QA Ecommerce Website

**QA Test Plan**

**Revision V 5**

***QA Lead* –** **James Ugdang**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version #** | **Date** | **Author** | **Description** |
| V0.1 | 9/18/2024 | James Ugdang | First Draft – Revised Cover Page, |
| V0.2 | 9/19/2024 | James Ugdang | Fleshed out Section 1.1 Identification and 5.1 software development cycle (waterfall) |
| V0.3 | 9/22/2024 | James Ugdang | Added reference in section 2, added details on 5.1 (STLC) and 5.2 (Test Process) |
| V0.4 | 9/23/2024 | James Ugdang | Fleshed out 5.3.1 (Test Levels) and 5.3.2 (Test Types) |
| V 1 | 9/24/2024 | James Ugdang | Continued adding details on 5.3.2 (Test Types), Fleshed Out 5.3.3 (Static Tests) and 5.3.4 (Dynamic Tests) |
| V1.1 | 10/1/2024 | James Ugdang | Fleshed out Section 2 (Reference), Section 2.2 (Management and Organization), Section 7.1 (Resource Requirements), Section 7.2 (Training Considerations) |
| V 2 | 10/2/2024 | James Ugdang | Added more definitions in Section 2.1, Updated the Table of Contents |
| V2.1 | 10/21/2024 | James Ugdang | Fleshed out Section 4.6 (Data Requirements) |
| V 3 | 10/22/2024 | James Ugdang | Continued fleshing out section 4.6 (Data Requirements, started fleshing out 4.6 (Test Data Requirements), 5.5.1 (Defect Tracking Tool), 10.1 (Defect Priority Definition), 10.2.1 (Readiness (Entrance Criteria)), 10.2.3 (Suspension and Resumption), 10.2.4 (Completion (Exit Criteria)), 10.3.1 (Readiness (Entrance Criteria)), 10.3.3 (Suspension and Resumption), 10.3.4 (Suspension and Resumption), 10.4.1 (Readiness (Entrance Criteria)), 10.4.3 (Suspension and Resumption), 10.4.4 (Completion (Exit Criteria)) and Section 2 (References) |
| V 3.1 | 10/30/2024 | James Ugdang | Fleshed out section 1.3 (Assumptions) |
| V 3.2 | 10/31/2024 | James Ugdang | Continued fleshing out section 1.3 (Assumptions), added details in section 2 (references), started fleshing out section 3.1 (Test Items), section 4.1 (Hardware Requirements), section 4.2 (Consumable Requirements), section 4.3 (Facilities and Hardware), section 4.5 (Software, Tools, and Utilities), and section 4.6 (Test Data Requirements), and section 11 (Risks and Contingencies). |
| V 4 | 11/02/2024 | James Ugdang | Finished up section 11 (Risks and Contingencies). |
| V 4.1 | 11/11/2024 | James Ugdang | Added details in Section 4.1(Hardware Requirements) fixed the exact date, Finalized section 4.6 (Test Data Requirements) |
| V 5 | 11/12/2024 | James Ugdang | Fleshed Out Section 6 (Estimates and Schedules), Appendix A and Appendix B, added details in Section 2 (References) and updated the table of contents. |

Table of Contents

[1 Introduction 5](#_Toc182257273)

[1.1 Identification 5](#_Toc182257274)

[1.2 Objectives 5](#_Toc182257275)

[1.3 Assumptions 5](#_Toc182257276)

[2 References 5](#_Toc182257277)

[2.1 Definitions, Acronyms, and Abbreviations 6](#_Toc182257278)

[2.2 Management and Organization 7](#_Toc182257279)

[3 Software Work Products to be Tested 8](#_Toc182257280)

[3.1 Test Items 8](#_Toc182257281)

[3.2 Features to be Tested 9](#_Toc182257282)

[3.3 Features Not To Be Tested 9](#_Toc182257283)

[4 Environmental Needs 10](#_Toc182257284)

[4.1 Hardware Requirements 10](#_Toc182257285)

[4.2 Consumables Requirements 10](#_Toc182257286)

[4.3 Facilities and Hardware 10](#_Toc182257287)

[4.4 Configuration of Facilities and Hardware 11](#_Toc182257288)

[4.5 Software, Tools, and Utilities 11](#_Toc182257289)

[4.6 Test Data Requirements 12](#_Toc182257290)

[5 Test Approach 14](#_Toc182257291)

[5.1 Software Development Lifecycle (Waterfall) 14](#_Toc182257292)

[5.2 Test Process 15](#_Toc182257293)

[5.3 Test Scope 16](#_Toc182257294)

[5.3.1 Test Level 16](#_Toc182257295)

[5.3.2 Test Types 17](#_Toc182257296)

[5.3.3 Static Testing 17](#_Toc182257297)

[5.3.4 Dynamic Testing 17](#_Toc182257298)

[5.4 Reports 18](#_Toc182257299)

[5.5 Defect Tracking 18](#_Toc182257300)

[5.5.1 Defect Tracking Tool 18](#_Toc182257301)

[5.5.2 Defect Tracking Approach 19](#_Toc182257302)

[6 Estimates and Schedule 19](#_Toc182257303)

[7 Resources and Training 20](#_Toc182257304)

[7.1 Resource Requirements 20](#_Toc182257305)

[7.2 Training Considerations 21](#_Toc182257306)

[8 Deliverables 21](#_Toc182257307)

[9 Acceptance of Externally Developed Software 22](#_Toc182257308)

[10 Criteria 22](#_Toc182257309)

[10.1 Defect Severity Definition 22](#_Toc182257310)

[10.1 Defect Priority Definition 23](#_Toc182257311)

[10.2 Build Test Correct – Build 1 and Build 2 (Fix Build) 23](#_Toc182257312)

[10.2.1 Readiness (Entrance Criteria) 23](#_Toc182257313)

[10.2.2 Item Pass/Fail Criteria 24](#_Toc182257314)

[10.2.3 Suspension and Resumption 25](#_Toc182257315)

[10.2.4 Completion (Exit Criteria) 25](#_Toc182257316)

[10.3 System Test 25](#_Toc182257317)

[10.3.1 Readiness (Entrance Criteria) 25](#_Toc182257318)

[10.3.2 Item Pass/Fail 25](#_Toc182257319)

[10.3.3 Suspension and Resumption 25](#_Toc182257320)

