Test Plan Document

For

Workshop Management Web Application

1 Introduction 3

2 Business Background 3

3 Test Objectives 3

4 Scope 3

5 Test types Identified 3

6 Problems Perceived 3

7 Architecture 3

8 Environment 3

9 Assumptions 3

10 Functionality 3

11 Security 4

12 Performance 4

13 Usability 5

14 Test Team Organization 6

15 Schedule 6

16 Defects Classification Mechanism 6

17 Configuration Management 6

18 Release Criteria 6

Test Plan

# Introduction

Jasmine is a behavior-driven development framework for test JavaScript code. It will be used to test both the frontend application as well as the backend REST APIs.

# Business Background

The Workshop Management Web Application is designed to track workshops, their participants, and the equipment used within them.

# Test Objectives

The Jasmine tests will ensure that all Angular components are fully functioning along with the server routes and pipe files and that all Express REST APIs are responding correctly.

# Scope

***Inclusions***

Front End: Angular

Back End: Express

***Exclusions***

None

# Test types Identified

Jasmine unit testing

# Problems Perceived

There aren’t any problems perceived at this time. The unit tests should ensure that the system is working properly.

# Architecture

The Jasmine tests for the front end will be written inside the Angular project. Each component has a file which contains the unit tests for that component. There is also a file for the server routes and for each of the pipes. These tests can be run using the “ng test” command from the command line. The Jasmine tests for the back end will be stored in a file and executed from the command line. The back-end tests will test all of the http calls to the database.

# Environment

Jasmine unit tests executed from the Windows command line.

# Assumptions

If the unit tests all pass, then the software is functioning correctly.

# Functionality

***Constraints and Resolutions***

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Customer Constraints** | **Infosys Limitations** |
| Constraint 1 |  |  |
| Constraint 2 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| . |  |  |
| - |  |  |

***Risk Identified & Mitigation Planned***

***Test Strategy***

The unit tests will test both the front end and back-end functionality.

***Automation Plans***

***Deliverables***

# Security

***Constraints and Resolutions***

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Customer Constraints** | **Infosys Limitations** |
| Constraint 1 |  |  |
| Constraint 2 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

***Risk Identified & Mitigation Planned***

***Test Strategy***

***Automation Plans***

***Deliverables***

# Performance

***Constraints and Resolutions***

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Customer Constraints** | **Infosys Limitations** |
| Constraint 1 |  |  |
| Constraint 2 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

***Risk Identified & Mitigation Planned***

***Test Strategy***

***Automation Plans***

***Deliverables***

# Usability

***Constraints and Resolutions***

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Customer Constraints** | **Infosys Limitations** |
| Constraint 1 |  |  |
| Constraint 2 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

***Risk Identified & Mitigation Planned***

***Test Strategy***

***Automation Plans***

***Deliverables***

***Compatibility Constraints and Resolutions***

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Customer Constraints** | **Infosys Limitations** |
| Constraint 1 |  |  |
| Constraint 2 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

***Risk Identified & Mitigation Planned***

***Test Strategy***

***Automation Plans***

***Deliverables***

# Test Team Organization

The test team consists of just one person, James Schlesener.

# Schedule

Unit tests will be run upon any updates made to the components in Angular or the REST APIs in Express to ensure that they pass and are working correctly.

# Defects Classification Mechanism

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type of Defects | Functionality | Performance | Security | Usability | Compatibility |
| Critical |  |  |  |  |  |
| Major |  |  |  |  |  |
| Minor |  |  |  |  |  |
| Cosmetics |  |  |  |  |  |

***Defects Logging and Status Changing Mechanism***

***Turn Around Time for defect fixes***

# Configuration Management

T

# Release Criteria

The software can be released if all the tests pass.