ACCELERATE
CONSULTING

Phase 3: Construction

5.08 Test Suite

Version 2.0

Revision History

Date	Version	Description	Author
2023-04-11	2.0	Updated Test Suite	Cathy Zhao
2023-03-27	1.0	Test Suite	Cathy Zhao

Table of Contents

1. Introduction

1.1 Purpose

The purpose of the test suite is to ensure that the web portal for C3 is reliable, meets the requirements of the stakeholders, and functions as expected, which includes the login, claims report and work order form filling, reports and analytics form, and logout functionality. To identify defects and to uncover defects or bugs in the software application or system. By running different test cases, the tester can identify areas where the web portal is not functioning as expected. To ensure software quality, that the software meets the desired level of quality. By running different types of tests, such as functional tests, performance tests, and security tests, the tester can identify areas where the web portal needs improvement. This document will discuss the benefits of the web portal and the risk associated with deploying the software application or system. By identifying and fixing defects early in the development cycle, the risk of costly and time-consuming rework is reduced. The deliverable will highlight ways in which the web portal complies with industry standards, regulatory requirements, and customer requirements.

1.2 Scope

The scope would be that the portal provides a positive user experience and meets the needs of the dealership's customers and employees.

- User interface (UI) testing: ensure that the portal's layout and design are visually appealing and easy to navigate. This would include functional testing of features such as logging in and out, filling in claims and work order forms
- Performance testing: ensure that the web portal operates quickly and efficiently. This would include stress testing the portal to identify any performance bottlenecks or scalability issues.
- Security testing: ensure that the web portal is secure and protects sensitive customer and dealership data from unauthorized access or theft. This would include testing features such as user authentication, data encryption, and access controls.
- Integration testing: ensure that the web portal integrates seamlessly with other software applications and systems used by the dealership, such as inventory management, CRM, and finance and accounting software.
- Usability testing: ensure that the web portal is easy to use and meets the needs of the dealership's customers and employees. This would include testing the portal's features and functionality from the perspective of different user personas, such as sales staff, service technicians, and customers.

1.3 Definitions, acronyms, and abbreviations

CSV - comma separated variables

1.4 References

https://www.ibm.com/docs/en/elm/6.0?topic=sections-test-suite-template-reference
https://www.parasoft.com/blog/how-to-write-test-cases-for-software-examples-tutorial/
https://learn.microsoft.com/en-us/previous-versions/azure/devops/test/mtm/plan-manual-tests-with-microsoft-test-manager?view=tfs-2017&viewFallbackFrom=azure-devops
https://katalon.com/resources-center/blog/automation-testing-tools
https://www.selenium.dev/

1.5 Overview

The document will include an introduction, test suite objectives, test suite scope, test strategy, test cases, test data, test environment, test schedule, test results, and a conclusion. The introduction provides an overview of the document and the car dealership organization web portal. The test suite objectives section outlines the objectives of the test suite, and the scope section defines the specific functional areas of the web portal that will be tested. The strategy section outlines the types of testing that will be performed and the tools and methodologies that will be used. The test cases section provides a detailed description of each test case, while the test data section describes the test data that will be used to execute the test cases. The environment section describes the test environment that will be used, while the schedule section provides a timeline for executing the test suite. The results section describes the results of the test suite, including any defects or issues that were identified, and the corrective actions that were taken. Finally, the conclusion section provides a summary of the test suite and recommendations for future testing.

2. Test Strategy

2.1 Test Approach and Scope

The tests will include a combination of different testing methodologies such as functional testing, performance testing, security testing, and usability testing.

1. Functional testing involves testing the different functionalities and features of the web portal, making sure they meet the expected requirements and specifications.
2. Performance testing checks the responsiveness and speed of the web portal, ensuring it can handle the expected traffic and load.

3. Security testing checks the web portal's security features, such as login authentication, user access controls, and data protection measures.
4. Usability testing checks the overall user experience of the web portal, ensuring it is user-friendly, easy to navigate, and provides a satisfactory experience for users. Surveys and interviews would be the main source of information for the tests, as users are asked about their experience when they visit the dealerships.