[10.3.4 Completion (Exit Criteria) 25](#_Toc182257321)

[10.4 Customer Acceptance Test 26](#_Toc182257322)

[10.4.1 Readiness (Entrance Criteria) 26](#_Toc182257323)

[10.4.2 Item Pass/Fail 26](#_Toc182257324)

[10.4.3 Suspension and Resumption 26](#_Toc182257325)

[10.4.4 Completion (Exit Criteria) 26](#_Toc182257326)

[10.5 System Acceptance 26](#_Toc182257327)

[11 Risks and Contingencies 26](#_Toc182257328)

[12 Approvals 27](#_Toc182257329)

[13 Appendix A – Full Estimate 28](#_Toc182257330)

[14 Appendix B - Schedule 31](#_Toc182257331)

# Introduction

The purpose of this document is to highlight the test plan for the QA Ecommerce Website project with the use of Waterfall methodology. The testing will focus on functional aspects only to ensure that the system adheres to the Software Requirements Specification (SRS).

## Identification

The project's name is QA Ecommerce Website, for which the WBS code is QAP1536.

## Objectives

The purpose of this test plan is to define all processes and practices that will be employed on this project to ensure the final product achieves a high level of quality.

## Assumptions

The following assumptions have been made and should be considered as it relates to this project and quality.

***Key Assumptions***

* Assumption 1 – We assume the software for each planned build will be delivered by the test execution start date for each build as defined in the test schedule.
* Assumption 2 – We assume 2 fully trained and skilled testers and 1 highly experienced test lead will be available from the onset of the planning phase of the project.
* Assumption 3 – We assume that specific hardware such as desktops, laptops, etc., and other office equipment for testing will be provided for the QA team.
* Assumption 4 – We assume the physical setup of the test lab, such as chairs, desks, and security features, is provided for the QA team.
* Assumption 5 – We assume the development team will resolve issues at a rate that allows no interruption to testing and still falls within each phase's entrance criteria.

# References

The following references below are considered upon creation of this document.

|  |  |
| --- | --- |
| SN | Reference |
| 1 | IEEE 829 Test Plan template |
| 2 | SRS file – Ecommerce System Requirement Specification\_V1 |
| 3 | Ecommerce Mindmap\_James Ugdang\_V1 |
| 4 | RTM - Ecommerce Requirement Priorities |
| 5 | Ecommerce Mindmap\_James Ugdang\_V2 |
| 6 | Test Cases\_Ecommerce\_James\_V1 |
| 7 | Estimate\_James Ugdang\_V1 |
| 8 | Project Schedule\_James Ugdang\_V1 |

## Definitions, Acronyms, and Abbreviations

There are standard quality assurance terms, acronyms, and abbreviations found in this document that are not widely known. The following is a list of these terms and their definitions.

* **CAT (Customer Acceptance Testing)-** When the customer accepts the project to perform their internal testing before releasing it.
* **Software defect log**- A defect tracking system utilized to tracks defects throughout the project. The log can be created through defect management software (JIRA) or in Excel.
* **Requirements**- The needs or expectations the customer has for the project are the building blocks or foundation for the project; they need to detail every area of the project and the expected result.
* **RTM (Requirements Traceability Matrix**)- is a document that maps out the requirements to the test cases to ensure all the requirements are covered during testing.
* **Defect Escapes**- When a defect is not found until the product is released to the customer for CAT.
* **BTC (Build, Test, Correct)-** The program is broken up into manageable testing phases; there are 2 BTC's or builds of test cases.
* **Functional Testing**- Tests the behavior of the program; it's based on the requirements and the functionality of the program.
* **Non-Functional Testing**- Tests the quality characteristics (Reliability, Usability, Efficiency, Maintainability, Portability) and the program's security.
* **Static Testing**- Testing, which does not involve the software code directly, dynamic testing is performed to detect defects in work products such as documents, schedules, diagrams.
* **Dynamic Testing**- Testing of the actual code and software functionality, it is used to detect defects in how the software performs.
* **SRS (Software Requirement Specification) –** is a document that outlines the functional and non-functional requirements of a software system.
* **IEEE (Institute of Electrical and Electronics Engineers) –** Is a professional organization that is well known for their contributions to various fields, such as information technology, computer science, electrical engineering and whatnot.
* **Smoke Testing –** it is a test that needs to perform before taking the new software into a deeper level of testing. This is necessary to ensure that the new software’s basic functionalities are working as intended, and if it fails, it is considered too unstable for further testing and the build is rejected.
* **ISTQB (International Software Testing Qualifications Board) –** it is a global, non-profit organization that offers certification and standardized qualification for software testers.

## Management and Organization

The table below list the expectation of each Project Team Members all throughout the Testing phase that involves planning and execution of the test.

|  |  |  |
| --- | --- | --- |
| Roles | Responsibilities | Names |
| Business Analyst | * Analyze Organization * Document Business Process or Systems * Assess the Business Model and its integration with Technology * Create SRS (Software Requirement Specification) | Emma Wilson |
| Project Manager | * Overseeing Project Planning * Managing Procurement * Project Execution | Liam Tremblay |
| QA Manager | * Arrange Training for the Junior QA Tester * Training Junior QA for Formal Inspection Process * Training Junior QA for JIRA Functionality * Training Junior QA for TestCollab | Sophie Patel |
| Test Lead | * Initial Analysis and Costing * Develop Initial Schedule * Develop Test Plan * Attend Test Plan Inspection Meeting * Update/Rework Test Plan * Attend Requirement Inspection * Prepare RTM (Requirement Traceability Matrix) * Update estimate in all phases * Create Test Sets during System Testing (3 BTC’s) * Prepare Test Cases/Scenarios * Review Test Cases/Scenarios * Participate in Test Case/Scenario peer reviews * Update/Rework Test Cases/Scenarios * Attend Test Readiness Meetings * Setup Environment * Maintenance Test Cases * Conduct Smoke Tests * Create Test Incident Reports (TIR) in JIRA | James Ugdang |
| QA Tester | * Prepare Test Cases/Scenarios * Review Test Cases/Scenarios * Participate in Test Case/Scenario peer reviews * Update/Rework Test Cases/Scenarios * Attend Test Readiness Meetings * Setup Environment * Maintenance Test Cases * Conduct Smoke Tests * Create Test Incident Reports (TIR) in JIRA | Noah MacDonald |
| Junior QA Tester | * Prepare Test Cases/Scenarios * Review Test Cases/Scenarios * Participate in Test Case/Scenario peer reviews * Update/Rework Test Cases/Scenarios * Attend Test Readiness Meetings * Setup Environment * Maintenance Test Cases * Conduct Smoke Tests * Create Test Incident Reports (TIR) in JIRA Learn Formal Inspection Process * Learn JIRA Functionality * Learn Test Collab | Olivia Nguyen |
| Software Developer | * Frontend (GUI) Developer | Ethan Roy |
| Software Developer | * Backend Database (DB) Developer | Ava Thompson |

# Software Work Products to be Tested

This section details the high-level areas that will be tested as a part of this project, as well as the areas that specifically will not be tested.

