Version: 1.0

Created by: Payal Brahmbhatt

Status: draft

TEST PLAN

Dashboard Client App - Automation Testing

**TABLE OF CONTENTS (TOC)**

1. INTRODUCTION
   1. Purpose
   2. Project Overview
2. SCOPE
3. Testing Strategy
4. Execution Steps
5. Reports

# INTRODUCTION

## Purpose

This test plan describes the testing approach and overall framework that will drive the testing of the Dashboard Client App. Further, the document describes coverage of testing, scope, type of testing covered and execution steps.

## Project Overview

This project includes Manual and Automated Tests for Dashboard Client App.

Automation testing is a Software testing technique to test and compare the actual outcome with the expected outcome. Automated testing entirely relies on the pre-scripted test which runs automatically to compare actual result with the expected results. This helps the tester to determine whether an application performs as expected.

Once the test suite is automated, no human intervention is required. This will improve ROI of Test Automation. The goal of Automation is to reduce the number of test cases to be run manually and not to eliminate Manual Testing altogether.

This automation suite is designed for a simple single page web application and it can be easily scalable for a wider application. Currently, we need to manually trigger the beginning of test suite, however, its implementation can be extended to execute as a part of build process.

# SCOPE

Testing of following functionalities is automated:

* Users need to be able to filter/search for the records on any one of the given properties of a record. When filter/searching by date users will supply a range, all other fields would use their corresponding data type. Users can use more than one filter at a time.
* The name/title of the application/page is: Project Dashboard

The scope of testing can be easily enhanced to other functional and non-functional requirements of the application.

 Functionalities not covered:

* Each record needs to allow a user to: change the status, change the Project Owner, and adjust the budget. These features should be available in-line without going to another page. When a record is changed please provide some sort of feedback to the user.
* Each record needs to supply a link/button/etc to go to a more in-depth page. The link is not active and you don’t need to create the secondary page. This is only to show that each record gives the user the ability to go to another page.
* Management has asked that the screen offer some high-level/summations/statistics about the current records. They are not interested in charts/graphics (they are minimalists at heart), but want some statistics related to the records in the UI, something that tells the story at a glance. They have given you some discretion in terms of what to show and how to show it, but it should reflect the current dataset.
* Please provide inactive/placeholder (non-working) UI elements for Users to:
  + add a new record
  + export the records on screen to Excel or PDF

# TESTING STRATEGY

**Unit Testing**

As a part of Unit Testing, Test cases to be automated :

## Launch Client App and verify components present

## Search/Filter for an Existing record

## Search/Filter for a Non-Existing record

## Search records using multiple properties (Logical AND operator)

## Search records using multiple properties (Logical OR operator)

# EXECUTION STEPS

Project is available at git repo: <https://github.com/PayBhatt/DashboardAppTesting>

Execute below maven command from command line:

“mvn test”

Note: We need to start the development server before executing the test suite.

# REPORTS

Test Report is automatically generated at projectDir/target/cucumber-reports/ location in html format. This report shows

* Statistics of total features executed, total scenarios and steps executed
* Each Scenario with details steps executed – Pass OR Fail OR Skipped
* Time taken by each scenario and total time taken for the test suite, date and timestamp of test suite execution
* Different views of the test status for different audiences
* Detail logs for failed scenarios with screenshot of the application at the time of failure

Sample Automation test suite execution report:

