

|  |  |
| --- | --- |
| ROQ Main Contact Details: | <<Client>> Main Contact Details |
| * <<Name>> | * <<Name>> |
| * <<Title>> | * <<Title>> |
| * <<Telephone No>> | * <<Telephone No>> |
| * <<Email Address>> | * <<Email Address>> |

CredersiVend Admin

CredersiVendAdmin

14-12-2021

Test Plan

1

Contents

[1. Introduction 3](#_Toc90040310)

[1.1. Project Background 3](#_Toc90040311)

[1.2. Purpose 3](#_Toc90040312)

[1.3. Test Objectives 3](#_Toc90040313)

[2. Scope 3](#_Toc90040314)

[2.1. Test Scope – Inclusions 3](#_Toc90040315)

[2.1.1. Systems Under Test 3](#_Toc90040316)

[2.1.2. Features Under Test 4](#_Toc90040317)

[2.2. Test Scope – Exclusions 4](#_Toc90040318)

[3. Approach 5](#_Toc90040319)

[4. Acceptance Criteria 6](#_Toc90040320)

[4.1. Entry Criteria 6](#_Toc90040321)

[4.2. Exit Criteria 6](#_Toc90040322)

[4.3. Suspension Criteria 6](#_Toc90040323)

[5. Tasks and Deliverables 6](#_Toc90040324)

[5.1. Test Project Plan 6](#_Toc90040325)

[5.2. Test Milestones 6](#_Toc90040326)

[5.3. Test Deliverables 7](#_Toc90040327)

[6. Roles and Responsibilities 7](#_Toc90040328)

[7. Test Environment Needs 7](#_Toc90040329)

[8. Test Data Needs 8](#_Toc90040330)

[9. Staffing and Training Needs 8](#_Toc90040331)

[10. Test and Defect Management 8](#_Toc90040332)

[10.1. Test Management 8](#_Toc90040333)

[10.2. Defect Management 8](#_Toc90040334)

[11. Assumptions 8](#_Toc90040335)

[12. Constraints 9](#_Toc90040336)

[13. Risks 9](#_Toc90040337)

[14. Templates 10](#_Toc90040338)

[15. Document Control 11](#_Toc90040339)

[15.1. Document Review 11](#_Toc90040340)

[15.2. Change History 11](#_Toc90040341)

[15.3. Distribution 11](#_Toc90040342)

[15.4. Referenced Documents 12](#_Toc90040343)

[15.5. Document Sign-off 13](#_Toc90040344)

1. Introduction
   1. Project Background

The testing of the credersi vend application this application is a web application for engineers to find vending machines on the site they are visiting. Once on the chosen site the vending machines will appear in a route format allowing the engineer to find the next closest machine after servicing a machine. Also on the web application the users can also add information to each section if new machines get added for example this information can then be simply updated via the web application.

which will consist of three major sections this being the frontend which will be what users can see for example the web page and if this does work as intended. So testing like selenium can be conducted for this section of the application. The next section is the backend this a section that is gathering all the information from behind the scenes like the database for example which holds the information for the vending machines one example of testing that will be conducted will be functional testing this is performed keeping in mind an end-user point of view; whether the required transactions and operations run by the end-users meet the business specifications. The tests that will be done under functional testing will be white and black box testing. The last main section is the routes this section is the glue of the application which connects everything together via api’s for example so api testing will be a vocal point in this section.

* 1. Purpose

The purpose of the Credersi-Vend application to be tested is to ensure the end user can interact and use all components of the application according to the specification, encountering as minimal defects as possible. Given the time constraints and customer requirements, the scope of the project will be mapped to ensure all functionality is tested including all 3 of the base components, the front-end, back-end and route. Integration testing will be completed to test the interactions between each component, as well as some stress testing to ensure quality is at a level determined in the planning stage.

The Credersi-Vend application has a front-end UI that the end user will be adding data into the database, this UI will be tested using JavaScript to ensure all functionality works, allowing the end user to perform the actions that are required for the application to operate.

* 1. Test Objectives

State the major objectives of the testing.

The testing that to be undertaken will ensure that the main objectives are completed, and that the application works as per the expectation of the users and the stakeholders, the specified requirements will be fulfilled during the testing phase. Top objectives of the application are to test components and prevent defects from occurring.