## Test Items

The high-level test cares that will be covered with formal test cases to ensure adherence to requirements and fitness for use are as follows:

|  |  |
| --- | --- |
| **Component Area** | **Description** |
| **Homepage** | * **Username – valid and invalid character types, min/max. and above character limits, if the field is a required field, valid username and password combination, label text, location, tab order. All error handling and messages for negative tests.** * **Password - valid and invalid character types, min/max. and above character limits, if the field is a required field, valid username and password combination, label text, location, tab order. All error handling and messages for negative tests.** * **Register option – Functions on click, error handling on click, label text, placement, navigates to the “New User Registration Page”.** |
| **Forgot Password Page** | * **Email - valid and invalid character types, min/max. and above character limits, if the field is a required field, valid email format, location, tab order. All error handling and messages for negative tests.** |
| **New User Registration Page** | * **Username – valid and invalid character types, min/max. and above character limits, if the field is a required field, valid username and password combination, label text, location, tab order. All error handling and messages for negative tests.** * **Password - valid and invalid character types, min/max. and above character limits, if the field is a required field, valid username and password combination, label text, location, tab order. All error handling and messages for negative tests.** * **Full name - valid and invalid character types, min/max. and above character limits, if the field is a required field, label text, location, tab order. All error handling and messages for negative tests.** * **Email - valid and invalid character types, min/max. and above character limits, if the field is a required field, valid email format, location, tab order. All error handling and messages for negative tests.** * **Captcha - valid and invalid character types, min/max. and above character limits, if the field is a required field, label text, location, tab order. All error handling and messages for negative tests.** |
| **Book a Hotel Page** | * **First name - valid and invalid character types, min/max. and above character limits, if the field is a required field, label text, location, tab order. All error handling and messages for negative tests.** * **Last name - valid and invalid character types, min/max. and above character limits, if the field is a required field, label text, location, tab order. All error handling and messages for negative tests.** * **Billing Address - valid and invalid character types, min/max. and above alphanumeric character limits, if the field is a required field, label text, location, tab order. All error handling and messages for negative tests.** * **CVV - valid and invalid character types, min/max. and above numeric character limits, if the field is a required field, label text, location, tab order. All error handling and messages for negative tests.** |

## Features to be Tested

All high-level areas noted in the Test items section will have a full and complete detailing of formal test cases written and executed to ensure a high-quality delivery.

## Features Not To Be Tested

Non-Functional Test types will not be included at this time in our test efforts with exception of basic usability. Also, no third-party software interfacing with this project’s software will be tested and is assumed to have been tested by the developing company. Only the interfaces with our product will be verified.

# Environmental Needs

In the following section, all the facilities, hardware, consumables, software, and data requirements are listed with costs, quantities, and deadlines.

## Hardware Requirements

The table below is the list of Hardware requirements needed to carry out the Testing.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Hardware | Description | Quantity | | Price | Date needed by |
| Desktop Computer | Varying makes/models for testing | | 3 | $3,600 CAD | November 11 |
| Laptop | Varying makes/models for testing | | 3 | $3,000 CAD | November 11 |
| Monitor | Standard size, high resolution | | 3 | $600 CAD | November 11 |
| Keyboard | USB or Bluetooth for desktop setups | | 3 | $60 CAD | November 11 |
| Mouse | Optical mouse for desktop and laptop | | 3 | $90 CAD | November 11 |
| Mouse Pad | Standard, non-slip | | 3 | $15 CAD | November 11 |
| Printer | All-in-one printer for report and test case printing | | 1 | $300 CAD | November 11 |

## Consumables Requirements

The table below is the list of Consumables requirements needed by the Testing team in support with the daily operations.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Product | Description | Quantity | Price | Date needed by |
| Toner/Ink | Compatible with chosen printer model | 2 sets | $120 CAD | November 11 |
| Paper | Standard letter-size paper, pack of 500 sheets | 3 packs | $15 CAD | November 11 |
| Pens | Ballpoint pens, blue and black | 1 box (50) | $10 CAD | November 11 |

## Facilities and Hardware

The table below is the list of Facilities and Hardware needed to carry out the Testing.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Description | Quantity | Price | Date needed by |
| Desktop Computer | Multiple models for varied test scenarios | 3 | $3,600 CAD | November 11 |
| Laptop | Multiple models for varied test scenarios | 3 | $3,000 CAD | November 11 |
| Monitor | 24-inch, full HD display | 3 | $600 CAD | November 11 |
| Keyboard | Standard USB keyboard | 3 | $60 CAD | November 11 |
| Mouse Pad | Non-slip mouse pad | 3 | $15 CAD | November 11 |
| Printer | Laser printer with wireless capabilities | 1 | $150 CAD | November 11 |
| Desk | Workstation desk with space for desktops and monitors | 3 | $750 CAD | November 11 |
| Chair | Ergonomic office chair | 3 | $450 CAD | November 11 |
| Laptop Tray | Adjustable laptop stand | 3 | $90 CAD | November 11 |
| Anti-Static Mat | Anti-Static mat for desktops | 3 | $60 CAD | November 11 |
| Printer Stand | Stand for printer | 1 | $40 CAD | November 11 |
| Bookshelf | For storing documentation | 1 | $80 CAD | November 11 |
| Security Camera | Surveillance camera for securing the test lab | 1 | $100 CAD | November 11 |
| Door Lock | Swipe card access for secured entry | 1 | $200 CAD | November 11 |

## Configuration of Facilities and Hardware

All required hardware such as servers, network connection and the like will be supplied by the Contractor.

