Product Quality **STRATEGY** for Talent Rocket

Project Name:	Talent Rocket Challenge Test

Prepared by :	Harsharan Deep

Version Number:
1.0

Date:
April 5**, 20**19

The Test Strategy Document is a living document that is created in the project's Requirements Definition phase, after the Requirements have been specified. The Test Strategy document describes the scope, approach, resources and schedule for the testing activities of the project. This includes defining what will be tested, who will perform testing, how testing will be managed, and the associated risks and contingencies. The Test Strategy document is maintained throughout the life of a project.

Table of Contents

Checked By

Role	Name
Head of Product & Marketing	Philipp Commandeur

1. Introduction

1.1 Overview

This is the Test Strategy for Talent Rocket Login and Registration Page
This document shall be completed and used by the project test team to
guide how testing will be managed for this project. The test effort will be prioritized and executed based on the project priorities as defined in the Project Plan and Requirements Specification. This is a living document that may be refined as the project progresses. The :A Manager, Test Team Lead, Product Manager, Project Manager, and Development Manager ETC. shall review and approve the final version of the Test Strategy document.

1.3 Definitions and Acronyms

Project name Talent Rocket

TalentRocket is HR Tech for niche markets. We are a recruiting tool with Data Driven Candidate Approach. Known reach of a network of the past with today's technology. We will tangibly shape the future. Our intelligent matching algorithm simplifies recruiting efforts and reduces HR costs.

2. Scope and Limitations.

2.1 Scope

Installation will be tested on the different platforms as described in the Requirements Specification. The testing for this will cover the installation on these platforms, as well as a set of critical functions to determine that the code will work on all platforms.

Scope of Talent Rocket will Contain Various Types of Test for all of its Sections including Jobs, Events Law school ,Training ,Login and Registration

3. Testing Approach.

3.1 Scope

The testing approach for this release shall be done in a fashion that will accommodate the current functionality in Talent Rocket products being developed for Law Professionals that it fits Law professionals requirement

Testing will be designed to encompass the following.

- Testing will cover functionality testing for all changes through the use of the test interface. This will validate base functions of the new code as it relates to the standard model of presentation for data and user entered data.

.

3.2 Test Types

3.2.1 Unit

Unit testing is testing performed to determine that individual program modules perform per the design specifications.

- **Owners**

Corresponding Lead Developers:.

- **Implementation Approach**

At the discretion of the Developer

- **Tools/Techniques**

Manual tests Mocha , CHai and Jasmine

3.2.2 Assembly

Assembly testing is designed to test a related group of program modules.

- **Owners**

Corresponding Lead Developers:.

- **Implementation Approach**

At the discretion of the Developer

- **Tools/Techniques**

Manual tests.

3.2.3 System

System testing is the process of testing an integrated system to verify that it meets specified requirements. This testing will determine if the results generated by information systems and their components are accurate and that the system performs according to specifications.

- **Owners**

Test and QA Individuals will perform this testint

- **Implementation Approach**

The objective of system testing is to verify the correctness of the newly designed items, and their interaction with th e existing functionsTesting will focus on functionality of the whole platform so that there are no bugs or issues are left
Testing will be accomplished through an organized testing process that will have repeatable tests This process will be accomplished by use of the scripts created and designed to match the requirements being developed for Customers

Planning the execution of test scripts for new functionality and regression tests will be done in coordination with the plan for developing New
Releases Testing and development will be executed in parallel, based on phased implementations, wherever possible
Test scripts will be structured to give a full range of coverage to the converted functions in both a Positive and Negative fashion, simulating what a potentially unfamiliar user might do during use. Positive test cases will reflect that the application functions as expected and described in the Requirements Specification and the Project Plan. Negative test cases are tests that exercise the limits and boundaries outside the expected designs. The results of this testing will give us some idea as to the stability for the application and its components.

Additional testing beyond the scripted test may be done where feasible to exercise the application to verify error handling and system recovery due to incorrect data or entry into fields.

- **Tools/Techniques**

This will Incluse Selenium,Jmeter , TestNG, Maven , NightWatch Js , WebDriverIO and More tools

3.2.4 Usability

Usability testing tests the ease with which users can learn and use a product.

► **Owners**

Testing Individuals -

► **Implementation Approach**

Will perform UAT and Functionality testing

► **Tools/Techniques**

Same tools used in System Testing

► **Stress**

Stress testing is conducted to evaluate a system or component at or beyond the limits of the specified requirements.

► **Owners**

Testing Individuals □

► **Tools/Techniques**

Load Runner and Jmeter

3.2.5 Load

Load testing simulates multi-user or multi-threaded access to an application or module to ensure that components and databases can be used to perform specified requirements with no catastrophic failures.

