**Purpose of the Document**

This document describes the testing approach and overall framework that will drive the testing of the project.

Table of Contents

[1 Introduction 2](#_Toc139844596)

[1.1 Purpose 2](#_Toc139844597)

[1.2 Project Overview 3](#_Toc139844598)

[2 Scope 3](#_Toc139844599)

[2.1 In-Scope 3](#_Toc139844600)

[2.2 Out-of-Scope 3](#_Toc139844601)

[3 Testing Strategy 3](#_Toc139844602)

[3.1 Test Objectives 3](#_Toc139844603)

[3.2 Data Approach 4](#_Toc139844604)

[4 Execution Strategy 4](#_Toc139844605)

[4.1 Entry Criteria 4](#_Toc139844606)

[4.2 Exit criteria 4](#_Toc139844607)

[4.3 Defect Management 4](#_Toc139844608)

[5 Environment Requirements 5](#_Toc139844609)

[5.1 Test Environments 5](#_Toc139844610)

[6 Risks and Contingencies 5](#_Toc139844611)

# Introduction

## Purpose

This Test Plan is documented to describe the predefined scope, approach, resources, schedule of all the testing activities of the project Unicourt Support Portal, and the risks associated with and corresponding mitigation plan.

## Project Overview

As part of this project a graphical user interface will be developed for Unicourt support personnel consisting of members of Engineering, Support and Sales team.

# Scope

## In-Scope

i) Functional testing of the portal pages

ii) Non-functional testing of the portal

## Out-of-Scope

i) Security Testing

ii) Database Testing

iii) API Testing

# Testing Strategy

## Test Objectives

* Functional testing will be focused around validating
* User Authentication
* User roles
* Authorization
* Non-functional testing will be focused around validating
* Volume of users
* Request load
* Automation scripts will be executed

## Data Approach

Data for testing will be created by the data team based on the data request raised by the QA team

# Execution Strategy

## Entry Criteria

* Test environment is available
* Test data is available
* Code Merging is completed successfully
* Unit Testing is completed by the development team
* Test cases are completed, reviewed and approved
* Test scripts are completed, reviewed and approved

## Exit criteria

* All test cases are executed with 90% pass rate
* There are no open critical defects
* Test results and reports are prepared
* Any open defect will be considered as change request in future
* Defects are tracked in defect tracking tool

## Defect Management

* Defects are created in defect tracking tool
* Each defects should consist of

1. One line summary of the defect
2. Detailed steps to replicate the defect along with Expected result and Actual result
3. Test data and environment details
4. Screenshots and Logs

* Defect should be assigned to the development lead
* Once the fix is available defect should be retested
* Details of the retest should be logged accordingly
* RCA of the defect should be provided by the development team

Defects found during the Testing should be categorized as below:

1. Critical
2. High
3. Medium
4. Low

# Environment Requirements

## Test Environments

* Environment is configured accordingly with access to entire QA team
* Read access is provided to capture server logs
* Updated version of the browser is available for UI functional testing
* Tools required for NFT are installed
* Members of the QA team have read/write access to Defect tracking tool

# Risks and Contingencies

|  |  |
| --- | --- |
| **Risks** | **Mitigation** |
| Environment is down | Back-up environment should be planned |
| Dependency on data team | Application to be tested with Mock response |
| Resource unavailability | Back-up resource to be identified |