The organization will use a combination of testing tools to ensure the software is on par with stakeholder expectations. Various testing tools and techniques such as automated testing tools, manual testing techniques, and usability testing techniques like user interviews or surveys would be utilized. Automated testing tools will be utilized, such as Azure Test Plans. Manual testing would also be used throughout the process to pick up bugs and enhance the system as the developers progress.

2.2 Test Data

The test data portion of the test automation strategy would include the data used for testing the various functionalities of the web portal. This would typically include a range of data for different scenarios, such as:

- Valid user data: Test data that represents a valid user, including valid login credentials and personal details.
- Invalid user data: Test data that represents an invalid user, including invalid login credentials and personal details.
- Error message data: Test data that represents various error messages that the user might encounter, such as invalid input data, incorrect login credentials, etc.
- Test scenarios data: Test data that represents various test scenarios, such as positive and negative test cases, edge cases, stress testing, etc.

The test data can be stored in a separate file, such as a CSV or Excel file, and accessed by the automation scripts as needed. The test data is comprehensive, accurate, and covers all the possible scenarios that could occur on the web portal. Additionally, the test data will be updated and maintained regularly to ensure that it stays relevant and up-to-date with any changes to the web portal's functionalities.

2.3 Execution

The test scripts will be executed on a regular basis to ensure that the web portal is functioning correctly. The frequency of the test runs will depend on the frequency of changes to the web portal.

2.4 Maintenance

The scripts and framework will be regularly updated to ensure that they remain relevant and effective.

2.5 Reporting

The test strategy will include reporting mechanisms to provide information on the performance of the web portal, including the results of the tests. The reports will be easy to understand and use.

2.6 Test Environment Requirements

Requirements that are needed for the test environment to function properly.

1. Operating System: Windows 10 or 11
2. Browsers: Chrome 95, Firefox 95, Safari 15, Edge 95.
3. Test Environment Setup: Python 3.11, Django 3.2.
4. Network: Internet connectivity with a minimum speed of 10 Mbps. Firewall should allow traffic on ports 80 and 443.
5. Hardware: Minimum 8 GB RAM, Intel i5 or equivalent CPU, 50 GB free disk space.
6. Test Data: Data should be stored in a local instance that will be refreshed before every test run. Test data should include valid and invalid credentials for logging in and product details.

2.7 Test Schedule

Test Phase	Timeline
Login and Authentication Testing	Day 1-2
Claim Entry Page Testing	Day 3-4
Sales Page Testing	Day 5-6
Reports and Analytics Testing	Day 7-8
User Acceptance Testing	Day 9-10

Integration Testing	Day 9-10
Regression Testing	Day 11-12
Security Testing	Day 13

2.8 Test Metrics

Test metrics are important to track and report on during the testing process.

- Test case execution status: This metric tracks the number of test cases that have been executed and their current status (pass/fail/pending). This metric can be used to determine the progress of the testing effort and to identify any areas that may need additional attention.
- Defect density: This metric calculates the number of defects found per unit of code or functionality. This can be useful in identifying areas of the application that may be particularly problematic and require additional attention.
- Test coverage: This metric tracks the percentage of the application that has been tested. This can help identify areas of the application that may have been overlooked and require additional testing.
- Test case efficiency: This metric tracks the amount of time it takes to execute a test case. This can be useful in identifying test cases that may be too time-consuming and may need to be revised or removed from the test suite.
- Test case prioritization: This metric ranks test cases by their priority based on the risk they pose to the application. This can help ensure that the most critical areas of the application are tested thoroughly and that any issues are identified and addressed quickly.
- Test cycle time: This metric tracks the amount of time it takes to complete a testing cycle. This can help identify areas of the testing process that may be causing delays and require improvement.

By tracking and reporting on these test metrics, C3 can ensure that their testing efforts are effective and efficient and that any issues are identified and addressed in a timely manner.

3. Test Plan

3.1 Test Objectives

- Verify that all pages on the web portal are responsive and display correctly on different screen sizes and devices.

