Barangay South Signal Village Web-App

Test Plan

Prepared by:

Aspeli, Mabelle T.

Villamin, Sofia Emmanuelle R.

August 24, 2023

**Table of Contents**

[Revision History 3](#_Toc146695315)

[Introduction 4](#_Toc146695316)

[Purpose 4](#_Toc146695317)

[Project Overview 4](#_Toc146695318)

[Audience 5](#_Toc146695319)

[Test Strategy 6](#_Toc146695320)

[Test Objectives 6](#_Toc146695321)

[Test Assumptions 7](#_Toc146695322)

[Test Principles 8](#_Toc146695323)

[Data Approach 10](#_Toc146695324)

[Scope and Levels of Testing 11](#_Toc146695325)

[Exploratory 11](#_Toc146695326)

[Functional Test 11](#_Toc146695327)

[User Acceptance Test (UAT) 12](#_Toc146695328)

[Test Deliverables 13](#_Toc146695329)

[Milestone List 15](#_Toc146695330)

[Test Effort Estimate 17](#_Toc146695331)

[Test Areas and Specifications 18](#_Toc146695332)

[Features to be Tested (Use Cases) 18](#_Toc146695333)

[Execution Strategy 19](#_Toc146695334)

[Entry and Exit Criteria 19](#_Toc146695335)

[Test Cycles 23](#_Toc146695336)

[Validation and Defect Management 24](#_Toc146695337)

[Test Metrics 26](#_Toc146695338)

[Defect Tracking and Reporting 27](#_Toc146695339)

[Test Management Process 28](#_Toc146695340)

[Test Management Tool 28](#_Toc146695341)

[Test Design Process 28](#_Toc146695342)

[Test Execution Process 30](#_Toc146695343)

[Test Risks and Mitigation Factors 32](#_Toc146695344)

[Role Expectations 33](#_Toc146695345)

[Project Management 34](#_Toc146695346)

[Test Planning (Test Lead) 34](#_Toc146695347)

[Test Team 35](#_Toc146695348)

[Development Team 40](#_Toc146695349)

[Test Environment 42](#_Toc146695350)

[Approvals 43](#_Toc146695351)

# **Revision History**

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

**Approvers’ List**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Role** | **Approver/Reviewer** | **Approval/Review Date** |
| Wilkins Caducio | Project Manager of Team Developmentality |  |  |
| Jose Eugenio Quesada | Course Professor |  |  |

**Reference Document**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Document Name** |
|  |  |  |
|  |  |  |

# **Introduction**

## **Purpose**

The primary objective of this test plan is to create a thorough framework for comprehensively evaluating the Barangay South Signal Village Web Application from the customer’s perspective. This framework encompasses a detailed outline of the testing processes, covering various aspects such as test procedures, test execution methodologies, and overall test management. The overarching goal is to ascertain that the application's functionality and performance align seamlessly with the client's overarching business requirements. Additionally, the plan incorporates a comprehensive strategy for documentation, encompassing the creation and maintenance of essential records, including but not limited to test cases, test strategies, and the expected results throughout the entirety of the testing phase. This documentation not only aids in maintaining clarity and traceability but also serves as a critical resource for ensuring the accuracy and completeness of the testing efforts.

## **Project Overview**

Barangay South Signal Village, situated within the vibrant city of Taguig in the Philippines, faces significant challenges stemming from its reliance on manual-based operations and extensive paper records. Recognizing the need for modernization and efficiency improvements, the village is exploring the implementation of a comprehensive document management system. This proposed system envisions a transformative shift towards digitizing transactions and establishing a secure and accessible database hosted on servers or cloud platforms. Such a transition promises to streamline administrative processes, reduce paper consumption, and enhance the overall effectiveness of barangay operations. Currently, the barangay relies heavily on its Facebook page as a primary means of information dissemination, highlighting the urgency of allocating resources to integrate its existing processes into a cohesive and digital framework, thereby ensuring more efficient service delivery and improved communication with its residents.

## **Audience**

This test plan is primarily tailored to cater to the specific needs and interests of two key audiences: the testers and the technical writers involved in the project. For testers, this document serves as a crucial reference point that provides a comprehensive understanding of the testing processes, methodologies, and objectives, enabling them to conduct testing activities effectively and in accordance with the predefined guidelines. It offers insights into the scope of testing, expected outcomes, and the strategies to be employed during the testing phase. Simultaneously, the test plan also addresses the needs of technical writers, offering them valuable context and information essential for crafting accurate and informative documentation. By aligning their understanding with the testing procedures and objectives, technical writers can produce user manuals, help documents, and other materials that precisely convey the software's functionality and usage, contributing to a seamless and well-documented end-user experience. Thus, this test plan ensures that both testers and technical writers have the necessary resources to carry out their respective roles effectively, enhancing the overall quality and completeness of the project deliverables.

# **Test Strategy**

## **Test Objectives**

This web application's main goal is to give Barangay South Signal residents and officials a centralized, effective platform, simplifying information administration and enabling smooth transactions for each resident. Through the utilization of this application, citizens can obtain essential information, make requests, and communicate with local government agencies in a more structured and intuitive way. In addition, it gives barangay officials the means and instruments to handle resident data, reply to queries, and improve community relations. This main goal highlights the dedication to enhancing the general standard of living for locals by utilizing technology to develop a more user-friendly and accessible interface for obtaining necessary services and information inside the barrio.

After the extensive testing phase, we have two goals in mind for the web application. Initially, our goal is to provide a reliable and fully functional online application that the target barangay's citizens may easily access. This entails making certain that every feature and function operates without a hitch and gives locals a dependable platform for information access and effective transaction processing. Second, the application is intended to be a long-lasting and durable fix. Long-term operation is our goal, and we have procedures in place for continuous testing and error/bug identification. This attention to ongoing maintenance and improvement demonstrates our commitment to provide a solution that not only satisfies current needs but also continues to be flexible and strong in the face of upcoming obstacles and changing specifications. Our objective is to offer a trustworthy, long-lasting resource that will continue to help the community for a very long time.

## **Test Assumptions**

Below are the key assumptions that underpin our testing strategy and inform critical aspects of the testing process:

* 1. The stability of the testing environment: The consistent availability and reliability of the hardware, software, and network infrastructure required for testing purposes, ensuring a dependable foundation for conducting tests and producing accurate results.
  2. The availability of required resources: The essential testing tools, hardware, software licenses, and skilled personnel will be accessible as needed throughout the testing phase, ensuring the efficiency and effectiveness of the testing efforts.
  3. The accuracy and accessibility of test data anticipates: The necessary datasets will be both precise in their representation of real-world scenarios and readily available for testing purposes, enabling thorough and representative assessments of the application's functionality and performance.
  4. The timely resolution of critical issues: The team is expecting that any defects or problems discovered during testing will be addressed promptly and effectively to maintain the testing schedule and ensure the application's reliability.
  5. The adherence to project timelines presupposes: All project activities, including testing phases, will proceed according to the defined schedule, ensuring that testing efforts align with the broader project's timeline and objectives.
  6. The expertise and availability of the testing team: The possess the requisite skills, knowledge, and availability to execute testing activities effectively and efficiently, ensuring that testing objectives are met.
  7. The effectiveness of security measures: Implemented security protocols, such as firewalls, encryption, and access controls, can safeguard the application and its data against potential threats and vulnerabilities, bolstering the overall security posture.
  8. The continued functionality of third-party integrations: The external services, APIs, or components, which the application relies on, will maintain their reliability and compatibility, ensuring the seamless operation of the integrated features within the application.

# **Test Principles**

These fundamental data concepts serve as the cornerstones of our project's data management and use strategy.

1. Bringing software development to a higher level of efficiency and quality starts with testing as early in the lifecycle as possible. This proactive approach minimizes resources and costs needed for later phases of correction by identifying and addressing issues in their early stages.
2. Defects and vulnerabilities must be methodically found and documented in order to test this program. This emphasizes the practical goal of finding problems rather than trying to prove that error-free software is an unachievable ideal.
3. The application's testing procedure is specially designed to fit the project's particular context, including its priorities, resource limitations, and specified requirements. This guarantees that testing approaches and strategies complement the project's unique features.
4. This application's testing methodology is made to change with time to efficiently respond to shifting conditions. New test cases and approaches are added as needed in order to find and fix new flaws and uphold a high level of quality assurance.
5. This application's testing strategy uses a risk-driven approach, which strategically allocates testing efforts with a focus on areas that are thought to be more likely to have defects. This way, resources are maximized where they can have the biggest effects on the overall quality and reliability of the product.
6. The team is committed to maintaining its independence and creating an environment that minimizes potential conflicts of interest. This will allow the focus to remain on finding flaws and evaluating the software quality objectively, ensuring an unbiased and impartial evaluation of the application.
7. T The team intends to conduct an exploratory approach to software review, employing creative problem-solving methodologies and critical thinking skills to comprehensively study the application. This will enable them to methodically find potential hidden problems that may evade traditional testing methods.
8. Application testing is a continuous, iterative process that includes all phases of the software development lifecycle. It is driven by feedback and enables software to be continuously improved in terms of both quality and performance.

# **Data Approach**

The web application has pre-existing data, which is a useful tool for carrying out extensive testing operations. The information in this existing collection is varied and reflects real-world events and usage patterns. Examples of this information include transaction records, user profiles, and different system setups. There are several benefits to using this data for testing, including the capacity to analyze performance under various load levels, mimic user interactions, and appraise the application's functioning under realistic circumstances. Furthermore, having access to this pre-populated dataset saves time and resources during testing by removing the need to manually generate enormous amounts of test data, all the while guaranteeing that the testing is reliable and indicative of real-world usage scenarios.

## **Scope and Levels of Testing**

### **Exploratory**

The purpose of exploratory testing is to uncover unexpected defects and issues by allowing testers to evaluate the software dynamically and creatively without predefined test cases, making it particularly effective for finding complex and hidden problems.

The scope of exploratory testing encompasses the entire application, allowing testers to investigate all aspects and functionalities of the software in a flexible and adaptable manner to uncover defects and vulnerabilities.

The responsibility for conducting exploratory testing lies squarely with the testing team, entrusting them with the task of dynamically and creatively evaluating the software to identify defects and vulnerabilities beyond the scope of predefined test cases.

The method of exploratory testing involves testers using their domain knowledge, intuition, and creativity to interact with the software, seeking out defects, and documenting their findings in real-time.

### **Functional Test**

Functional testing is to confirm that a software application fits the requirements and does its intended tasks accurately and consistently. It does this by checking that its features and functionalities work as intended.

A thorough assessment of every feature and component of the program is included in the scope of functional testing to make sure they meet specified criteria and reliably and consistently carry out their intended duties.

The testing team oversees carrying out functional testing. They will methodically assess every feature and component of the program to make sure they comply with the requirements and successfully and reliably carry out their intended functions.

Utilizing Selenium IDE, an effective and popular tool well-known for its capacity to automate web application testing and support the methodical evaluation of features and functionalities, is the selected approach for functional testing.

### **User Acceptance Test (UAT)**

By evaluating a software application's functionality, usability, and general user satisfaction, user acceptance testing (UAT) aims to confirm that it satisfies end users' expectations and requirements and is suitable for deployment.

Testing for user acceptability includes determining how well the program fulfills end users' needs and expectations while also evaluating its functionality, usability, and compatibility with specific end user criteria and organizational goals.

The task of systematically assessing the software's usability, functionality, and compliance with end users' needs to determine whether it is ready for deployment falls under the purview of the testing team's user acceptance testing technique.

The approach that has been selected for user acceptance testing involves using Apache JMeter, an open-source testing tool that is useful for both load and performance testing. It can also be beneficial to UAT by making sure that the application performs as expected in a variety of scenarios.

## **Test Deliverables**

While the mentioned documents are essential deliverables for functional testing, it is important to note that the scope of deliverables extends beyond these specific documents, encompassing any additional artifacts, records, or reports that are deemed necessary to comprehensively document and communicate the outcomes and progress of the functional testing process.

* + - 1. Test Plan: An overarching document outlining the strategy, objectives, and scope of functional testing.
      2. Test Cases: A set of documented test cases specifying inputs, expected results, and execution steps for testing each functional requirement.
      3. Test Data: Necessary data sets and parameters required to execute the test cases effectively.
      4. Test Scripts: For automated functional testing, scripts that automate the execution of test cases.
      5. Defect Reports: Documentation of any defects, including their severity, steps to reproduce, and status.
      6. Test Summary Report: A comprehensive report summarizing the results of functional testing, including pass/fail status, defect metrics, and coverage.
      7. Traceability Matrix: A matrix mapping test cases to specific functional requirements to ensure comprehensive coverage.
      8. Test Environment Setup: Documentation of the configuration and setup of the test environment.
      9. Test Progress Reports: Periodic reports providing an overview of the testing progress, including test execution status and any deviations from the test plan.
      10. Test Completion Report: A final report summarizing the overall outcome of functional testing, highlighting any significant findings or recommendations.
      11. Test Closure Report: Documentation of the formal closure of functional testing, including lessons learned and recommendations for future testing efforts.
      12. User Acceptance Test (UAT) Results: If applicable, documentation of UAT results and any related findings.

These deliverables are crucial for documenting, tracking, and communicating the progress and outcomes of functional testing efforts.

## **Milestone List**

While these milestones represent critical points in the functional testing process, it is essential to recognize that the specific milestones may vary depending on the project's unique requirements and complexity, and additional milestones or adjustments may be introduced as needed to ensure the comprehensive and successful execution of functional testing activities.

* + 1. Test Planning Complete: The development of the functional testing plan, including objectives, scope, and resource allocation, is finalized, and approved.
    2. Test Environment Setup: The test environment, including hardware, software, and data, is prepared, and configured to support functional testing.
    3. Test Case Design: The creation and documentation of test cases, encompassing various functional scenarios and requirements.
    4. Test Execution Initiation: The formal commencement of functional testing, with the execution of the first set of test cases.
    5. Mid-Testing Review: An interim review of testing progress and findings, often used to reassess the testing strategy and make necessary adjustments.
    6. Defect Identification: The detection and documentation of defects, including their severity and impact on functionality.
    7. Test Execution Completion: The conclusion of test case execution, signaling the end of the functional testing phase.
    8. Defect Resolution: The process of addressing and resolving identified defects, including retesting to confirm fixes.
    9. User Acceptance Testing (UAT) Initiation: If applicable, the initiation of UAT involves end-users validating the software's functionality.
    10. UAT Completion: The conclusion of the UAT phase, with user feedback and approval.
    11. Test Closure: The formal closure of the functional testing phase, including documentation of results and lessons learned.
    12. Final Test Summary Report: A comprehensive report summarizing the entire functional testing effort, including test results, defect metrics, and overall coverage.
    13. Testing Sign-off: Formal approval from stakeholders indicating that the software has passed functional testing and is ready for release.
    14. Release to Production: The deployment of the tested and approved software to the production environment.

These milestones serve as key checkpoints and decision points throughout the functional testing process, ensuring that testing progresses systematically and aligns with project objectives and timelines.

## **Test Effort Estimate**

These time effort estimates represent the anticipated allocation of resources and hours required for each specific testing process, serving as essential benchmarks for project planning and ensuring the efficient and effective execution of testing activities.

|  |  |
| --- | --- |
| **Activity** | **Hours** |
| Test Duration | 24 hrs. |
| Test Case Development | 96 hrs. |
| Test Execution | 48 hrs. |
| Test Data Preparation | 48 hrs. |
| Test Environment Setup | 48 hrs. |
| Defect Management | 72 hrs. |
| Test Reporting and Documentation | 120 hrs. |
| Test Automation | 24 hrs. |
| User Acceptance Testing (UAT) | 72 hrs. |
| Test Coordination and Management | 24 hrs. |
| Test Review and Analysis | 96 hrs. |
| Testing Tool Implementation | 48 hrs. |

Table 1. Test Effort Estimate

# **Test Areas and Specifications**

## **Features to be Tested (Use Cases)**

The following list outlines the specific features and functionalities that are designated for testing within the web application.

|  |  |  |
| --- | --- | --- |
| **Use Case #** | **Use Case Name** | **Test Type** |
| BSSVUC01 | Barangay Resident Login | Exploratory, Functional, UAT |
| BSSVUC02 | Barangay Resident Account Creation | Exploratory, Functional, UAT |
| BSSVUC03 | Barangay Resident Update Account | Exploratory, Functional, UAT |
| BSSV04 | Resident Reset Password | Exploratory, Functional, UAT |
| BSSVUC05 | Create Request | Exploratory, Functional, UAT |
| BSSVUC06 | Track Request | Exploratory, Functional, UAT |
| BSSVUC07 | Create Concern | Exploratory, Functional, UAT |
| BSSVUC08 | Track Concern | Exploratory, Functional, UAT |
| BSSVUC09 | Payment Option | Exploratory, Functional, UAT |

Table 2. Features to be Tested

# **Execution Strategy**

## **Entry and Exit Criteria**

The following criteria represent the predefined conditions and prerequisites that must be met before the testing process for the web application can commence.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phase** | **Entry Criteria** | **Result** | **Notes** | **Date** |
| Test Duration | Ensuring the testing environment, resources, and essential components are adequately prepared and available. | ✔ | **DONE** | 11/14/23 |
| Test Case Development | Ensuring the necessary requirements, specifications, and design documentation are accessible and comprehensible to create effective test cases. | ✔ | **DONE** | 11/14/23 |
| Test Execution | Ensuring the testing environment, test data, and test scripts are ready for use, and any dependencies are met. | ✔ | **DONE** | 11/14/23 |
| Test Data Preparation | Ensuring the relevant data sources are available, data generation or extraction tools are configured, and data privacy or security concerns are addressed. | ✔ | **DONE** | 11/14/23 |
| Test Environment Setup | Ensuring that hardware, software, networks, and other infrastructure components are in place and ready for testing. | ✔ | **DONE** | 11/14/23 |
| Defect Management | Ensuring that defect tracking tools, processes, and communication channels are established and functional. | ✔ | **DONE** | 11/14/23 |
| Test Reporting and Documentation | Ensuring the templates, formats, and reporting mechanisms are in place and agreed upon. | ✔ | **DONE** | 11/14/23 |
| Test Automation | Ensuring the automation tools, frameworks, and test scripts are set up and aligned with the testing objectives. | ✔ | **DONE** | 11/14/23 |
| User Acceptance Testing (UAT) | Ensuring the software is functionally complete, that the test environment mirrors the production environment, and that the UAT team is prepared to conduct testing. | ✔ | **DONE** | 11/14/23 |
| Test Coordination and Management | Ensuring that the test plan, schedule, and resource allocation are established, and the testing team is adequately briefed and organized. | ✔ | **DONE** | 11/14/23 |
| Test Review and Analysis | Ensuring that test execution has been completed, and test data and logs are available for analysis. | ✔ | **DONE** | 11/14/23 |
| Testing Tool Implementation | Ensuring that the selected tools are installed, configured, and integrated into the testing environment effectively. | ✔ | **DONE** | 11/14/23 |

Table 3. Entry Criteria

The specified criteria serve as the predefined benchmarks and conditions that must be satisfied for the successful completion and conclusion of the testing process for the web application.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Phase** | **Exit Criteria** | **Test Team** | **Technical Team** | **Notes** | **Date** |
| Test Duration | Indicating the allocated time for testing has been used efficiently and effectively. |  |  |  |  |
| Test Case Development | Ensuring all planned test cases are created, reviewed, and documented as per the testing objectives. |  |  |  |  |
| Test Execution | Indicating all planned test cases have been executed, defects have been identified and resolved, and testing objectives have been met. |  |  |  |  |
| Test Data Preparation | Ensuring all required test data is generated, validated, and made available for testing activities. |  |  |  |  |
| Test Environment Setup | Indicating the testing environment, including hardware, software, and configurations, is ready for testing activities. |  |  |  |  |
| Defect Management | Ensuring all identified defects have been tracked, prioritized, resolved, and documented in accordance with the project's defect management process. |  |  |  |  |
| Test Reporting and Documentation | Indicating all test results, reports, and documentation have been completed, reviewed, and approved as per the testing objectives. |  |  |  |  |
| Test Automation | Ensuring all automated test scripts are developed, executed, and maintained as per the testing objectives, and that they provide reliable test results. |  |  |  |  |
| User Acceptance Testing (UAT) | Indicating the software has been thoroughly validated by end-users, meets their acceptance criteria, and is ready for production deployment. |  |  |  |  |
| Test Coordination and Management | Ensuring the testing activities have been effectively planned, executed, monitored, and that relevant reports and documentation are complete and reviewed. |  |  |  |  |
| Test Review and Analysis | Indicating test results have been analyzed, reviewed, and appropriate actions have been taken based on the findings. |  |  |  |  |
| Testing Tool Implementation | Ensuring the selected testing tools are successfully integrated, configured, and operational in the testing environment as per project requirements. |  |  |  |  |

Table 4. Exit Criteria

## **Test Cycles**

These represent the sequential phases or test cycles that the team will undergo during the evaluation of the web application, each with its specific objectives and testing focus to ensure comprehensive testing coverage and software quality.