Test the levels

Frontend – UI Testing (UAT Testing), Automation Testing eg(Selenium, Cucumber), Manual Testing

Backend – Functional Testing: Black box and White box testing, Database testing, maybe non-function testing.

Routes – Api Testing

1. Scope
   1. Test Scope – Inclusions
      1. Systems Under Test

Define the system under test. Summarise its constituent testable systems, sub-systems or component parts. Include versions where possible.

| Item | Purpose | Version |
| --- | --- | --- |
| Frontend | User Interface for end user interaction. The front end developed using Svelte. E.g. submitting data about a new vending machine. | 1.0 |
| Backend | Database side of application. Can serve the rest API and also serves up the web application (front end). | 1.0 |
| Routes | Lowest level and accesses the neo4J database. | 1.0 |
|  |  |  |

* + 1. Features Under Test

Summarise all the major features and combinations of features to be tested, include non-functional aspects and documentation references, e.g. Business Requirements. Include versions where possible.

| Feature | Sub-Feature | Reference Document and Version |
| --- | --- | --- |
| Login |  |  |
| RESTAPI | Data retrieval and submission to the database. |  |
| Neo4j Database |  |  |
| Web App UI | Static web page interfaces |  |
| Dynamic Web App Interface | Buttons/Submission features for the database. |  |
| Creating | Customer  Machines  Location / Site |  |
| Deleting | Customers  Machines  Location / Site |  |
|  |  |  |
|  |  |  |

* 1. Test Scope – Exclusions

Identify all items and/or software features excluded from the scope of testing, explain why. Note that this information is often more important than specifying what is in scope.

The exclusions within the test scope will likely be performance testing and security testing – due to the fact that the specification does not require these to be completed. The requirements document that was provided is broad - allowing all types of testing to be conducted according to the group's strengths, this is one of the reasons why security and performance testing are not going to be completed, as our strengths reside in other areas e.g., Selenium/UAT/JUnit/Automation.

1. Approach

We will be as proactive as possible in our approach but also be reactive to when certain things occur.

The method we will use is consultative, since we are not experts in the software we will need to question regarding the technical details. This will help us ensure that all technical aspects are thoroughly tested and working as intended. Also since the documentation given to us is extremely minimal it will be vital for us as a team to consult the SME to ensure we get all the information and data we need.

We will use various testing tools to ensure we cover as much as we can. We will use:

Visual studio code – Selenium, Cucumber and Gherkin for automation testing and Jest. These can be carried out to do User acceptance testing.

Eclipse – to carry out Component and Integration testing, this will be done in J-Unit

Neo4J –

Postman –

We will also test the application on multiple browsers to ensure that the application works as intended on multiple bowsers. We will ensure it is tested on Chrome, Microsoft Edge and FireFox.

If the plan covers multiple types or phases of testing then this section can be split into an Overall Approach and a separate specific approach for each of the testing types. Where the different types of testing will be handled in a very different way it is better to have multiple test plan documents instead.

Specify the major activities, techniques and tools that are to be used to test the items and features in scope.

The approach should be sufficiently detailed to enable identification of the major tasks and estimation of time to do each one. When developing the approach consider:

The test policy/strategy (if applicable)

Significant constraints on testing, e.g. test resource and/or environment availability, time, etc and the impact on the approach

Risk based testing

How much testing is required: too much is a waste of time and money, as is too little and the wrong testing is the worse for little or no gain

The type of industry

Contractual, legal, regulatory or specific customer requirements

Team experience

Resource availability

Documentation availability

Etc.

Define the test specification document requirements, how many specifications will be required, map these to the items/features identified as being in scope.

State the expected number of test cycles.

Indicate how test coverage and completion will be determined.

Define how the tests will be developed and test results captured. Describe how repeatability is assured, the extent of regression needs and how these are supported.

1. Acceptance Criteria
   1. Entry Criteria

Detail the entry criteria that must be satisfied prior to commencing the test execution. (Include consideration for the system under test handover requirements - how is its content and status defined and determined, how are known outstanding issues identified?).

For the test to commence we need to ensure that all the main fuinctionaliuty of the we application is working, and the web app connencts to the database.

We need to ask Floz - how are known outstanding issues identified.

* 1. Exit Criteria

Specify the exit criteria that will be used to judge the completeness of the test activity – the comprehensiveness and completeness of testing. This section addresses how the system under test will be deemed fit for purpose; when this point is reached the testing activity should stop.