- Ensure that all functionalities on the web portal are working as expected
- Test the performance of the web portal under different user loads and traffic scenarios.
- Check the security of the web portal and ensure that it is protected from unauthorized access and data breaches.
- Verify that the web portal is compliant with all relevant regulations and standards, such as accessibility requirements and data protection laws.
- Test the integration between the web portal and any third-party services, such as payment gateways and inventory management systems.
- Verify that the web portal provides a seamless user experience and is easy to navigate for both new and returning users

3.2 Test Deliverables

- Test plan document: This document outlines the overall approach and objectives of the testing process.
- Test suite document: This document includes the collection of tests, test frameworks and test-log info.
- Test case document: This document outlines the individual test cases that will be executed as part of the testing process.
- Test scripts: These are the automated scripts used to execute the test cases.
- Test data: This includes the input data used in executing the test cases.
- Test reports: These reports provide information on the test execution status, including any defects found during the testing process.

3.3 Test Risks

Some risks could include:

- Integration Issues: The web portal may have integration issues with third-party applications or systems, which could affect the overall functionality of the portal.
- Security Issues: The web portal may be vulnerable to security threats, such as hacking or data breaches, which could lead to the compromise of sensitive information.
- Performance Issues: The web portal may experience performance issues, such as slow load times or crashes, which could result in a negative user experience and decreased customer satisfaction.
- Usability Issues: The web portal may have usability issues, such as confusing navigation or unintuitive interfaces, which could lead to user frustration and dissatisfaction.
- Compatibility Issues: The web portal may have compatibility issues with certain devices or browsers, which could limit its accessibility and usability for some users.

3.4 Test Dependencies

Deliverables that would need to be in place before the testing occurs would include:

- Development of the web portal: The development of the web portal needs to be completed before testing can begin.
- Test environment: A suitable test environment needs to be created, including appropriate hardware, software, and network configurations.
- Test data: Test data should be available before testing begins, including test cases, test scripts, and test scenarios.
- Test automation tools: If automation is being used, the test automation tools must be set up before testing begins.
- User acceptance: The test process may be affected by user acceptance criteria and timelines, so this needs to be considered in the test dependencies.

4. Test Suite Strategies

4.1 Smoke Test Suite

The smoke test suite is a set of tests that are executed to verify the basic functionalities of the system. The purpose of a smoke test suite is to ensure that the system is stable enough to proceed with further testing. The smoke test suite includes a set of high-level test cases that are executed after each build or release to ensure that the basic functionalities of the system are working as expected. The smoke test suite would include the following test cases:

- Login Test Case: This test case verifies that the user is able to log in to the system with valid credentials.
- Logout Test Case: This test case verifies that the user is able to log out of the system.

The smoke test suite helps to identify any major issues with the system early on in the testing process and ensures that the system is ready for further testing.

4.2 Integration Test Suite

Integration testing focuses on verifying the interactions between different components or systems of an application. It can be used to test the interactions between different modules or components of the portal, such as the user interface, the database, and the back-end services.

The Integration Test Suite for the C3 web portal would include test cases that focus on verifying the following:

Integration between the user interface and the back-end services:	This would include test cases that simulate user actions ● Clicking on a button

	● Entering data into a form ● Verifying that the corresponding service is called and the correct data is returned
Integration between different back-end services:	This can include test cases that verify that data is correctly passed between different services and that the services can communicate with each other.
Compatibility with different browsers and devices:	This can include test cases that verify that the web portal functions correctly on different browsers and devices.
Data integrity:	This can include test cases that verify that data is correctly stored and retrieved from the database, and that there are no data inconsistencies or errors.
Performance and scalability:	This can include test cases that verify that the web portal can handle a large number of users and transactions without performance issues or downtime.

The Integration Test Suite should be executed after the Smoke Test Suite and before the Functional Test Suite, as it helps to identify and resolve any issues related to the integration between different components of the portal.

4.3 Functional Test Suite

A functional test suite for C3 would include test cases that verify the functionality of the web portal according to the requirements and specifications. The functional test suite would cover different areas of the web portal, such as:

- User management: This would cover test cases that verify the registration, login, and logout functionality of the web portal. It would also cover test cases that verify the user profile management functionality.
- Car details: This would cover test cases that verify the car details functionality of the web portal. This would include verifying that users can view the details of a particular car, such as the specifications, features, and price.
- Reports and Analytics management: This would cover test cases that verify the report and analytics functionality of the web portal. This would include verifying that the reports are updated in real-time and that administrators can view them.