► **Owners**

Test Team consisting of and additional team members where available.

► **Implementation Approach**

The software will need to have some load testing done The
approach will be to have the databases reside on a single computer in the QA lab and have multiple users access the same database through the use of other computers in the QA lab. A designed test will be written and approved prior to any testing activity for this. The script will be written by the QA team and approved by the development staff.

► **Tools/Techniques**

QA test machines and RestAssureed Jmeter , LoadRunner
software, scripted scenarios for multiple users accessing the same database.
Found defects will be logged as such in the defect tracking tool

3.2.6 Performance

Performance testing is conducted to evaluate the compliance of a system or component's response time, and the ability to function in various operating environments.

► **Owners**

Test Team consisting of no of individuals and additional team members where available.

► **Implementation Approach**

The approach to this will be a manual testing of critical functions agreed on with the Development team. QA will do all the allocated tasks

3.2.7 Regression

Regression testing involves re-testing a previously tested program following modification to ensure that faults have not been introduced or uncovered as a result of the changes made. In this release this will be covered by the ongoing use of manual tests being executed after each successful build of the application, prior to release of the build for general testing use.

► **Owners**

Test Team consisting of and additional team members where available.

► **Implementation Approach**

The Test Team will do a pass through all the test scripts that were developed for this project. This will encompass the re-testing of each item in each test script as well as the re-verification of each repaired defect that is decided on as an items to be regressed based on the severity of the defect and the knowledge of the evelopment staff as to which areas of the application are the most volatile due to new features being implemented.

Additionally the re-test of the install and platforms will be done a final time. This must be done after the application is stable and considered Code Complete. Any defect found during this process must be determined to be "Must Fix" before release or deferred to the next release.

TTest Team will perform as many test scenarios as is feasible. Testing of the application will include limits and boundaries. The functional testing of the application will be covered this way.

Positive test cases will reflect that the application functions as expected. Negative test cases are tests that exercise the limits and boundaries outside the expected designs. The idea is that the application should be able to recover and / or

set error messaging as needed to accommodate this type of testing. This is important in this release as no design specification for the individual control objects exist other then the past functionality of releases not on the .Net framework. The results of this testing will give us some idea as to the design parameters and stability for the components.

3.2.8 Recovery

Recovery testing forces the failure of the software in a variety of ways to verify that recovery is properly performed.

- **Owners**

Test Team and additional team members where available.

3.2.9 Conversion

Conversion testing involves testing programs or procedures used to convert data from existing systems for use in Replacement systems.

- **Owners**

Test Team consisting of and additional team members where available.

- **Implementation Approach**

This release will cover conversion of Existing Cases, User Defined Data and Report Templates that were used in the past applications. The testing will be executed by using old and newly created datafiles all results should match. A ny differences will be reported as defects and will be addressed by the development staff.

3.2.10 Security

Security testing evaluates whether the system meets its specified security objectives by attempting to break in or disable a system by improper acquisition of a password, bypassing security measures, browsing through insecure data, or overwhelming the system with requests.

3.2.12 Documentation Verification

Documentation verification involves reviewing for accuracy all supporting User Documentation, Help Files, and supplemental materials.

- **Owners**

QA and Product manager

- **Implementation Approach**

Manual and using References

- **Tools/Techniques**

Word and Power Point

Testing Tools That can be Used

1. Selenium
2.Jenkins for continous integration
3.Jira for test and task management
4. Jmeter for load and performance testing
5.Mocha Chai and jasmine for Unit Testing
6. Cucumber for behavior Driven development Testing
7. TestNg and Junit for unit testing
8. SOAP UI for api and rest api testing
9.Apache Maven and Apache ant
10. Sauce labs for Cloud and cross browser testing
11.BrowserStack

4. Organization

4.1 Testing deliverables and Milestone

Deliverables and Milestones are as follows;

- Test scripts complete, signed off by the team managers.
- Data sets to be used for testing, (spread sheets and tables with client data designed to reflect the actual use by an XXXX client)

Milestones will be decided after release of the final project timeline

4.2 Roles and Responsibilities

Role	Assigned to	Responsibilities
QA Manager		Oversees QA processes for all projects.
Build Manager		Monitors and updates the automated build procedure for applications and components.
Test Lead Software Quality Engineer		Manages and tracks software system test planning and testing.
Software Quality Engineer		Execute test scripts and log defects. Validate repaired defects.
Software Quality Engineer		Execute test scripts and log defects. Validate repaired defects.
Software Engineer		Software developer
Software Engineer		XSoftware developer
Product Management		Product Management