

|  |  |
| --- | --- |
| ROQ Main Contact Details: | Credersi-Vend Main Contact Details |
| * Mr M | * Floz |
| * Academy Consultants | * System Developer |
| * 0799911155 | * 0799911144 |
| * contact@roqtesting.com | * [contact@credersi-vend.com](mailto:contact@credersi-vend.com) |

Credersi-Vend

2022-09-16

Credersi-Vend Admin Test Plan

Version 1

**Contents**

[1. Introduction 3](#_Toc113434289)

[1.1. Project Background 3](#_Toc113434290)

[1.2. Purpose 3](#_Toc113434291)

[1.3. Test Objectives 3](#_Toc113434292)

[2. Scope 4](#_Toc113434293)

[2.1. Test Scope – Inclusions 4](#_Toc113434294)

[2.1.1. Systems Under Test 4](#_Toc113434295)

[2.1.2. Features Under Test 4](#_Toc113434296)

[2.2. Test Scope – Exclusions 5](#_Toc113434297)

[3. Approach 5](#_Toc113434298)

[4. Acceptance Criteria 7](#_Toc113434299)

[4.1. Entry Criteria 7](#_Toc113434300)

[4.2. Exit Criteria 7](#_Toc113434301)

[4.3. Suspension Criteria 7](#_Toc113434302)

[5. Tasks and Deliverables 8](#_Toc113434303)

[5.1. Test Project Plan 8](#_Toc113434304)

[5.2. Test Milestones 9](#_Toc113434305)

[5.3. Test Deliverables 9](#_Toc113434306)

[6. Roles and Responsibilities 10](#_Toc113434307)

[7. Test Environment Needs 11](#_Toc113434308)

[8. Test Data Needs 12](#_Toc113434309)

[9. Staffing and Training Needs 13](#_Toc113434310)

[10. Test and Defect Management 14](#_Toc113434311)

[10.1. Test Management 14](#_Toc113434312)

[10.2. Defect Management 14](#_Toc113434313)

[11. Assumptions 15](#_Toc113434314)

[12. Constraints 16](#_Toc113434315)

[13. Risks 17](#_Toc113434316)

[14. Templates 18](#_Toc113434317)

[15. Document Control 19](#_Toc113434318)

[15.1. Document Review 19](#_Toc113434319)

[15.2. Change History 19](#_Toc113434320)

[15.3. Distribution 19](#_Toc113434321)

[15.4. Referenced Documents 20](#_Toc113434322)

[15.5. Document Sign-off 21](#_Toc113434323)

1. Introduction
   1. Project Background

Provide a high-level explanation of the project background, and ROQ’s involvement in the project.

Founded in 2009 in Bury St Edmunds, Credersi-vend supplies and operates Vending Machines that sell premium Suffolk based product to museums and other tourist sites of historical interest

Credersi-Vend is small family-firm operating using predominantly analogue and paper-based workflows and organisational methods. Recently, Credersi-Vend has identified a need to update and transition these workflows and processes to more modern techniques.

A small group within the company have begun diversifying into software development, with the intention of creating bespoke in-house software to transition their workflows over to digital environments and processes.

ROQ have been hired by Credersi to review their Credersi Vend Application.

* 1. Purpose

*The purpose of this document is to describe the scope and high-level approach for the work to be undertaken, along with supporting material on factors that will affect the testing*.

The purpose of this document is to gather information about what work will be carried out. This is to ensure everyone is working collaboratively is organised, given accurate information, confidence so that the tasks are known to all members involved. Additionally, it is important for all members to clearly identify what is tested and what is not as well as not required. So, that if any contingencies come up there is a course of action for everyone to undertake to ensure that the project is going as expected.

The components of the systems that will be needed to be tested will be the four test levels which will be acceptance level, system level, integration testing and component testing. This will ensure a moderate-high level of coverage relative to the time constraints for this test cycle.