Office interior setup will be accommodated by the Contractor.

All set up required such as Test Management tool, Defect tracking tool and other testing tools that will be utilized during testing will be handled by the Testing team.

## Software, Tools, and Utilities

This section defines the Operating Systems needed to carry out the testing.

***Operating Systems***

|  |  |  |  |
| --- | --- | --- | --- |
| OS | Version | Language | Internet Browser |
| Windows | 11 | English | IE 9, 10, 11  Google Chrome (latest version),  Microsoft Edge, Brave |

***Applications/Tools***

|  |  |  |  |
| --- | --- | --- | --- |
| Vendor | Application | Version | Purpose |
| Microsoft | Excel | **Version 2409** | Status Reporting/Metrics Reporting |
| Microsoft | Word | **Version 2409** | Functional Design, Test Case Management |
| Microsoft | Power point | **Version 2409** | For data presentation |
| Microsoft | Microsoft Outlook | V16.0 | For Email communicator and meeting schedules |
| Microsoft | Edge | Version 113.0.1774.42 | Main web browser to be used in test execution |
| Google | Google Chrome | 129.0.6668.100 | Additional web browser to be used in test execution |
| Tricentis | qtest | 1.2.4 | This is a Test management tool for test case management |
| Atlassian | JIRA | V 8.3.0 | Defect tracking tool |

## Test Data Requirements

The test data will be built through test execution as defined below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Input | Test Data Type | Test Case ID | Comment |
| Username | John.Doe | Valid | 2 | Used for all test cases unrelated to negative login |
| Username | JohnJohnJohnJohnJohnJohnJohnDo | Valid | 6 | Max Character Test |
| Username | JohnJohnJohnJohnJohnJohnJohnDoe | Invalid | 5 | Above Max Character Limit |
| Username | J | Valid | 6 | Minimum Character Test |
| Username | John123 | Invalid | 7 | Invalid numeric characters |
| Password | Password1 | Valid | 9 | Used for all test cases unrelated to negative login |
| Password | PasswordPasswordPassword234567 | Valid | 12 | Max Character test |
| Password | PasswordPasswordPassword1234567 | Invalid | 11 | Above Max alpha numeric character Limit |
| Password | P | Valid | 12 | Minimum alphanumeric character Test |
| Email | first.last@companyname.com | Valid | 21 | Valid Email Format Test |
| Email | first.last123456789012345678901234@companyname.com | Valid | 24 | Max alphanumeric character test |
| Email | first.last1234567890123456789012345@companyname.com | Invalid | 23 | Above Max alphanumeric character Limit |
| Email | first.last | Invalid | 25 | Invalid Email Format test |
| Full Name | John Doe | Valid | 37 | Used for all test cases unrelated to negative login |
| Full Name | Johnathanjohnathan Maxwellmaxwell Doe Anderson Jr. | Valid | 40 | Max Character test |
| Full Name | John Doe 123 | Invalid | 41 | Invalid numeric characters |
| Full Name | JohnathanjohnathanJohnathan Maxwellmaxwell Doe Anderson Jr. | Invalid | 42 | Above Max alpha character Limit |
| Full Name | John Doe !@## | Invalid | 44 | Invalid special characters |
| Captcha | kFgGhQ | Valid | 52 | Used for all test cases unrelated to negative login |
| Captcha | kFgGhQkFgGhQkFgGhQkFgGhQkFgGhQ | Valid | 56 | Max Character Test |
| Captcha | kFgGhQkFgGhQkFgGhQkFgGhQkFgGhQk | Invalid | 57 | Above Max Character Limit |
| Captcha | K | Valid | 56 | Minimum Character Test |
| Captcha | kFgGhQ1 | Invalid | 58 | Invalid numeric characters |
| First Name | John | Valid | 62 | Used for all test cases unrelated to negative login |
| First Name | JohnJohnJohnJohnJohnJohnJohnJo | Valid | 64 | Max Character Test |
| First Name | JohnJohnJohnJohnJohnJohnJohnJohn | Invalid | 65 | Above Max Character Limit |
| First Name | John!@# | Invalid | 66 | Invalid Special Characters |
| First Name | J | Valid | 64 | Minimum Character Test |
| First Name | John123 | Invalid | 68 | Invalid numeric characters |
| Last Name | Doe | Valid | 70 | Used for all test cases unrelated to negative login |
| Last Name | DoeDoeDoeDoeDoeDoeDoeDoeDoeDoe | Valid | 72 | Max Character Test |
| Last Name | DoeDoeDoeDoeDoeDoeDoeDoeDoeDoeDoe | Invalid | 73 | Above Max Character Limit |
| Last Name | D | Valid | 72 | Minimum Character Test |
| Last Name | Doe123 | Invalid | 76 | Invalid numeric characters |
| Billing Address | 123 Maple Street Apt 4B Saint John, New Brunswick E2L 3V9 Canada | Valid | 78 | Normal Billing Address Positive Test |
| Billing Address | Apartment 12B, Building 9, Oakwood Residences  456745674567 Pinecrest Boulevard,  Near Greenfield Park, Block 4,  District 78, Maple Hills,  Springfield, New Brunswick,  E2K 5L9, Canada | Valid | 81 | Max 300 alphanumeric character test |
| Billing Address | Apartment 12B, Building 9, Oakwood Residences  4567456745674567 Pinecrest Boulevard,  Near Greenfield Park, Block 4,  District 78, Maple Hills,  Springfield, New Brunswick,  E2K 5L9, Canada | Invalid | 82 | Above Max Character Limit |
| Billing Address | A | Valid | 81 | Minimum Character Test |
| Billing Address | 123!@#@# Maple Street Apt 4B Saint John, New Brunswick E2L 3V9 Canada | Invalid | 84 | Invalid special characters |
| Credit Card Number | 30569309025904 | Valid | 86 | Used for all test cases unrelated to negative login |
| Credit Card Number | 4111111111111111 | Valid | 89 | Max numeric character limit |
| Credit Card Number | 41111111111111111 | Invalid | 90 | Above Max Numeric Character Limit |
| Credit Card Number | 4 | Valid | 89 | Minimum Numeric Character Test |
| Credit Card Number | 411111111111111a | Invalid | 91 | Invalid Alpha Characters |
| CVV | 123 | Valid | 97 | Minimum Numeric Character Test |
| CVV | 1234 | Valid | 97 | Max numeric character limit |
| CVV | 12345 | Invalid | 99 | Above Max Numeric Character Limit |
| CVV | 12 | Invalid | 100 | Below Minimum Numeric Character Limit |
| CVV | 123a | Invalid | 98 | Invalid Alpha Characters |

# Test Approach

This section covers the test Approach that will be taken inclusive of the software development methodology, the test process, the test levels and the test types.