1. The team plans to initially conduct one cycle of exploratory testing, and if and only if errors or defects are discovered during this cycle, subsequent cycles of exploratory testing will be conducted to address and validate the identified issues. It allows us to focus additional testing efforts on areas of the application where problems have been found, ensuring a thorough evaluation and resolution of defects.
2. The team's testing approach involves an initial cycle of functional testing, and if errors or defects are detected during this cycle, subsequent cycles of functional testing will be initiated exclusively to address and rectify the identified issues. This iterative strategy ensures that the software's functionality is rigorously examined, and any shortcomings are thoroughly addressed through multiple testing cycles.
3. The team's User Acceptance Testing (UAT) strategy entails an initial cycle of UAT testing. Only in the event of identifying errors or issues during this cycle, subsequent UAT cycles will be initiated specifically to address and resolve the identified concerns. It aims to ensure that the software meets the expectations of end-users by allowing for focused testing and refinement when necessary.

## **Validation and Defect Management**

Validation is an integral component of the software development and testing process, serving as a comprehensive assessment mechanism to ascertain that the software meets precisely defined requirements and operates in a manner consistent with its intended functionality. This multifaceted process not only verifies the correctness of the software but also validates its relevance and value proposition to stakeholders. By undergoing validation, the team meticulously examines every aspect of the software, meticulously comparing its behavior and output against predetermined criteria. This validation process, rooted in thoroughness and precision, ensures that the software serves its designated purpose effectively and that it holds the potential to meet or exceed the expectations of end-users and business objectives, thus reaffirming its suitability for deployment and use in real-world scenarios.

Defect management is a systematic and indispensable facet of the software testing and quality assurance process, allowing the testing team to methodically detect, document, monitor, prioritize, and rectify any imperfections or irregularities detected throughout the testing phase. These defects encompass a wide spectrum, ranging from coding errors and functional malfunctions to deviations from specified requirements or any unexpected and undesirable behavior exhibited by the software. Ensuring that concerns are promptly documented and resolved in an organized and effective manner are two of the most important benefits of the defect management process, which is critical to maintaining and improving the quality of software. The software's overall quality is enhanced by this methodical approach to defect management, which benefits stakeholders and end users alike by increasing the software's dependability and performance.

The team will utilize TestlLink, a test management tool, to facilitate automated reporting and tracking of anomalies and discrepancies encountered during the testing process of the web application, streamlining the identification and resolution of issues for improved efficiency.

## **Test Metrics**

The team generated test metrics and it would seek approval from the project manager to monitor the progress and effectiveness of the testing process, with the following metrics as examples:

|  |  |  |
| --- | --- | --- |
| **Report** | **Description** | **Frequency** |
| Test Preparation & Execution Status | Provides an overview of the readiness and progress of the testing activities, including test planning, case development, and execution. | Update daily/weekly (if necessary) |
| Daily Execution Status | Offers a snapshot of the testing progress, summarizing the outcomes of daily testing activities, including test execution results, defects identified, and any issues encountered. | Update daily |
| Project Weekly Status Report | Provides a comprehensive summary of the project's progress, including achievements, challenges, milestones reached, and upcoming tasks, serving as a valuable communication tool for stakeholders. | Update the report on a weekly basis, as necessary, using a template provided by the project team to ensure consistency and standardized reporting. |

Table 5. Test Metrics

## **Defect Tracking and Reporting**

This section aims to establish a structured approach for the team to actively identify, thoroughly document, closely monitor, and effectively communicate any issues, anomalies, or defects encountered throughout both the testing and development phases. This systematic process is designed to streamline defect resolution efforts, contributing to the enhancement of software quality and reliability.

