[https://www.guru99.com/images/logo/logo.png](https://www.guru99.com/)

D7 Auto Service Center Web-App

**Test Strategy**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Description |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Scope 3](#_Toc144496854)

[2. Test Approach 4](#_Toc144496855)

[3. Test Environment 5](#_Toc144496856)

[4. Testing Tools 6](#_Toc144496857)

[5. Release Control 7](#_Toc144496858)

[6. Risk Analysis 9](#_Toc144496859)

[7. Review and Approvals 10](#_Toc144496860)

# Scope

A testing strategy is more than just a plan; it's a compass that guides a project toward quality and success. This all-around method spells out the most important parts of effective and efficient testing. There are two significant concerns:

This testing plan will be looked over by Jakerson Bermudo. Their ideas make sure that the plan is right, fits with industry standards, and can be done within the scope of the project.

While Wilkins Caducio has the authority to approve and complete this testing plan. They use their knowledge of the project and attention to detail to make sure the strategy fits our project goals correctly.

**Testing Strategy Document**

Project Name: D7 Auto Service Center Web-App

Testing Start Date: [Start Date]

Testing End Date: [End Date]

Test Manager: Wilkins Caducio

Version: 1.0

1. Test Planning Phase (date)
2. Test Environment Setup (date)
3. Test Case Design and Preparation(date)
4. Test Execution Phase (date)
5. Defect Reporting and Tracking (date)
6. Test Completion and Reporting (date)
7. Test Closure Phase (date)

# Test Approach

In web application development, the test approach acts as a strategic road map to make sure the program is dependable, safe, user-friendly, and performs effectively in a variety of user scenarios. It helps to develop a product that satisfies user and business needs while addressing the particular difficulties faced by web applications.

1. **Unit Testing:**

* Conducted by the development team.
* Focuses on testing individual components, functions, and modules.
* Ensures that each component performs as expected and meets specifications.

1. **Functional Testing:**

* Validates the functional requirements of the application.
* Tests features like user registration, task creation, collaboration, and reporting.
* Focuses on testing user interactions and workflows.
* Covers positive and negative scenarios to ensure feature reliability.

1. **Usability Testing:**

* Evaluates the user interface's intuitiveness and user-friendliness.
* Tests user interactions, navigation, and overall user experience.
* Addresses design and usability issues to enhance user satisfaction.

1. **User Acceptance Testing (UAT):**

* Conducted by end users to validate the application's readiness for release.
* Tests real-world scenarios and user workflows.
* Gathers user feedback to ensure the application meets business requirements.

Defects will be tracked using a designated defect tracking tool. Each defect will be categorized, prioritized, assigned, and tracked until resolution. Regular testing progress reports will be generated, detailing test coverage, test cases executed, defects found, and overall application quality. Any critical issues or roadblocks will be highlighted for immediate attention.

# Test Environment

The team will be utilizing two standard testing environments which will be used to conduct a variety of testing activities that will evaluate the functionality of the web application. The activities that will be conducted will revolve around testing the system's functionality in terms of the following:

1. **Localhost** - which is intended for the development team's use wherein the testing team will do the necessary testing for the updated functionalities depending on the updated requirements. In using localhost as the testing environment, the testing team will need the updated source code from the development team and then, it will be run on the localhost environment of the testing team specifically the XAMPP or another similar platform. The setup enables the testing team to assess and validate the functionality of the web application in a controlled and safe environment prior to broader testing and deployment.
2. **SocitCloud**, - The socitcload is the temporary hosted environment of the development team which intended to test the web application in the state of hosting the actual deployed system.

The team will also employ Selenium-IDE to execute the program within the browser based on recorded scripts of actions and associated data. By using this integrated development environment (IDE), the testing team will assess whether the following development team requirements are met with the intended level of quality.

# Testing Tools

The testing team will utilize testing tools for different stages of the software testing life cycle. These testing tools will help the testing team to perform different testing procedures, including User Acceptance Testing, Functional Testing, Performance Testing, and other activities that are necessary for the software testing.

The testing team will be focusing on functional testing by using the testing tools of TestLink which is an open-source test management tool. TestLink is use for the testing teams to organize their test plans, test strategies, as well as their test executions which aim to provide a centralized system that facilitates collaborations throughput the software testing life cycle that making it easier to track progress and maintain the quality of software testing.

Each member of the team will have access to TestLink, enabling them to efficiently create, organize, and manage test cases according to the test plan. By using this tool, the process of execution in the software testing life cycle will be simplified, and it will allow the team to record results and monitor progress effectively.

In addition, the testing team will use the Selenium, an open-source tool for doing the test automation that are suitable for automating for web applications. In particular, the team will utilize the Selenium-IDE (Integrated Development Environment) which is the plug-in version which give GUI (Graphical User Interface) using the browser specifically in Mozilla Firefox and Google Chrome that will record the user actions as well as the input values. This IDE will help the testing team to validate if the specific functions are work based on the requirements.

# Release Control

As an essential component of our Testing Strategy, we've implemented a Release Control process. This process includes a comprehensive Release Management Plan that holds a detailed version history. The primary objective of this plan is to ensure a thorough and effective test execution process for all modifications encompassed within a specific release.

**Release Management Plan**

In order to manage and monitor software releases effectively, our Release Management Plan plays a crucial role. It encompasses the following key elements:

1. **Version History**

We maintain a meticulous record of versions released, outlining essential details such as version numbers, release dates, descriptions of modifications made, and the corresponding testing outcomes.

1. **Change Documentation**

Each modification introduced in a release is carefully documented. This documentation provides a clear understanding of the changes and categorizes them based on their significance, criticality, and scope.

1. **Release Testing Scope**

Our plan defines the specific scope of testing for every release. We focus on key modules, functionalities, and user scenarios that require rigorous testing to meet project priorities.

1. **Test Coverage Strategy**

We've outlined a strategy for comprehensive test coverage, which encompasses unit, integration, functional, performance, and regression testing. Our test cases are designed to effectively validate modifications introduced in the release.

1. **Test Environment Setup**

The plan outlines the process of setting up testing environments to mirror the configuration of the production environment. This step ensures consistency in testing conditions.

1. **Test Execution Schedule**

We've established a well-defined schedule for executing tests, ensuring alignment with the release timeline. This approach optimizes resource utilization and ensures timely feedback.

1. **Defect Management Process**

Our plan details the process for reporting, tracking, and addressing defects. It provides guidelines for effectively prioritizing and resolving defects to maintain efficient testing cycles.

1. **Regression Testing Approach**

We've laid out our approach for regression testing, which confirms that existing functionalities remain intact amidst new changes. This process combines both automated and manual testing methods as per feasibility and coverage requirements.

**Benefits of Release Control:**

1. Assures a comprehensive and meticulous testing process for each release.
2. Maintains a detailed record of modifications and corresponding testing outcomes.
3. Facilitates communication and coordination among development, testing, and deployment teams.
4. Mitigates the risk of critical defects reaching production environments.
5. Supports the consistent delivery of high-quality software that aligns with project goals and user expectations.

# Risk Analysis

Indicated below are the possible risk that the team might encounter during the testing phase of the system D7 Automotive Services Web-Application:

|  |
| --- |
| 1. Resource Constraints: |
| **Description:** Testing resources might be incapable of providing proper testing environments that might delay the testing process |
| **Impact:** Testing delays, and reduced quality. |
| **Mitigation:** Research and allocate time in finding possible testing resources as a back-up option if the given tools aren’t functioning as planned. |

|  |
| --- |
| 1. Communication Gaps |
| **Description:** Misunderstanding might occur due to poor communication. |
| **Impact:** Missed requirements, and inadequate testing coverage |
| **Mitigation:** Establish regular status meetings, and the use of appropriate communication channels. |

|  |
| --- |
| 1. Unforeseen Technical Challenges |
| **Description:** Unexpected and random technical issues might occur during the testing process. |
| **Impact:** Testing schedule delays |
| **Mitigation:** Assessing the testing environment and tools regularly to ensure stability before use. |

|  |
| --- |
| 1. Incomplete Requirements |
| **Description:** Missing or inadequate requirements may lead to a lack of test coverage |
| **Impact:** Test cases might not cover all functionality which might lead to some defect not being fixed. |
| **Mitigation:** Engage with the stakeholders and the dev team to clarify requirements. |

# Review and Approvals

Reviewed and approved by the following:

|  |  |  |
| --- | --- | --- |
| Name | Signature | Date |
| **Wilkins V. Caducio**  (Project Manager for testing team.) |  |  |
| **Jakerson B. Bermudo**  (Business Analyst) |  |  |
| **Andre Viernes**  (Stakeholder for development team) |  |  |
| **Jose Eugenio L. Quesada**  (Subject Adviser) |  |  |