## Software Development Lifecycle (Waterfall)

The Waterfall Methodology will be used as the SDLC on this project. This methodology can be described as:

It starts with collecting ‘Requirements’ and putting them into the document called Software Requirements Specification (SRS). This document serves as a guide for both developers and testers providing a detailed description of what the software is supposed to behave.

Followed by ‘Software Design’ and ‘Development’. The software will be designed and developed by software developers, and they will base the design on the ‘Software Requirements Specification (SRS).

Once the coding is completed and the software is finally developed, the ‘Testing’ will be started to ensure that the system adheres to the Software Requirements Specification (SRS). The software will go through a lot of testing phases where the QA Testers are responsible for the quality of the project to ensure that the product and/or the software will be ready for the deployment phase.  
  
And after all the testing phases have been done, the software will be ready for the ‘Deployment phase’, marking the start of the production and real-world use.

Lastly, the software will undergo ‘Maintenance’ to address bugs and issues that users will find and encounter after the deployment phase.

## Test Process

The Test Lifecycle or Process is defined as follows:

It is a subset of Software Development Lifecycle (SDLC), focused only on testing and quality assurance ensuring that the project or software works as it needs to be.

The process starts with the ‘Test Planning’ where the requirements documents are analyzed and validated (formal review) and the scope of testing is defined. The QA manager assigns the testers their roles and responsibilities and plans about the objectives of the testing and what approach can be used to meet those objectives. Also, within this phase, the risks, such as insufficient resources or tight timelines are analyzed.

‘Test Monitoring’ where the progress of the testing throughout the Software Testing Lifecycle (STLC) is monitored and ensures that a sufficient portion of the software is being tested. Within this phase, the progress of the testing is also communicated to stakeholders via test progress reports and provide important insights into the quality of the program/software.

‘Test Analysis’ which is the crucial part of the Software Testing Lifecycle (STLC). This is where the testers are going to analyze which parts of the specific requirement are going to be tested, in short, they’re going to break down the information into testable parts and identify the test conditions. Basically, the testers will analyze what to test, how to test it, and under what conditions. Also, this is where the testers evaluate the test basis (requirements) ensuring that there will be no defects that can affect the future quality of the software.

‘Test Design’ where the planning on choosing what techniques or strategies the testers are going to use like if they’re going to use functional, non-functional, Blackbox testing or whatnot. Also, within this phase, the testers are going to create detailed test cases as well.

‘Test Implementation’ involves the preparation of everything necessary to execute the test cases. The testers write or develop test cases within this phase and gather data such as user information, products or transactions and group the test cases for the functionality they will receive in a folder. The testers ensure that the test environment simulates production environment as precisely as possible. Also, within this phase, the test design and test implementation are often combined, meaning they can overlap and be part of the same activity.

‘Test Execution’ this is where the testers run the test cases, and the goal is to validate whether the system behaves as expected and logging the results whether the test case passed or failed. If the test case has failed, the tester will provide details like screenshots or logs and steps to reproduce the issue. The testers are going to report the defects based on the failures observed and once the defects have been resolved, the testers are going to do a retest to make sure that the defect has been really fixed and if it is, the testers will do a regression testing to ensure that there are no other new existing defects because of the newly fixed issue.

‘Test Completion’ is the final phase of the Software Testing Lifecycle (STLC), this is where all the previous testing phases have been carried out, and the QA team is ready to finalize the testing phase. The testers are going to create a test summary report for the tests results conducted, issues/defects found, issues/defects resolved, and any critical issues. Finalizing the test environment, the test data, and other testware, thus it can be used for later reuse. Within this phase, the team evaluates whether all objectives, coverage and the quality standards that are in the plan have been fulfilled. A formal meeting may be held for closure, ensuring that all stakeholders agree that the program/system is ready for production or the next phase.

## Test Scope

The following section defines to scope of testing that will be performed on this project inclusive of the test levels, test types and the work products that will undergo static and dynamic testing.

### Test Level

The following are the test levels that will be performed on this project.

There are 4 levels of software testing and here’s the breakdown:

‘Unit Testing’ is the first level in the software testing process and focuses on verifying the smallest testable parts of the software such as tests unit, classes in the code or individual functions. Unit Testing is usually performed by Developers who write the code and its part of the white box testing as well.

‘Integration Testing’ is the second level in the software testing process and focuses on how components or modules of the software work together and ensure they interact as specified by the technical design. Its main purpose is to ensure that combined modules or components function correctly as expected when integrated. This level of testing is usually performed by the development team when they combine their code but if we talk about comprehensive integration testing, it is more likely performed by QA teams. It is critical to ensure that the combined components work harmoniously and meet the overall system requirements which is why ‘Integration Testing’ is essential.

‘System Testing’ is the third level in the software testing process. This level of testing is crucial as it involves testing the integrated system software to evaluate if it has aligned with the specified requirements. System testing reviews the entire application, that includes its components and their interactions. It also confirms that the system meets functional and non-functional requirements. It is usually performed by a QA Tester and is done in an environment that simulates the production environment. System Testing also involves end-to-end scenarios that mirror real user workflows to ensure the system performance has no problem with its operations from start to finish. ‘Black-box’ testing is usually involved in this phase as well as the ‘Smoke Testing’ which is a type of software testing that focuses on checking the basic functionality of the system to ensure that the most important features of the system work properly.

‘Acceptance Testing’ is the final level of software testing. It involves evaluating the system to confirm if it meets the business requirements and if it’s ready for deployment. Its main purpose is to verify if the software is acceptable or usable for the end users. It is sometimes referred to as ‘User Acceptance Testing’ or ‘Customer Acceptance Testing’ and is usually performed by a subset of users such as customers and QA Team. Finding defects or issues is not the focus of this level but to ensure that the final product aligns with user needs and business goals.

### Test Types

The following are the test types that will be performed on this project.