This will be achieved by conducting tests using manual testing, automation testing such as selenium, Junit, Jest and postman API. The components of the system that will not be tested will be different accounts for logging in this is due to only one existing admin account being made available for use by the client Credersi.

* 1. Test Objectives
* Ensure that the main product requirements are met (user stories and other documentation), and where they are not these issues are documented
* Check to verify an overall high level of QA for the Credersi-Vend-Admin software
* Provide a high level of test coverage of the software
* Utilise different test types to levels to find as severe and high priority defects as is possible within the sprint period
* From a user perspective, ascertain if there is a good user experience
* Perform testing at different levels- component, system, integration, user acceptance
* **Overall objective of testing: to provide sufficient evidence to evaluate levels of confidence in the quality of the software.**
* Note: testing of Credersi-Vend Admin is not aiming to explicitly evaluate the levels of confidence, but instead create a body of evidence from which evaluation of the alpha version can then take place.
* To test all core user stories from the perspective of different user groups for the Credersi-Vend Admin tool
* To test that the Credersi Vend Admin application meets the required minimum level for the user to be able to use the tool- core functionality works as anticipated.
* To test and assess the overall quality of user will have a good experience using the application

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 Component | The frontend purpose is to ensure the user can interact with the system without facing failure. | 1.0.1 |
| Backend Component | The purpose of the backend folder is to store the frontend code so that the browser can request it based on the route. | 1.0.1 |
| Routes Component | The routes folder allows the backend to communicate with the database. | 1.0.1 |
| APIs | The purpose of the API is to enable companies to share data and functionality with external develops, business partners and internal departments. | 1.0.1 |

* + 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 |
| --- | --- | --- |
| Logging in page | Log In button and the username and password matches an existing account. |  |
| Responsive mobile |  |  |
| Creating new companies/sites/machines |  |  |
| View the companies/sites/machines, forwards and backwards |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

* 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.

Testing different user logins, as the admin login details are the only details set up.

1. Approach

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.

* Routes testing
  + JUNIT
* Frontend testing
  + Selenium with Cucumber
  + Selenium with Jest
* Integration
  + Backend testing with routes – postman API
* System
  + Functional
  + Non-functional (low priority)
* User acceptance
  + Cucumber and gherkin, user stories, Trello boards

The user acceptance testing will be a high priority for our team, due to the importance of the user experience for the Credersi Vend Application. For these tests we have assigned two days to do this, the goal is to start on 09/09/22 with an end date of 10/09/22. The user acceptance tests will all revolve around the frontend of the application, as that is what the user will see. This will involve some major features like the login page, the ability to create new companies/sites/machines, the ability to view all the information up and down the chain. The expected number of cycles will be two. The cycles will be split with one cycle focusing on using the cucumber/gherkin software and the other cycle using

Frontend testing will be a high priority for our team as the user experience is very important for the Credersi Vend application. This will be done using selenium to test the different components on the website. These will be decided with use of user stories so we can work out what will be needed to test to calculate the minimum required for a good user experience. These will be written on the Trello board so we can keep the testing process well documented to ensure we stay on track. Another tool that can be used to test the user stories would be cucumber and gherkin. There are two main constraints with this activity. The first being time and the other being the teams experience. Overall, we have nine days to complete the project from the planning stages to fully testing. This leaves us very little time to sort out any blockers that may crop up. Therefore, we are going to keep make sure we have a high level of communication within the group, to ensure we all keep up to date with each other. This will allow for the team to help each other out if one part starts to take longer than originally planned. Also, we will stick to a strict plan to ensure we finished everything we wanted to do with a set priority list. Another constraint is the teams overall limited experience due to still being in training. This may cause blockers if we get stuck writing scripts. We will tackle this by assisting each other and asking any questions if we begin to struggle. As well as asking any questions to the Credersi educators. Due to the small scale of the project risk-based testing will not be used for this project to determine the priority for the tasks, as we have our own order of which ones will be the most important to showcase to the rest of the team. As stated, before due to the small time scale the number of tests will have to be limited, however we want to achieve a minimum standard for the user experience.

