*Work Request Application*

Test Plan

Version *1.0*

*Nov 5 2021*

VERSION HISTORY

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version #** | **Implemented**  **By** | **Revision**  **Date** | **Approved**  **By** | **Approval**  **Date** | **Reason** |
| 1.0 | *Dave Leake* | *Nov 5 2021* | *Ian Oliver* | *Nov 8 2021* | Test Plan Draft |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Contents

[1 Introduction 4](#_Toc87219048)

[2 Application Components Tested 4](#_Toc87219049)

[3 Application Features Tested 4](#_Toc87219050)

[4 Types of Tests Needed 4](#_Toc87219051)

[5 Test Documentation 5](#_Toc87219052)

[6 Testing Environment and Needs 5](#_Toc87219053)

[7 Testing Schedule 5](#_Toc87219054)

[8 Testing Breakdown 6](#_Toc87219055)

[8.1 Manual Testing 6](#_Toc87219056)

[8.2 Unit Testing 6](#_Toc87219057)

[9 Acceptance Testing 6](#_Toc87219058)

[10 Testing Criteria 7](#_Toc87219059)

[11 Potential Risks 7](#_Toc87219060)

# Introduction

The Project Test Plan (PTP) defines the scope, methods, resources, and schedule of all testing activities to ensure that the Work Request application meets all the project requirements. This plan outlines all of the application components and features that need testing, the specific type of testing required, the team members responsible for testing, the testing schedule, and any potential risks involved with executing the test plan. The test team will consist of the participating members of Team 3 Dave Leake, Ian Oliver and Tchouente-Tsebo, Aurelien(Will).

# Application Components Tested

The Work Request Application consists of three primary components which can be categorized into frontend and backend functional areas. The frontend component will consist of a user interface (UI) written with QT and Java. The backend components consist of Microsoft SQL Server. Each of these components are tested to ensure they function as intended when interacting with one another.

For the UI, all user accessible functions are tested manually. The backend SQL Server is tested via various scripts and reports. This guarantees that the Work Request Application operates as expected and ensures the user experience is consistent. Manual code review also takes place for all components.

# Application Features Tested

* UI
  + Front Page/ Splash Page
    - Login Page
    - Username/Email field
    - Password
  + Work Request
    - Construction
    - Engineering
    - Environmental
  + Data Analysis
    - Status Widget (Current status of selected work request)
    - Resources Widget
    - Timeline Widget
    - Calendar Schedule
* Backend
  + Unit Tests
  + Manually verifying any errors found

# Types of Tests Needed

Multiple forms of testing will be implemented including, manual, unit, and acceptance testing. The Test Director developed unit tests for each function in the SQL backend. Acceptance testing is performed by the entire team to determine that the Work Request application fulfills all the project plan requirements. As outlined above, each feature is tested to confirm expected functionality and error handling.

# Test Documentation

Each test is documented using a Pass/Fail table which includes, the test case, all inputs, expected output, actual output, and the result (Pass/Fail). Screenshots accompany each test case and are included with testing documentation along with the Pass/Fail table. All release versions of the Work Request application have undergone testing. A testing report has been generated to document testing results throughout each phase of development and version release. The test reports include a heading with test outline, a pass/fail table, and a summary of the test results. A sample of the testing table is shown below in Figure 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID #** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| 1 |  |  |  |  |
| 2 |  |  |  |  |

*Figure 1: Example of Pass/Fail table for test documentation.*

# Testing Environment and Needs

Testing has been performed on the *Windows* environment. The Work Request application has multiple dependencies that must be resolved to successfully launch and test the application. These dependencies have been formally documented and provided to the testers and are included in the testing report.

At a minimum, the testing environment will require an up-to-date version of:

* Java Development Kit (ex. JDK-17.0)
  + <https://www.oracle.com/java/technologies/downloads/#jdk17-windows>
* Junit 5
  + <https://github.com/junit-team/junit5/releases/tag/r5.8.1>
* Eclipse IDE
  + <https://www.eclipse.org/ide/>
* Qt 5.15 or Qt 6.x and the matching Qt Jambi version
  + <https://www.qt.io/download-qt-installer>
  + <https://github.com/OmixVisualization/qtjambi/wiki>
* The test system will also need internet access, and an up-to-date web browser (ex. Chrome or Firefox)
* Word processing software (ex. Microsoft Word) for documentation.

Once the application successfully passes all test cases in the development environment, it will transition to a production staging environment and undergoes additional testing before deploying to production. All tests are conducted by following appropriate testing guidelines and a detailed user guide.