How the system will be suitable for exit criteria, First of all have trust in the system that most defects have been spotted through the testing and the main three sections have been tested thoroughly.

The first major test area that needs to be complete is testing of the frontend. This will be completed via Selenium.

The second major test area that needs to be completed is testing of the backend. This will be completed via Junit.

The last major test area that needs to be completed is testing of the routes. This will be completed via Api testing.

* 1. Suspension Criteria

Specify the circumstances and criteria in which all or a portion of the testing activity might be suspended. Specify in what circumstances and how the test activity will be restarted and what tasks must be undertaken.

1. Tasks and Deliverable.
   1. Test Project Plan

Define the project plan which governs the test phases and activities, summary Gantt chart style, showing testing tasks, timescales, dependencies and milestones, and resource assignments. Show how this fits with the overall project plan.

* 1. Test Milestones

Document all the milestones associated with this test plan.

| Task | Milestone | Planning Date |
| --- | --- | --- |
| High level task, e.g. Test Build | The milestone being met, for example build complete | The initial planned date to hit this milestone |
| Completed Selenium tests |  |  |
| Completed Junit tests |  |  |
| Completed Rest Api tests |  |  |

* 1. Test Deliverables

Document all the deliverables associated with this test plan.

| Deliverable | Description | Task |
| --- | --- | --- |
| The physical item to be delivered, e.g. a Test Specification | Describe the deliverable in the context of the planned work, e.g. This document records the testable requirements for system x | The associated high level task, e.g. Test Analysis |
| Can user log in |  |  |
| Can create a customer |  |  |
| Can create a machine |  |  |
| Can add a location |  |  |
| Can delete customer |  |  |
| Can delete a machine |  |  |
| Can delete a location |  |  |
| When user clicks the V by the breadcrumb trail does it take them to the homepage. |  |  |
| When new created fields are added they should be in alphabetical order |  |  |
| Can the user progress through the route |  |  |
| Can the user go back a step if a mistake in the route taken |  |  |
|  |  |  |

1. Roles and Responsibilities

Define the roles which need to be undertaken to fulfil the plan, and list the responsibilities of that role. Also record who will undertake each role. Remember that each role can potentially be carried out by more than one person, and a single person can potentially undertake multiple roles.

| Role | Responsibilities | Person(s) |
| --- | --- | --- |
| Testers |  | Callum |
| Testers |  | Toby |
| Testers |  | Abdullah |
| Testers |  | Ade |
| SME |  | Floz |

Please note that a single member of staff may be able to undertake more than one role, subject to time constraints.

1. Test Environment Needs

**Java**

* JUNIT Testing for the components of the system
* Eclipse IDE for testing and execution of code

**JavaScript**

* VS Code – Compilation of code and use for Selenium

**Selenium**

* Cucumber – Software tool for UAT testing and BDD
* Gherkin - Language parser needed for Cucumber

**Postman**

* Manual testing for Integration (REST API)

The frontend UI for Credersi-Vend is an external

Where testing requires several environments with different attributes then it is advisable to split these out into separate sub headings.

Define the test environment and the support and controls needed. Identify any aspects which will need construction and build (e.g. harnesses or simulators).

Include non-computer system requirements e.g. rooms and other facilities.

Identify what exists and what needs to be procured or commissioned.

Make clear how peripheral devices might be allocated, shared and used.

Make clear how external interfaces to other systems or outside the organisation are supported.

Summarise any key administrative requirements and functions (e.g. backup/restore, batch execution and schedule management, printer management).

1. Test Data Needs

Describe the data strategy, how it supports the testing defined in this plan and how data will be built or generated.

Make clear how data is aged if this is required in the test approach.

Make clear how production data is sourced, and any timing or data protection issues associated with production data use.

1. Staffing and Training Needs

Identify the staff required to deliver this plan, indicate whether these resources are available and assigned, if not how they will be obtained, identify any specialist skills required.

Consider and identify any training needs to prepare the test team for the test activity, for example testing training, training in the system under test, and training in the use of a test tool. Ensure that any training activities are included in the test project plan.

In looking at the build of credersi-vend there were uses of rest apis this is a new concept for the team so this will need to be looked at by the team so tests can be conducted.

Apache Tomcat – Launches a web server used in the program could be useful to an understanding of what this is and if testing is useful for this.

1. Test and Defect Management
   1. Test Management

Describe how the work will be test managed. Most importantly this section needs to define what the reporting channels will be for the planned work, and specifically what will be reported. Explain if and how and test tools will be used to manage or coordinate the work.

We will use Trello to assign tasks for each other as it s a visual and once someone is assigned a task they are notified via email. For things which we all have to do we have created a shared document where we can all edit at the same time maximising the input.

* 1. Defect Management

Explain in broad terms how defects will be managed. Typically this involves stating the defect management tool to be used (if any) and a reference to the defect coordination document. Defect classifications and SLAs for defect resolution should be detailed here.

1. Assumptions

Record any assumptions used during the preparation of this plan. Assumptions are typically positive things which enable project work, but cannot currently be demonstrated or proved to be true. Also explicitly state what the impact to the plan would be if the assumption proved to be false or incorrect.

All assumptions need to be agreed by an appropriate authority, usually the client project manager (although some items are better confirmed by subject matter experts); record who agreed the validity of the assumption and when. The plan should not be signed off without all the assumptions first being explicitly agreed to.

| Description | Impact | Agreed By | Agreed Date |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Constraints

Constraints are things which do or definitely will restrict the way that work is carried out. Typically they are unequivocal and factual in nature. As well as stating a constraint it is essential to explain the impact this constraint will have on the testing in terms of efficiency, scope or risk.

The constraints need to be signed off by a suitable authority, typically the client project manager (though sometimes a subject matter expert is better placed to agree the constraints); the purpose of this is to ensure the client is aware of any constraints and so can potentially choose to do something about them.

| Description | Impact | Agreed By | Agreed Date |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Risks

A risk is something which might happen in the future and if it did would negatively impact the project. The risk is given numeric rating for impact (1-4) and likelihood (1-3). These numbers are multiplied together to generate the overall Risk Factor; the higher the risk factor the more effort should be invested to ameliorate the risk. Each risk needs an owner who is charged with monitoring the risk and taking proportionate steps to see that the risk does not occur. Alternatively, a low risk can simply be accepted by the project.

During the planning phase it is often possible to express risks as Assumptions or Constraints.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Impact | Likelihood | Risk Factor | Owner |
| SME going on holiday | 3 | 3 | 2 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. Templates

State which set of document templates will be used to support the project. This should be either the ROQ set of document templates, the client’s set of templates or some (defined) combination of the two.

1. Document Control
   1. Document Review

Record who has participated in both the internal and external reviews of the document. Where a person reviews a document multiple times it is only necessary to record the last date of review.

| Name | Role Title | Date |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

* 1. Change History

Record the different versions of the document which get distributed. Each new version of the document should receive a minor increment (e.g. from 0.01 to 0.02) unless the document is a major revision (e.g. the document has been signed off)(e.g. from 0.03 to 1.00). The level of information recorded in the Description of Changes field depends on the amount of sign-off received. Where external sign-off has been achieved the changes should be recorded in a high level of detail.

Not every version needs approval. Generally approval indicates passing internal review and therefore being made available to the client, or passing external review and being signed off by the client.

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Description of Changes | Approval |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

* 1. Distribution

List the people who the document will ultimately be circulated to. The roles are:

Review: The named person will review the document and feedback

Approval: The named person will approve the document on behalf of their organisation

Information: The names person may be interested in the document but is not reviewing or approving the document

|  |  |  |
| --- | --- | --- |
| Name | Organisation | Document Role |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

* 1. Referenced Documents

List all the documents referenced in the production of this document. Each item needs a number so it can be uniquely identified. The document title and version should be specified. Finally, it is important to record who or where the document is available from so a reader of this document can get copies of all the references documents.

|  |  |  |
| --- | --- | --- |
| Ref | Document and Version | Available from |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

* 1. Document Sign-off

This document has been reviewed, and approved for issue at the indicated issue status by the following:

<<Client>> Project Manager or Authorised Representative

|  |  |
| --- | --- |
| Name: |  |
| Position: |  |
| Signature: |  |
| Date: |  |

ROQ Test Project Manager or Authorised Representative

|  |  |
| --- | --- |
| Name: |  |
| Position: |  |
| Signature: |  |
| Date: |  |