‘Functional Testing’ focuses on verifying the functions of the software and ensures the system is aligned to the specified requirements. i.e. – what the system is supposed to do. Functional testing verifies the operation of the software by testing the input and output. It ensures that the system responds correctly to user interactions and processes all the data accurately.

‘Non-Functional Testing’ focuses on verifying the performance of the system and its quality characteristics. It is the testing of ‘how well the system behaves’, meaning, The focus on this testing is to test the speed/response time, compatibility/portability, and reliability of the system. There is no functional testing involved.

‘Black-box Testing’ is a software testing technique which involves evaluating the functionality of the system without necessary knowledge about the internal code structure. It solely focuses on confirming if the functionality of the system is working as intended according to the specified requirements and usually performed by QAs. This method mirrors the real-world environment and ensures that the experience is the same.

‘White-box Testing’ is a software testing technique which involves evaluating the code structure, logic, and flow of the program, allowing the tester to confirm the internal operations in detail. It is the opposite of ‘Black-box Testing’ and usually performed by Developers. It solely focuses on the underlying issues in the internal code structure, ensuring they are uncovered.

### Static Testing

The following static testing activities will take place in order to ensure quality work products

|  |  |  |
| --- | --- | --- |
| **Work Product** | **Testing Activity** | **Project Phase** |
| Software Specification Requirement (SRS) | Formal Inspection | Requirement Analysis |
| Test Plan | Formal Inspection | Requirement Analysis |
| Design Document | Peer Review | Designing |
| Test Cases | Peer Review | Designing |
| Code | Peer Review | Development |
| Final Test Summary Report | Peer Review | Deployment |

### Dynamic Testing

The following dynamic testing activities will take place to ensure code quality and adherence to the requirements.

|  |  |  |
| --- | --- | --- |
| **Work Product** | **Testing Activity** | **Project Phase** |
| Code | Unit Testing, Integration Testing | Development |
| Build or BTC 1 | Smoke Test, Component/feature, Defect Testing | Testing |
| Build or BTC 2 | Defect Testing, Regression Test, Impact Analysis, Confirmation Test, | Testing |
| Full System Test | Operational Acceptance Testing, Contractual Acceptance Testing, Compliance/Regulatory Testing | Deployment |
| User Acceptance Test | Feedback Gathering, User Validation | Deployment |
| Customer Acceptance Test | Feedback Gathering, Customer Validation | Deployment |

## Reports

The following table displays the reports that will be created in the testing phase, the frequency of the reports, the author of the report, and the recipients of the report.

|  |  |  |  |
| --- | --- | --- | --- |
| **Report Name** | **Frequency** | **Author(s)** | **Recipient** |
| Defect Tracking Log | Completion pf each execution build/phase | QA Lead, QA Testers | Project Team |
| Root Cause Analysis Report | Completion of the Customer Acceptance Test Phase | Project Manager, QA Manager, QA Lead, Developers, Business Analyst | Project Tach team |
| Test Incident Reports | Conclusion of every build | QA Lead | QA Team |
| Test Summary Report | End of every phase | QA Manager, QA Lead, Developers | Project Team |

## Defect Tracking

In the following sections, information on how defects are detected throughout the testing phase is detailed. The tools and approach that is used for both static and dynamic testing are included.

### Defect Tracking Tool

The following tools will be used for static testing:

* **Document Defect Log** – This is where we document all the defects that we find during formal inspections and peer reviews.

The following tools will be used for dynamic testing:

* **JIRA** – This tool is commonly used to manage testing activities and to effectively log all the defects without having difficulties tracking them.
* **Mindmap** – We used this tool to easily visualize ideas, tasks, and information by organizing them properly using a flow chart. This tool is where we document all the test conditions or test names.
* **Test Cases** – These are detailed test instructions or step-by-step instructions that are used by QA testers to verify whether the software or feature works as expected.
* **RTM (Requirement Traceability Matrix)** – It is a document usually used for testing to ensure that all test cases are covered. This document links to the requirements throughout the testing and verification process and helps to know whether the product is being tested and developed according to the specified requirements.

### Defect Tracking Approach

For this project we will be following the IEEE 1044 Defect Lifecycle. This process consists of 4 main steps:

* Step 1 – Recognition: Recognition occurs when we observe an anomaly, that anomaly being an incident. If the cause of the anomaly is a defect, then this incident is a failure. This can occur in any phase of the software life cycle.
* Step 2 – Investigation: After recognition, investigation of the incident occurs. Investigation can reveal related issues. Investigation can propose solutions. One solution is to conclude that the incident does not arise from an actual defect. For example, it could be a problem in the test data. (Performed by defect triage team)
* Step 3 – Action: The results of the investigation trigger the action step. We might decide to resolve the defect. We might want to take action to prevent future similar defects. If the defect is resolved, regression testing and confirmation testing must occur. Any tests that were blocked by the defect can now progress.
* Step 4 – Disposition: With action concluded, the incident moves to the disposition step. Here we are principally interested in capturing further information and moving the incident to a terminal (i.e. end) state.

# Estimates and Schedule

The following tables display the overall estimate for hours and cost for the project, and each phase within it also a schedule outline for the entire project and then each phase

An overview of the projects Cost is as follows:

|  |  |
| --- | --- |
| **Total days** | **#47** |
| **Total Schedule Cost** | **$60,020.13** |
| **Total Travel Cost** | **$5,713** |
| **Total Environmental Cost** | **$16,945** |
| **Total Overall Cost** | **$82,678.13** |

An overview of the high-level project dates are as follows:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Overall Project** | **Planning** | **Development** | | **Build 1** | **Fix Build** | **System Test** | **Customer Test** |
| **Start Date** | November 11, 2024 | January 11, 2024 | January 19, 2024 | December 1, 2024 | | December 12, 2024 | December 21, 2024 | December 27, 2024 |
| **End Date** | January 6, 2025 | January 19, 2024 | December 1, 2024 | December 12, 2024 | | December 21, 2024 | December 27, 2024 | January 12, 2025 |

# Resources and Training

In this section, detailed information on the required resources and personnel training is provided.

