach accomplishes a couple of things. First, it will prevent critical errors from downstream effects. Second, it will reduce the costs and time spent handling critical defects that might otherwise make it to production.

Implementing risk management techniques (identifying, prioritizing, and mitigating risks) is important in reducing risks. According to Spillner et al, risk is “the mathematical product of the loss or damage due to failure and the probability (or frequency) of failure resulting in such damage” (2014). Testing, itself, can be used as a technique for risk management as it “provides information about existing problems and the success or failure of correction” (Spillner, Linz, & Schaefer, 2014).

**Incident Reporting**

Throughout the development and testing of the E-Commerce Website, incident reporting will be done for documenting and managing incidents. Every significant and legitimate defect, that is not a result of badly designed testing, should be documented. The following template from *Software testing foundations: A study guide for the certified tester exam* (4th ed.) will be used for documenting incidents (Spillner, Linz, & Schaefer, 2014).

Table

Description automatically generated

Using a template such as this allows incident communication to be consistent, in turn, allowing developers to easily reproduce the defects in order to fix them. “Determining the impact in an

understandable way for developers (data loss, functionality loss, software instability, etc.) and for customers and the hierarchy (impacts in financial terms or in usability terms, noncompliance to requirements, etc.) enables a quick recognition of the anomaly.” (Homès, 2012).

Spillner et al states that an incident report will contain the tested software, test environment, name of the tester, the class containing the defect, the defect prioritization, and information relevant to locating and reproducing the defect (2014). Incident reporting may be done by testers, developers, clients, and users, and the incident reports should be updated upon corrections, thus helping with tracking and managing incidents.

**Defect Classification**

Prioritizing incidents related to the E-commerce website will be implemented through defect classification. There are five levels of severity that reflect the level of impairment a defect causes: 1-Fatal, 2-Very Serious, 3-Serious, 4-Moderate, and 5-Mild. Following is a table from *Software testing foundations: A study guide for the certified tester exam* (4th ed.) that shows the criteria for determining the severity level.

Table

Description automatically generated

Additionally, four levels of prioritization are to be used to identify how quickly the problem should be resolved: 1-Immediate, 2-Next Release, 3-On Occasion, and 4-Open. The following table from *Software testing foundations: A study guide for the certified tester exam* (4th ed.) has the criteria for the prioritization levels.

Graphical user interface

Description automatically generated with medium confidence

**Configuration Management**

Throughout the development of the E-commerce site, configuration management will be used to track the version history of the project and allow multiple developers to contribute without interfering with each others work on the project. A number of problems can be avoided with good configuration management, such as the overwriting of each others code by developers, component integration problems due to unknown versions, and testing difficulties from changes that are untraceable or not knowing which version of the test object that test cases belong to (Spillner et al, 2014). By implementing version management, configuration identification, incident and change status control, and configuration audits, these circumstances can be avoided.

**References**

ADA. (2007). Website Accessibility Under Title II of the ADA. Retrieved April 5, 2020 from <https://www.ada.gov/pcatoolkit/chap5toolkit.htm>

Carnegie Mellon Software Engineering Institute. (1991). The Software Technical Review Process. Retrieved April 18, 2020 from https://apps.dtic.mil/dtic/tr/fulltext/u2/a236122.pdf

Fox, C. (1999). Java Inspection Checklist. Retrieved April 18, 2020 from <http://www.cs.toronto.edu/~sme/CSC444F/handouts/java_checklist.pdf>

GeeksForGeeks.com. (n.d.). Software Engineering: Control Flow Graph (CFG). Retrieved April 18, 2020 from https://www.geeksforgeeks.org/software-engineering-control-flow-graph-cfg/

Google. (n.d.). Google Java Style Guide. Retrieved April 18, 2020 from <https://google.github.io/styleguide/javaguide.html>

Homès, B. (2012). Fundamentals of Software Testing, John Wiley & Sons, Incorporated. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/ashford- ebooks/detail.action?docID=1120766. Created from ashford-ebooks on 2020-04-25 14:33:08.

Kumar, J. (2015). [The e-commerce problem statement](https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781783981847/1/ch01lvl1sec12/the-e-commerce-problem-statement). In *Apache Solr patterns*. Packt Publishing. <https://bit.ly/2Rn8inE>

Nidagundi, P., & Novickis, L. (2016). Introduction to Lean Canvas Transformation Models and Metrics in Software Testing. Applied Computer Systems, 19(1), 30.

Spillner, A., Linz, T., & Schaefer, H. (2014). [*Software testing foundations: A study guide for the certified tester exam* (4th ed.)](https://ashford.instructure.com/courses/86077/modules/items/4350686). Rocky Nook.

Wiegers, K. E. (1999). [*Software requirements specifications for <project>*](https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc). [Template]. https://web.cs.dal.ca/~hawkey/3130/srs\_template-ieee.doc