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?).

The first entry requirement will be, to ensure the code is working and the website is working to be able to start testing on it. For the code this will be that the data base links up with the backend to allow the website to run. For the website, for it to work the user must be able to login and see the data on the website.

* 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.

The login screen will be simple, with it only needing for the user to test if login and log out is working, and if the website does not allow an incorrect username/password to work.

* 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.

One of the reasons for suspending testing, may be if the test scripts do not work. This will mean reviewing the test scripts to see where the problem may be. If we are unable to find the problem, then further assistance is required to get them to work. Once the problem is resolved then the tests can continue with the working test scripts being used. The plan may need to be changed if it takes a while to resolve the problem, with either more time being added to the task or less tests being run.

1. Tasks and Deliverables
   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 these 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 |
| Component tested | Build complete | 09/09/2022 |
| Integration tested | Build complete | 09/09/2022 |
| System tested | Build complete | 13/09/2022 |
| User acceptance tested | Build complete | 13/09/2022 |

* 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 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

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) |
| --- | --- | --- |
| Component tester | To test the frontend and backend components of the Credersi vend application | Alec, Avais, James |
| Integration tester | To test the backend components of the Credersi vend application | Alec, Avais, James |
| System tester | To test the functional and non-functional components of the Credersi vend application | Alec, Avais, James |
| User acceptance tester | To test the user acceptance components of the Credersi vend application | Alec, Avais, James |

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

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

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).

The test environment is created using Eclipse. This creates the Credersi vend website in which, we will be able to perform our tests on. As well as testing the scripts on Eclipse as well.

One of the requirements for the team is a room or space for us to work in, as we find as a team that we work better for collaborative work in person rather than online. However, if this is not possible for whatever reason a Microsoft teams call can be started and we will be able to work from home if required.

Currently we have the test environment all set up and ready to go, therefore we should have no blockers in the way. We will need to create the test scripts for the multiple test we have planned.

A we plan to be in the office, each member of the team will have access to multiple different peripheral devices. Most importantly having dual monitor set ups. As this is a small team there will be no need to share as each member will have their own. Each member has their own mouse and keyboard if they chose to use it, so again there will be no need to share. The testing requires no other peripheral devices.

All files are stored on OneDrive, and backup documents can be created. All the written documents have been shared to the group and each member can edit each one. GitHub

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.

Due to all the team still being in training, more development of the skills would ideally be needed. However, there is sufficient support mechanisms are in place to help support the team.

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.

A Trello board will be used to keep a track of all the test cases and their progress. This will allow the team to see exactly where we are with the testing, as well as being to assign tasks for the team. The test results will recorded on a spreadsheet and then committed onto GitHub in a deliverables folder, where they can be easily accessed as well as stored.

Postman API will be used to test the backend integration with routes, these results will be then recorded on a spreadsheet and committed on to GitHub.

Selenium will be used to test the frontend at the component level. Selenium will also be used to test the frontend at the system level, these results will be then recorded on a spreadsheet and committed on to GitHub.

Cucumber/Gherkin will be used to test the user acceptance, these results will be then recorded on a spreadsheet and committed on to GitHub.

* 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.

The GitHub issues tab will be used to record any defects found when testing Credersi Vend application

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 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 |
| --- | --- | --- | --- |
| James C- away 09-09-2022- 11-09-2022 | Minimal- weekend away, doesn’t impact on time | Alec & Avais | 2022-08-30 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 |
| Lost/corrupted files | 4 | 1 |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

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.

* ROQ Test Plan Template
* ROQ Test Strategy
* Microsoft Excel Simple Gantt Chart
* Trello Board Template

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 |
| --- | --- | --- |
| Steven Knowles |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

* 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: |  |