## Resource Requirements

The following table details the QA team members and their responsibilities on the project.

|  |  |  |
| --- | --- | --- |
| Title | Name | Duties/Responsibilities |
| **QA Manager** | Sophie Patel | Arrange Training for the Junior QA Tester  Training Junior QA for Formal Inspection Process  Training Junior QA for JIRA Functionality  Training Junior QA for TestCollab |
| **QA Lead** | James Ugdang | Initial Analysis and Costing  Develop Initial Schedule  Develop Test Plan  Attend Test Plan Inspection Meeting  Update/Rework Test Plan  Attend Requirement Inspection  Prepare RTM (Requirement Traceability Matrix)  Update estimate in all phases  Create Test Sets during System Testing (3 BTC’s)  Prepare Test Cases/Scenarios  Review Test Cases/Scenarios  Participate in Test Case/Scenario peer reviews  Update/Rework Test Cases/Scenarios  Attend Test Readiness Meetings  Setup Environment  Maintenance Test Cases  Conduct Smoke Tests  Create Test Incident Reports (TIR) in JIRA |
| **QA Tester 1** | Noah MacDonald | Prepare Test Cases/Scenarios  Review Test Cases/Scenarios  Participate in Test Case/Scenario peer reviews  Update/Rework Test Cases/Scenarios  Attend Test Readiness Meetings  Setup Environment  Maintenance Test Cases  Conduct Smoke Tests  Create Test Incident Reports (TIR) in JIRA |
| **QA Tester 2** | Olivia Nguyen | Prepare Test Cases/Scenarios  Review Test Cases/Scenarios  Participate in Test Case/Scenario peer reviews  Update/Rework Test Cases/Scenarios  Attend Test Readiness Meetings  Setup Environment  Maintenance Test Cases  Conduct Smoke Tests  Create Test Incident Reports (TIR) in JIRA  Learn Formal Inspection Process  Learn JIRA Functionality  Learn TestCollab |

## Training Considerations

The following training is needed for any junior test members to ensure they are fully functional before the project starts.

‘Formal Inspection Process’ is a structured, systematic review of the requirements to ensure all of them are testable and feasible. The goal here is to identify and address the defects as soon as possible before they can ruin the quality of the product in the future. The defects are classified by severity (e.g., Critical, Major or Minor). Also familiarizing the ‘ISTQB Standard Terminology’ is necessary for the QA testers to enhance their understanding of the field, communicate more effectively, and improve their overall testing practices.

‘JIRA' is widely used for tracking defects and issues in software development. It’s designed for managing tasks, bugs, user stories, and other project related issue. It also serves as a central hub that allows teams to communicate and help them log all the defects that they find and contribute to discussions regarding defect priorities and project statuses.

‘TestCollab’ is a valuable tool to improved efficiency and enhanced collaboration. This is used to create and organize test cases in a structured manner. QA testers should know about this tool because they can execute manual test cases and log results directly within the platform, which makes it more efficient.

‘XMind’ is a free software that helps QA testers to flesh out test conditions/test case names by creating a mindmap that can help them organize test cases hierarchically, grouping them by requirements which helps in understanding the overall testing scope.

# Deliverables

Detailed information on the products the QA team produces is provided here.

**Estimate-** Created during the planning phase by the QA lead, and approved and reviewed by the QA Manager and Project Manager.

**Schedule-** Created during the planning phase by the QA Lead and reviewed and approved by the QA Manager and Project Manager.

**Test Plan-** Created during the planning phase by the QA Lead then approved by the QA manager.

**Test Cases-** Created during the development phase by the QA lead, QA tester 1 and QA tester 2, then peer reviewed. .

**RTM-** Created during the development phase by the QA Lead to ensure each requirement is thoroughly covered and there is traceability for metrics and reports.

**Mindmap-** Created during the QA team's development phase to group and prioritize test cases by name. It is reviewed and approved by the QA manager.

**Test Defect Logs-** Created during testing of every phase to track any defects found. They are created by the QA team and then updated according to the defects found by every department. Document defect logs will be used for static testing, software defect logs at times for dynamic testing.

**Test Incident Reports-** Created during build 1, 2, and system testing phases by the QA team. It is used to report defects/bugs when using JIRA.

**Test Summary Report-** Created by the QA Lead during at the completion of each test phase of the project to record the project's status and provide the project team with updates.

# Acceptance of Externally Developed Software

During this project, there is no third-party software testing there is no requirement for acceptance criteria on this project.

# Criteria

The following section details the defect severities & priorities, the entry & exit criteria for build 1, build 2, system test, customer acceptance test, and under what condition the testing would be suspended/resumed. During the project, the quality gates are the formal reviews at the end of each phase, and the exit criteria for testing build 1, build 2, and system test.

## Defect Severity Definition

The following table explains the defect severity levels and their effect on the wok project.

|  |  |  |
| --- | --- | --- |
| **Severity** | **Description** | **Example** |
| 1. Fatal | System crash, possibly with loss of data. The test object cannot be released in this form | We’re testing an ecommerce site and when we log in the system goes down, it crashes. |
| 1. Very Serious | Essential malfunctioning; requirements not adhered to or incorrectly implemented; substantial impairment to many stakeholders. The test object can only be used with severe restrictions (difficult or expensive workaround). | Our ecommerce site is working, but when people make purchases the results aren’t automatically showing up in our warehouse system for the products to be packaged and shipped. There is a work around in that we can print a report and email it to the warehouse manager who then has to manually assign the products to employees to find and package for shipment. It takes more time and is far more difficult to manage. |
| 1. Serious | Functional deviation or restriction (“normal failure”); requirement incorrectly or only partially implemented; substantial impairment to some stakeholders. The test object can be used with restrictions. | Say our search field on our ecommerce site is partially working but there is a bug restricting some of the products from appearing. Users are still seeing most products they search for, and if they drill down through the menu items (products -> televisions -> 52inch) they will see all products. |
| 1. Moderate | Minor deviation; modest impairment to a few stakeholders. System can be used without restrictions. | The daily sales reports are working, so are the weekly sales reports, but in the monthly sales reports the tax column isn’t one or two would prefer to see it just to feel more secure. |
| 1. Mild | Mild impairment to few stakeholders; system can be used without restrictions. For example: spelling errors or wrong screen layout. | The shopping cart that always appears at the far right corner appears and inch to the left out of place, or there is a slight spelling error in a product description. |

## Defect Priority Definition

In the following table, the defect priorities levels explain how quickly a defect should be fixed.

|  |  |
| --- | --- |
| **Priority** | **Description** |
| 1. Immediate | The user’s business or working process is blocked or the running tests cannot be continued. The problem requires immediate, or if necessary, provisional repair (->patch release) |
| 1. Next Release | The correction will be implemented in the next regular product release or with the delivery of the next regular product release or with the delivery of the next (internal) test object version. |
| 1. On Occasion | The correction will take place when the affected system parts are due for a revision anyway. |
| 1. Open | Correction planning has not taken place yet. |

## Build Test Correct – Build 1 and Build 2 (Fix Build)

In this section we will define the Readiness to enter and exit the BTC Phase.