The functional test suite would ensure that the web portal meets the functional requirements and specifications and that it performs as expected.

4.4 Security Test Suite

A Security Test Suite for the C3 web portal would include a set of test cases that ensure the security of the web portal. This suite would cover various aspects of security such as authentication, authorization, confidentiality, integrity, and availability.

- Authentication Testing: This involves testing the authentication mechanism of the web portal to ensure that only authorized users can access it. Test cases would include verifying the strength of the passwords, testing password reset functionality, and ensuring that the session timeouts after a period of inactivity.
- Authorization Testing: This involves testing the authorization mechanisms of the web portal to ensure that users can access only those pages or functionalities for which they have been granted access. Test cases would include verifying role-based access control, testing access to restricted pages, and ensuring that unauthorized users are redirected to the login page.
- Input Validation Testing: This involves testing the web portal for input validation vulnerabilities that may be exploited by attackers. Test cases would include testing for SQL injection, cross-site scripting (XSS), and buffer overflow vulnerabilities.
- Confidentiality Testing: This involves testing the web portal to ensure that confidential information such as user credentials and credit card details are protected. Test cases would include verifying that HTTPS is being used, testing for sensitive data exposure vulnerabilities, and ensuring that no sensitive information is being stored in cookies.
- Integrity Testing: This involves testing the web portal to ensure that the data being transmitted or received is not being tampered with. Test cases would include verifying that digital signatures are being used, testing for man-in-the-middle (MitM) attacks, and ensuring that data is not being modified during transmission.
- Availability Testing: This involves testing the web portal to ensure that it is available to authorized users at all times. Test cases would include testing for denial of service (DoS) attacks, verifying that the web portal is resilient to high traffic, and ensuring that failover mechanisms are in place in case of a server outage.

By including a Security Test Suite, C3 can ensure that the web portal is secure and protected against potential security threats.

5. Test Suites

Each test suite groups related test cases together, based on the area of the system being tested. This allows for more efficient testing and easier organization of test results.

The test suites for the project are as follows:

5.1 Login and Authentication Test Suite:

Test Case 001: Login with Valid Credentials
Test Case 002: Login with Invalid Credentials
Test Case 003: Login with Blank Credentials
Test Case 004: Password Reset
Test Case 005: Authentication Timeout
Test Case 006: Account Lockout

5.2 Claim Entry Page Test Suite:

Test Case 001: New Claim Entry Page
Test Case 002: Status Update Approved
Test Case 003: Status Update Rejected
Test Case 004: Missing Required Field
Test Case 005: Repair Shop and Customer Information Format
Test Case 006: Problem Adding and Removing
Test Case 007: Claim Page Image Upload
Test Case 008: Claim Amounts
Test Case 009: Claim Expanded View

5.3 Sales Page Test Suite:

Test Case 001: Work Order Form
Test Case 002: Work Order Form Hyperlink

5.4 Reports and Analytics Test Suite:

Test Case 001: Generate Sales Report
Test Case 002: Export Sales Report
Test Case 003: View Claim Analytics

5.5 User Acceptance Test Suite:

Test Case 001: Login Functionality
Test Case 002: Full Claims Access
Test Case 003: Full Sales Access

6. Test Execution

6.1 Test Execution Results

The Test Execution Results presents the outcomes of the test execution. It includes the following information:

Test case ID
Test case description
Test result (pass/fail)
Test environment configuration
Test execution date and time
Test duration
Test execution logs

The Test Execution Results section helps the testing team to track the progress of the testing efforts, identify the test cases that passed and failed, and investigate any defects that were identified during the test execution. The results will be presented in a tabular format, with each row representing a single test case and each column providing the relevant details of the test execution. This information is critical in determining the quality of the web portal and making decisions about whether to release it to production or not.

