**Revature Batch Scheduler**

**TEST PLAN**

|  |  |  |  |
| --- | --- | --- | --- |
| Document History | | | |
| Version | Date | Author | Description Of Change |
| 1 | 11/8/2016 | Yusuf Kassim | Draft |

**Table of Contents**

**1** [**INTRODUCTION**](#_1fob9te)

[1.1 Objectives](#_3znysh7)

1.2 Team Members

**2** **SCOPE**

**3** **ASSUMPTIONS / RISKS**

[3.1 Assumptions](#_tyjcwt)

[3.2 Risks](#_4d34og8)

[**4**](#_2s8eyo1) **TEST STRATEGY**

[4.1](#_17dp8vu) Test Stages:

[4.1](#_17dp8vu).1 Unit Testing

4.1.2System And Integration Testing

4.1.3Acceptance Testing

4.2Testing Types:

4.2.1 GUI Testing

4.2.2 Performance and Stress Testing

4.2.3Functional Testing

**5** **TEST ENVIRONMENT**

**6 Defect Mangement**

**7 Features to be Tested**

**8** [**MILESTONES / DELIVERABLES**](#_26in1rg)

8.1 Test Schedule

[8.2 Deliverables](#_35nkun2)

**Introduction**

The Test Plan has been created to communicate the test approach to team members. It includes the objectives, scope, schedule, risks and approach. This document will clearly identify what the test deliverables will be and what is deemed in and out of scope.

## Objectives

Revature Scheduler is a web application that allows Admins and Trainers, to access, create, and modify Batches. It also allows the user to view a list of Trainers, Locations, and Batches

## Team Members

|  |  |
| --- | --- |
| **Resource Name** | **Role** |
| August Duet |  |
| Yusuf Kassim | SDET |

1. **Scope**

This Project will include all the requirements that are defined in the Requirements Specifications. These and any other requirements that get included must all be tested. At the end of the project, a user must be able to effectively use and interact with the application.

1. **Assumptions / Risks**

## Assumptions

This section lists assumptions that are made specific to this project.

## Risks

The following risks have been identified and the appropriate action identified to mitigate their impact on the project. The impact (or severity) of the risk is based on how the project would be affected if the risk was triggered. The trigger is what milestone or event would cause the risk to become an issue to be dealt with.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Risk** | **Impact** | **Trigger** | **Mitigation Plan** |
| 1 | Scope Creep | High | Delays in implementation date | Each sprint, functionality will be closely monitored. Priorities will be set and discussed by team members. |
| 2 | Focus: This team has to develop the application before testing it, which may muddy the water between debugging and testing. | High |  | Testing objectives must be defined and implemented early. |

1. **Test Strategy**

The project is using an agile approach, with weekly iterations. At the end of each week the requirements identified for that iteration will be delivered to the team and will be tested.

Exploratory testing will play a large part of the testing as the application is being developed iteratively.

## Test Stages:

* + 1. Unit Testing.

Tools: **Junit, TestNG.**

* + 1. System and Integration Testing.

Tools: **Jenkins.**

* + 1. Acceptance Testing.

Tools: **Selenium WebDriver, TestNG.**

## Test Types:

* + 1. GUI Testing.

User Interface testing verifies a user’s interaction with the software. The goal of UI Testing is to ensure that the User Interface provides the user with the appropriate access and navigation through the functions of the applications.

|  |  |
| --- | --- |
| Test Objective: | Verify the following:   * Navigation through the application properly reflects functions and requirements. * Window objects and characteristics, such as menus, size, position, state, and focus conform to standards. |
| Technique: | Create / modify tests for each window to verify proper navigation and object states for each application window and objects. |
| Completion Criteria: | |  | | --- | | Each window successfully verified. | |  | |
| Special Considerations: |  |

* + 1. Functional Testing.