# Testing Schedule

Testing will be completed by Monday of each week.

# Testing Breakdown

## Manual Testing

The Work Request UI portion is primarily tested using manual methods as outlined above in section 3. The SQL database is also manually inspected for accurate structure as defined in the Project Design documentation. Manual testing is repeated on all elements that were modified between development phases.

## Unit Testing

Each of the functions of the Java backend has been unit tested with the Junit framework, with each test case identified in table 1 below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID #** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| Construction Work Request | Construction Work Request Button Clicked | Blank Construction Work Request Form displayed |  |  |
| Engineering Work Request | Engineering Work Request Button Clicked | Blank Engineering Work Request Form Displayed |  |  |
| Environmental Work Request | Environmental Work Request Button Clicked | Blank Environmental Work Request Form Displayed |  |  |
| Construction Work Request  Submit | Construction Work Request Submit Button Clicked | Completed Form Displayed; Data exists in DB |  |  |
| Engineering Work Request  Submit | Engineering Work Request Submit Button Clicked | Completed Form Displayed; Data exists in DB |  |  |
| Environmental Work Request  Submit | Environmental Work Request Submit Button Clicked | Completed Form Displayed; Data exists in DB |  |  |
| Construction Report | Construction Report Button Clicked | Construction Analytics Report Displayed |  |  |
| Engineering Report | Engineering Report Button Clicked | Engineering Analytics Report Displayed |  |  |
| Environmental Report | Environmental Report Button Clicked | Environmental Analytics Report Displayed |  |  |
| String Field Special Character | Each String Field input with series of special characters | Database captures characters or ascii value |  |  |
| Monetary Fields Negative Value | Each Monetary Field input with negative values | Only numeric, $, and . allowed in field |  |  |
| Save Button | Save button on relevant field pressed | All fields on the same tab sends current field data |  |  |

*Table 1: Work Request Unit Tests.*

# Acceptance Testing

Acceptance testing will be performed throughout the week and feedback provided to developers to facilitate agile changes to be made based on team feedback. Final acceptance testing will be performed after the final phase of development to verify the application completely fulfills all requirements set forth by the project plan. Each user-facing function will be tested to ensure correct implementation. The table below breaks down all of the user facing functions into individual acceptance tests.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID #** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| Construction Work Request | Construction Work Request Button Clicked | Blank Construction Work Request Form displayed |  |  |
| Engineering Work Request | Engineering Work Request Button Clicked | Blank Engineering Work Request Form Displayed |  |  |
| Environmental Work Request | Environmental Work Request Button Clicked | Blank Environmental Work Request Form Displayed ed |  |  |
| Construction Work Request  Submit | Construction Work Request Submit Button Clicked | Completed Form Displayed; Data exists in DB |  |  |
| Engineering Work Request  Submit | Engineering Work Request Submit Button Clicked | Completed Form Displayed; Data exists in DB |  |  |
| Environmental Work Request  Submit | Environmental Work Request Submit Button Clicked | Completed Form Displayed; Data exists in DB ed |  |  |
| Construction Report | Construction Report Button Clicked | Construction Analytics Report Displayed |  |  |
| Engineering Report | Engineering Report Button Clicked | Engineering Analytics Report Displayed |  |  |
| Environmental Report | Environmental Report Button Clicked | Environmental Analytics Report Displayed ed |  |  |
| String Field Special Character | Each String Field input with series of special characters | Database captures characters or ascii value |  |  |
| Monetary Fields Negative Value | Each Monetary Field input with negative values | Only numeric, $, and . allowed in field |  |  |
| Save Button | Save button on relevant field pressed | All fields on the same tab sends current field data |  |  |

*Table 1: Work Request Acceptance Tests.*

# Testing Criteria

The result of a test, either pass or fail, is determined by the test output matching the expected/desired output. For example, if a user requests a password-reset link by entering a valid e-mail address and the user never receives the password reset email, then the test is considered failed. Contrarily, if the password reset email is received and the user can successfully reset his/her password then the test passes. This testing methodology has been applied to each developed test. If a test fails the development team will be alerted of the defect and a fix will be implemented before the completion of the current development phase.

# Potential Risks

The following Risk Register outlines some potential risks involved with executing this test plan.

|  |  |
| --- | --- |
| **Risk** | **Potential Mitigation** |
| Unable to complete tests in specified timeframe | Test only the components that changed during the last development phase. |
| Tests do not reflect the quality of the application | Adapt testing methods to better suite application requirements. |
|  |  |

# 