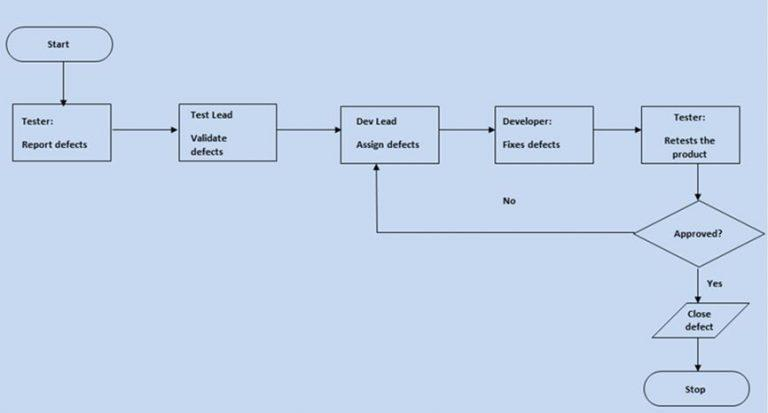


Figure 1. Defect Tracking and Reporting Diagram

# **Test Management Process**

## **Test Management Tool**

The team will leverage the combined capabilities of Selenium IDE and TestLink to orchestrate and streamline the comprehensive testing process for the South Signal Village Barangay Web Application, particularly focusing on assessing the functionality from the perspective of customers and users. Selenium IDE, with its robust automation capabilities, will be instrumental in executing test cases efficiently, automating repetitive tasks, and ensuring thorough test coverage. Simultaneously, TestLink will serve as a centralized hub for test case management and reporting, allowing for meticulous documentation of test cases, tracking of test results, and seamless communication of findings. This integrated approach not only enhances the efficiency of testing but also strengthens the team's ability to monitor and manage the quality assurance process effectively, ensuring that the web application aligns seamlessly with user expectations and functional requirements.

## **Test Design Process**

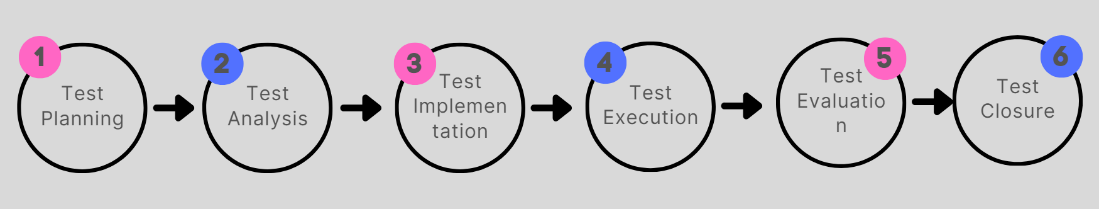


Figure 2. Test Management Process Diagram

The team will adhere to the systematic test design process, with each phase detailed below, to ensure a structured and comprehensive approach to software testing that encompasses all critical aspects of quality assurance.

1. **Test Planning** – During this phase, the team will meticulously identify and allocate the necessary resources and select appropriate test management tools, concurrently formulating a comprehensive test strategy, delineating clear test objectives, and establishing a well-structured test schedule to ensure the systematic and efficient execution of the testing process.
2. **Test Analysis** – During this crucial stage, the team will conduct a meticulous analysis of the software's requirements and specifications, with a primary focus on identifying the specific features and functionalities that require rigorous testing. Subsequently, based on this comprehensive analysis, the team will define targeted test scenarios, leveraging their insights to prioritize test cases systematically, considering factors such as risk and criticality, to ensure that testing efforts are optimized and aligned with the most pivotal aspects of the software.
3. **Test Implementation** – Within this phase, the team will meticulously configure the test environment, encompassing essential tasks such as the establishment of dedicated test databases, server configurations, and the integration of requisite testing tools, ensuring that a robust and conducive testing environment is prepared for executing thorough and effective testing procedures.
4. **Test Execution** – During the execution stage, the team will initiate and oversee the active operation of the test environment, diligently monitoring its performance, managing the entire test execution process, meticulously recording all test results, and promptly generating comprehensive test reports, culminating in a detailed and transparent account of the testing outcomes.
5. **Test Evaluation** – The team will engage in a comprehensive analysis to meticulously scrutinize and identify any test results, defects, errors, or other anomalies that may arise during the execution phase, prioritizing these findings for further investigation and resolution as necessary to ensure software quality.
6. **Test Closure** – In this conclusive stage of the test design process, the team will compile and present a comprehensive report detailing all testing activities and outcomes, encapsulating the culmination of their efforts, findings, and insights to provide a holistic view of the software's quality and testing effectiveness.

## **Test Execution Process**

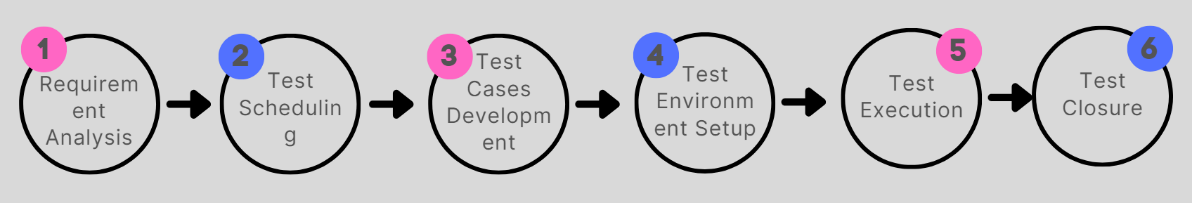


Figure 3. Test Execution Process

The team will rigorously adhere to the test execution process, and below, each phase of this meticulously planned approach is elaborated upon to ensure precise execution, comprehensive testing, and effective quality assurance.

1. **Requirement Analysis** - The team will diligently gather and assemble all requisite requirements essential for the seamless execution of the testing procedures.
2. **Test Scheduling** - The team will allocate and schedule specific dates and times for the testing of each feature, ensuring a structured approach to commence and conclude testing activities in an organized and efficient manner.
3. **Test Cases Development** - During this phase, the team will craft comprehensive test cases for each module, providing a detailed blueprint for conducting systematic and thorough testing of the software's individual components.
4. **Test Environment Setup** - The team will meticulously configure the network, hardware, and software components to facilitate an environment conducive to evaluating, enabling testing teams to seamlessly execute test cases and evaluate the software's performance.
5. **Test Execution** - In this phase, the team initiates the execution of tests, diligently capturing and documenting all identified defects, which are subsequently submitted for comprehensive correction and remediation to enhance the software's quality and reliability.
6. **Test Closure** - In this concluding stage, the team diligently records comprehensive information detailing the execution of the test, encompassing all aspects of the testing process to provide a comprehensive and transparent account of the testing activities and outcomes.

## **Test Risks and Mitigation Factors**

These are the identified risks and their corresponding mitigation strategies for the testing phase of the project, aimed at ensuring a successful and smooth testing process.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Category** | **Risk** | **Probability** | **Impact** | **Mitigation Plan** |
| Resource Risk | Insufficient testing personnel | Medium | High | Read online resources to enhance their testing skills and capabilities. |
| Technical Risk | Inadequate test environment | Low | High | Ensure proper setup and maintenance of the test environment, consider cloud-based solutions for scalability. |
| Schedule Risk | Development delays affecting testing timelines | Medium | Medium | Regularly monitor the development progress and adjust the testing schedule as needed. |
| Data Risk | Inaccurate or insufficient test data | High | High | Generate realistic and comprehensive test data, automate data generation where possible. |
| Scope Risk | Scope creep and changing project specifications | Medium | Medium | Clearly define and document project requirements, communicate changes effectively, and follow a change control process. |
| Tool Risk | Unreliable testing tools | Low | Medium | Continuously evaluate and update testing tools, have backup tools in case of failures. |
| Communication Risk | Poor communication among team members | Medium | Low | Establish regular team meetings, use collaboration tools, and encourage open communication. |
| Test Data Security Risk | Data breaches and security vulnerabilities | Low | High | Implement robust security measures, conduct regular security audits, and apply security patches. |

Table 6. Test Risks and Mitigation Factors

# **Role Expectations**

The team's role expectations will be clearly defined and communicated, outlining each team member's responsibilities, tasks, and contributions to ensure a coordinated and efficient approach to project execution.

|  |  |  |  |
| --- | --- | --- | --- |
| MEM# | *Name* | *Role* | *Contact Info* |
| MEM01 | Aspeli, Mabelle T. | Test Manager | [mtaspeli@student.apc.edu.ph](mailto:mtaspeli@student.apc.edu.ph) |
| Automation Tester |
| MEM02 | Villamin, Sofia Emmanuelle R. | Test Lead | [srvillamin@student.apc.edu.ph](mailto:srvillamin@student.apc.edu.ph) |
| Test Analyst |

Table 7. Role Expectations

### **Project Management**

The project manager will assume the responsibility of reviewing and granting approval for critical documentation, including the Test Plan and Test Strategy, ensuring that they align with project objectives and quality assurance standards.

### **Test Planning (Test Lead)**

Within this section, a detailed account of the various activities that must be diligently performed throughout the testing process is presented, ensuring a systematic and structured approach to quality assurance.

1. It is essential for the team to thoroughly comprehend the provided documentation to gain a clear understanding of what information and content should be appropriately incorporated into each section of the paper, ensuring alignment with project requirements and objectives.
2. Clearly delineate the roles and responsibilities of each team member, considering their individual knowledge, skills, and capabilities, to establish a cohesive and efficient framework for task allocation and accomplishment.
3. Maintain constant communication with the team about the accomplishments and status of the documentation, encouraging openness and keeping everyone in the loop for the duration of the project.
4. Keep lines of communication open with the project manager and all other involved parties to allow updates to be shared, promote teamwork, and guarantee that everyone is aware of the project's progress and advancements.
5. Thoroughly document all conducted tests, meticulously recording any identified errors and bugs, to provide a comprehensive repository of actionable information for project stakeholders to address and rectify.

### **Test Team**

The testing team comprises skilled professionals responsible for meticulously evaluating software through various testing methodologies to ensure its functionality, reliability, security, and overall quality.

The responsibilities associated with this role include the following:

1. Formulate comprehensive test plans and strategies by thoroughly analyzing project requirements and objectives.
2. Generate intricate test cases, meticulously craft test scripts, and devise comprehensive test scenarios to ensure thorough testing coverage.
3. Set up the test environment, encompassing the configuration of both hardware and software components, to create an optimal testing environment.
4. Generate or obtain the requisite test data essential for effective test execution.
5. Conduct test case execution, meticulously document test results, and promptly report any identified defects for resolution.
6. Identify, comprehensively document, prioritize, and diligently track defects throughout their lifecycle until resolution is achieved.
7. Create and consistently update automated test scripts leveraging testing tools to facilitate efficient and repeatable testing processes.
8. Generate comprehensive test reports, offering in-depth insights into the progress of testing activities and providing detailed test results for thorough assessment.
9. Engage in effective collaboration with developers, project managers, and stakeholders to promptly address issues, offer timely updates, and ensure seamless communication within the testing process.
10. Verify the successful completion of all test-related tasks and confirm that testing is prepared for final sign-off and approval.

**Test Lead**

The test lead serves as a pivotal role in test management, responsible for overseeing and coordinating the testing efforts of a team, ensuring adherence to evaluate plans, and facilitating communication among team members and stakeholders to achieve testing objectives effectively.

The responsibilities associated with this role encompass a wide range of activities, including test planning, test design, test execution, defect management, test reporting, and coordination with various stakeholders to ensure the successful execution of the testing phase are the following:

1. Responsible for meticulously crafting and consistently updating the test plan, which serves as a comprehensive document outlining the testing scope, objectives, testing approach, and a well-structured schedule to ensure that testing activities align with project goals and timelines.
2. Plays a critical role in strategically defining the overall testing approach, encompassing decisions regarding testing levels, various test types, and the selection of appropriate testing methodologies, all of which are essential components to ensure the thorough and effective evaluation of the software under test.
3. Initiative-taking identification of potential risks and challenges pertaining to the testing process, and their responsibility extends to the development and implementation of robust mitigation plans to preemptively address these issues, thus ensuring a smoother testing workflow.
4. Assumes the crucial role of supervising and guiding the testing team in the meticulous development of test cases, test scripts, and test data, all of which are intricately aligned with the project's specific requirements and detailed specifications to ensure comprehensive test coverage.
5. Oversees the execution phase, actively monitoring and managing the test case execution process to ensure that testing objectives are successfully achieved while meticulously recording and tracking defects for resolution, thus maintaining the quality and integrity of the testing efforts.
6. Produces comprehensive test reports that succinctly summarize the results of testing activities, emphasizing critical findings, and providing valuable insights to project stakeholders, facilitating informed decision-making, and ensuring transparency in the testing process.
7. Ensures that the entire testing process adheres to established quality assurance standards and industry best practices, maintaining the highest levels of quality and reliability in the software testing efforts.
8. Conducts regular performance evaluations and assessments of team members to gauge their individual and collective contributions to the testing process, ensuring the team operates effectively and efficiently.
9. Coordinates the setup and ongoing maintenance of the test environment, meticulously aligning it with the evolving project requirements to provide a stable and conducive testing environment for the team's activities.
10. Oversees the defect tracking and reporting process, carefully supervising the prioritization of defects to ensure that critical issues are addressed promptly, facilitating efficient defect resolution, and enhancing overall software quality.

### **Development Team**

The development team's primary duties include designing, creating, and maintaining the software, making sure it complies with industry standards for coding techniques and performs as intended in addition to meeting project requirements. To promote efficient communication and cooperation between the project's testing and development phases, Mabelle Aspeli, the team's test manager, will be extremely important in advising Team Developmentality's development team.

This role encompasses a wide range of responsibilities, including successful software design, development, testing, and maintenance, as well as effective cross-functional team communication, adherence to industry best practices and standards, and more. Among them are the following:

1. Code reviews, in which developers and testers jointly review the source code to evaluate its quality, spot errors, and detect possible problems early in the software development cycle, are an essential cooperative process that improves overall software quality and dependability.
2. Supplying pertinent technical documentation is essential to assist testers in comprehending the software's architecture and functionality, facilitating more effective and efficient testing processes.
3. For the software development process to proceed smoothly and a high-quality product to be delivered, defects or issues found by the testing team must be promptly addressed and resolved.
4. Helping to configure and set up the required databases, infrastructure, and test environments is crucial to guaranteeing that the testing team has the tools and circumstances needed to carry out exhaustive and efficient testing.
5. Actively collaborating with the testing team to clarify requirements, respond to inquiries, and offer context on software features is essential for fostering a productive and communicative relationship between development and testing, leading to more effective testing efforts and issue resolution.
6. Being open to feedback from the testing team and implementing required code changes to enhance software quality is a fundamental aspect of the development team's commitment to delivering a reliable and robust software product.
7. Sustaining transparent and effective communication with the testing team is paramount in ensuring a seamless testing process, promoting collaboration, and promptly addressing any emerging issues or concerns.
8. The capacity to adjust and modify plans in response to input from testing is a crucial component of a dynamic and responsive development process that produces higher-quality software.
9. Achieving proper reflection of software upgrades and modifications necessitates maintaining current technical documentation to guarantee that all interested parties have access to the most recent details regarding the system.
10. Working together with the testing team to find areas where the process can be made more efficient and effective promotes a continuous improvement culture that results in better software development procedures and higher-caliber outputs.

# **Test Environment**

Testing will be conducted using a combination of two applications to ensure precision and reliability in generating accurate results and outputs. These are the specific applications that have been selected for use in the comprehensive testing of the web application:

1. SoCIT Cloud Environment
2. TestLink – open-source test management tool, plays a crucial role in streamlining the organization of test cases, execution of tests, and reporting on the results, which is especially beneficial for managing User Acceptance and Functional testing within the team's testing process.

# **Approvals**

These individuals are responsible for reviewing and providing their approval for this document.

|  |  |
| --- | --- |
| **Signature** |  |
| **Name** |  |
| **Role** |  |
| **Date** |  |

|  |  |
| --- | --- |
| **Signature** |  |
| **Name** |  |
| **Role** |  |
| **Date** |  |

|  |  |
| --- | --- |
| **Signature** |  |
| **Name** |  |
| **Role** |  |
| **Date** |  |

|  |  |
| --- | --- |
| **Signature** |  |
| **Name** |  |
| **Role** |  |
| **Date** |  |