Testing of the application should focus on any target requirements that can be traced directly to use cases. The goals of these tests are to verify proper data acceptance, processing, and retrieval, and the appropriate implementation of the requirements. This type of testing is based upon black box techniques.

|  |  |
| --- | --- |
| Test Objective: | Verify that the application meets requirements. |
| Technique: | * Execute each use case, use case flow, or function, using valid and invalid data, to verify the following: * The expected results occur when valid data is used. * The appropriate error / warning messages are displayed when invalid data is used. |
| Completion Criteria: | |  | | --- | | All planned tests have been executed.  All identified defects have been addressed. | |
| Special Considerations: |  |

* + 1. Regression Testing.

Regression testing is used to insure that new additions do not break working features.

|  |  |
| --- | --- |
| Test Objective: | Insure that new features function without negatively impacting previous working features. |
| Technique: | Run tests with every Jenkins build. |
| Completion Criteria: | Successful build. |
| Special Considerations: |  |

* + 1. Performance and Stress Testing.

Performance testing measures response times, transaction rates, and other time sensitive requirements. The goal of Performance testing is to verify and validate the performance requirements have been achieved. Performance testing is usually executed several times, each using a different "background load" on the system. The initial test should be performed with a "nominal" load, similar to the normal load experienced (or anticipated) on the target system. A second performance test is run using a peak load.

|  |  |
| --- | --- |
| Test Objective: | Validate System Response times under a the following two conditions:  - normal anticipated volume  - anticipated worse case volume |
| Technique: | Modify data for the scripts created for UAT to increase number of transactions/Users/Iterations. |
| Completion Criteria: | Single Transaction / single user: Successful completion of the test scripts without any failures and within the expected / required time allocation (per transaction)  Multiple transactions / multiple users: Successful completion of the test scripts without any failures and within acceptable time allocation. |
| Special Considerations: |  |

1. **Test Environment**

To Be Added.

1. **Defect Management**

|  |  |
| --- | --- |
| **Severity** | **Impact** |
| 1 (Critical) | * This bug is critical enough to crash the system, cause file corruption, or cause potential data loss * It causes an abnormal return to the operating system (crash or a system failure message appears). * It causes the application to hang and requires re-booting the system. |
| 2 (High) | * It causes a lack of vital program functionality with workaround. |
| 3 (Medium) | * This Bug will degrade the quality of the System. However there is an intelligent workaround for achieving the desired functionality - for example through another screen. * This bug prevents other areas of the product from being tested. However other areas can be independently tested. |
| 4 (Low) | * There is an insufficient or unclear error message, which has minimum impact on product use. |
| 5(Cosmetic) | * There is an insufficient or unclear error message that has no impact on product use. |

1. **Features to be Tested**

* General Functionality: Login Page
* Trainer View.
* Admin View.

1. **Milestones / Deliverables**

## Test Schedule

The initial test schedule follows……….

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task Name** | **Start** | **Finish** | **Effort** | **Comments** |
| Test Planning | 11/4/16 | 11/8/16 | 4d |  |
| Review Requirements documents | N/A | N/A | N/A | Does not Exist. |
| Create initial test estimates | 11/4/16 | 11/8/16 | 4d |  |
| First deploy to QA test environment |  |  |  |  |
| Functional testing – Iteration 1 |  |  |  |  |
| Iteration 2 deploy to QA test environment |  |  |  |  |
| Functional testing – Iteration 2 |  |  |  |  |
| Regression testing |  |  |  |  |
| UAT |  |  |  |  |
| Resolution of final defects and final build testing |  |  |  |  |
| Deploy to Staging environment |  |  |  |  |
| Performance testing |  |  |  |  |
| Release to Production |  |  |  |  |

## Deliverables

|  |  |  |
| --- | --- | --- |
| **Deliverable** | **For** | **Date / Milestone** |
| Test Plan | Project Manager; Test Team |  |
| Traceability Matrix | Project Manager |  |
| Test Results | Project Manager |  |
|  |  |  |
| Metrics | All team members |  |
|  |  |  |