6.2 Test Incident Management

If an incident is identified during testing, it should be reported immediately to the project manager or test manager. The incident should be logged in the incident management system using the following information:

- Incident ID
- Incident description
- Date and time of incident
- Severity (Low/Medium/High)
- Priority (Low/Medium/High)
- Test case ID
- Steps to reproduce the incident
- Expected result
- Actual result
- Environment details (e.g. browser, OS, etc.)
- Screenshots (if applicable)

The incident should be assigned to the appropriate developer for investigation and resolution. The developer should update the incident in the incident management system with the following information:

- Status (Open/In Progress/Closed)
- Root cause analysis
- Corrective action taken
- Testing required (Yes/No)
- Date and time of resolution
- Tester to verify the resolution

The tester should verify that the incident has been resolved and update the incident in the incident management system with the following information:

- Verification status (Pass/Fail)
- Date and time of verification
- Comments (if any)

If the incident is not resolved within the specified timeframe, it should be escalated to the project manager or test manager for further action.

7. Test Summary Report

A Test Summary Report summarizes the results of the testing process. It provides a comprehensive view of the testing effort and the quality of the software being tested. The key components of a Test Summary Report in a Test Suite Document for the C3 Web Portal:

- Introduction: This section includes a brief overview of the testing process, the objectives of testing, and the scope of the testing effort.
- Test Results: This section provides a summary of the test results, including the number of test cases executed, the number of test cases passed and failed, the severity of the defects found, and any open issues.
- Test Coverage: This section provides an overview of the test coverage, including the percentage of requirements, functions, and code covered by the tests.
- Defect Metrics: This section provides information on the defects found during testing, including the number of defects found, the severity of the defects, the status of the defects, and the time taken to fix the defects.
- Risk Assessment: This section provides an assessment of the risks associated with the software being tested, including the likelihood and impact of the risks.
- Recommendations: This section provides recommendations for improving the quality of the software being tested and the testing process itself.
- Conclusion: This section includes a summary of the overall testing effort and the quality of the software being tested, as well as any final thoughts or recommendations.

A sample Test Summary Report for C3 Web Portal:

Test Summary Report
C3 Web Portal Testing

Introduction:

This report provides an overview of the testing process and the results of testing for the C3 Web Portal. The objective of testing was to ensure that the web portal meets the functional and performance requirements.

Test Results:

A total of 500 test cases were executed, out of which 450 test cases passed and 50 test cases failed. The defects found were classified as Critical (5), High (10), Medium (15), and Low (20). All the high and critical defects have been resolved, while medium and low defects have been assigned to future releases. There are no open issues.

Test Coverage:

The testing covered 100% of the functional requirements, and 85% of the code was covered. All critical paths of the system were covered, and more than 90% of the test cases were automated.

Defect Metrics:

A total of 50 defects were identified, with 5 classified as critical, 10 as high, 15 as medium, and 20 as low. The average time taken to fix a defect was 3 days.

Risk Assessment:

The risks associated with the software being tested were assessed as low, with no significant risks identified.

Recommendations:

1. Further automation of the test cases should be done to increase the testing coverage and reduce the time taken to execute tests.
2. The testing process should be reviewed to ensure that it is efficient and effective.
3. Additional performance testing should be conducted to ensure that the system can handle the expected load.

Conclusion:

The testing effort for C3 Web Portal was successful, with no major issues identified. The testing coverage was comprehensive, and the software meets the functional and performance requirements. Overall, the testing process was efficient and effective, and the testing team recommends that the software be released for production.```

8. Conclusion

The document should cover all aspects of the testing process, including the test strategy, test objectives, test scope, test techniques, test environment requirements, test data, test schedule, test metrics, test deliverables, test risks, test dependencies, test execution results, and incident management.

By following the guidelines and practices outlined in the document, the web portal should meet all the necessary requirements, including functional, security, and performance aspects, and that the end-users have a positive experience using the web portal. The test summary report can provide valuable insights and feedback on the testing process, allowing the team to identify areas for improvement and refine their testing approach in future projects. Overall, a comprehensive and well-executed test suite document can contribute significantly to the success of the car dealership organization's web portal.