### Readiness (Entrance Criteria)

The following readiness criteria will be used for entrance into testing Build 1.

* All features are fully coded.
* Unit test have been executed and any high severity defects corrected.
* Integration testing has been conducted and all high severity issues corrected.
* The requirements have undergone a formal inspection and all defects have been corrected.
* The requirements have been signed off/accepted by the customer.
* The design document has been created, approved and all defects corrected.
* The test conditions have been identified in a Mindmap.
* The Test Plan has been formally inspected and all defects corrected.
* All test cases have been created and peer reviewed.
* The test environment has been setup inclusive of all test data.
* The test sets are created.
* The defect management system is setup.

The following readiness criteria will be used for entrance into testing Build 2 (defect fixes).

* All Fatal and Very Serious defects have been identified as corrected by the development team, and 75% of the Serious, Moderate and Mild as well.
* A test set should be created to retest all corrected issues and corresponding test cases.
* The defects corrected need to be pushed forward to a state of “In Review” by the assigned developers.

### Item Pass/Fail Criteria

Please refer to section 10 above.

### Suspension and Resumption

Under the following circumstances the QA team will suspend and resume testing during Build/BTC testing.

* If the smoke test fails, and/or we encounter a high severity or critical error that significantly impacts or blocks our testing effort QA testing will cease (suspension), the development team will be notified of the impact.
* Testing will only resume once a new build with a fix applied has been delivered (resumption).

### Completion (Exit Criteria)

The following criteria will be used to know when the BTC phase is complete.

* All planned activities for the phase have been successfully completed.
* A phase review or quality gate meeting with all project participants has taken place.
* The entrance criteria for the next phase has been defined and in place.

## System Test

A full system test, or retesting of all previously executed test cases, will be conducted after BTC testing is complete.

### Readiness (Entrance Criteria)

The following readiness criteria will be used for entrance into System Test.

* A maximum of 0 Fatal, 0 Very Serious, 10 Serious, 20 Moderate, 50 Mild defects still open/unresolved.
* A test set should be created to retest all corrected issues and all previously executed test cases.
* The defects corrected need to be pushed forward to a state of “In Review” by the assigned developers.

### Item Pass/Fail

Refer to section 10 above.

### Suspension and Resumption

Under the following circumstances the QA team will suspend and resume testing during System Test.

* If the smoke test fails, and/or we encounter a high severity or critical error that significantly impacts or blocks our testing effort QA testing will cease (suspension), the development team will be notified of the impact.
* Testing will only resume once a new build with a fix applied has been delivered (resumption).

### Completion (Exit Criteria)

The following criteria will be used to determine when the System test phase is complete.

* All planned activities for the phase have been successfully completed.
* A phase review or quality gate meeting with all project participants has taken place.
* The entrance criteria for the next phase has been defined and in place.

## Customer Acceptance Test

Our QA team will assist the customer in performing a series of tests to ensure their expectations have been met and the software is ready to be deployed to production.

### Readiness (Entrance Criteria)

The following criteria will be used to determine if the project should enter into the CAT testing phase.

* A maximum of 0 Fatal, 0 Very Serious, 0 Serious, 0 Moderate, 20 Mild defects still open/unresolved.
* A final Test Summary report must be created and reviewed.
* The project manager must share the results and recommendation related to quality with the customer to determine if we shall enter this phase.

### Item Pass/Fail

Please refer to section 10 above.

### Suspension and Resumption

Testing will suspend and resume under the following circumstances during CAT.

* If the smoke test fails, and/or we encounter a high severity or critical error that significantly impacts or blocks our testing effort QA testing will cease (suspension), the development team will be notified of the impact.
* Testing will only resume once a new build with a fix applied has been delivered (resumption).

### Completion (Exit Criteria)

The following criteria will be used in addition to customer sign off to determine is the CAT phase is complete.

* All planned activities for the phase have been successfully completed.
* A phase review or quality gate meeting with all project participants has taken place.
* The entrance criteria for the next phase has been defined and in place.

## System Acceptance

The software is deemed ready for rollout to production once the customer gives their consent at the conclusion of their Customer Acceptance testing. The internal project team will continue to support the customer throughout their testing period and during deployment to production. The internal project team will also continue to support the customer for a period of 3 months after the deployment date, helping to explain how the functionality works, correcting any defects found, and delivering any agreed upon upgrades. After the 3-month period the maintenance team will take over the responsibility of maintaining the software.

# Risks and Contingencies

The following risks could potentially occur, if they do the contingency noted can resolve the issue to keep the project on track.

|  |  |
| --- | --- |
| **Potential Risk** | **Contingency Plan** |
| Requirements frequently change during testing, leading to problems like delays and confusion, and some of the functionalities might be missed. | We will conduct a retrospective with stakeholders to clarify any confusion and changes in requirements and implement a change management process to ensure that any type of changes is handled for impact on test cases. |
| The test environment may sometimes not be available due to some issues like configuration and maintenance. | Having backup environments or virtual machines to continue testing without causing delays. |
| The site may encounter issues like crashes and slow down because of high traffic, especially during busy seasons. | We will conduct performance and load testing to check for any bottlenecks, identify them in advance, and optimize code and infrastructure based on what we find during testing. |
| Not enough invalid test data can lead to results that are not accurate, and the defects can be missed. | A reliable test data management process must be set up to create and maintain more realistic data sets. |
| Many defects are identified during testing, which may cause delays and overwhelm the QA team. | Only focus on blocking and high-priority issues, creating a plan for defect prioritization and triaging. |

# Approvals

Date:

Approved by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Role of Approver(s):

# Appendix A – Full Estimate

The following is the Full Estimate for the QA test efforts on this project:

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# Appendix B - Schedule

The following is the Full Schedule for the QA test efforts on this project:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated