

|  |  |
| --- | --- |
| Contributors |  |
| * Joanne Greenhalgh |  |
| * Cat Orr |  |
| * Carl Dunn |  |
| * Mark Berryman |  |

Credersi-Vend

March 2023

PiggyBack Showcase Test Plan

Version 1

**Table of Contents**

[**Table of Contents** 2](#_Toc130374312)

[1. Introduction 3](#_Toc130374313)

[1.1. Project Background 3](#_Toc130374314)

[1.2. Purpose 3](#_Toc130374315)

[1.3. Test Objectives 3](#_Toc130374316)

[2. Scope 4](#_Toc130374317)

[2.1. Test Scope – Inclusions 4](#_Toc130374318)

[2.1.1. Systems and Features Under Test 4](#_Toc130374319)

[2.2. Test Scope – Exclusions 4](#_Toc130374320)

[3. Approach 5](#_Toc130374321)

[4. Acceptance Criteria 7](#_Toc130374322)

[4.1. Entry Criteria 7](#_Toc130374323)

[4.2. Exit Criteria 7](#_Toc130374324)

[4.3. Suspension Criteria 7](#_Toc130374325)

[7. Test Environment Needs 9](#_Toc130374326)

[8. Test Data Needs 11](#_Toc130374327)

[9. Staffing and Training Needs 11](#_Toc130374328)

[10. Test and Defect Management 12](#_Toc130374329)

[10.1. Test Management 12](#_Toc130374330)

[10.2. Defect Management 12](#_Toc130374331)

[11. Assumptions 12](#_Toc130374332)

[12. Constraints 12](#_Toc130374333)

[13. Risks 12](#_Toc130374334)

[14. Templates 13](#_Toc130374335)

[15. Document Control 13](#_Toc130374336)

[15.1. Document Review 13](#_Toc130374337)

[15.2. Change History 13](#_Toc130374338)

[15.3. Distribution 14](#_Toc130374339)

[15.4. Referenced Documents 14](#_Toc130374340)

[15.5. Document Sign-off 14](#_Toc130374341)

1. Introduction
   1. Project Background

Credersi-Vend is a family-owned business that supplies and maintains vending machines to other businesses. The business is looking to update their software and introduce a new frontend web application.

The software for test is made up of two elements; “Credersi-Vend Customers” and “Credersi Vend Admin.”

Credersi-Vend Customers is a legacy SQL relational database within a C# program. It can process queries using an ID, list customers and add them into the database, and perform a smoke test before clearing the database.

Credersi-Vend Admin is the new frontend web application designed to assist machine engineers with planning vending machine maintenance routes on site. This is the first software of its type in the business and will require extensive testing. It utilises a Neo4j Aura graph database.

The developer of the Vend Routes Java Library was employed on a short-term contract and is no longer with the business. It is possible to reach him via email, but responses could take time. However, the developer and product owner of the frontend web application are available for questions.

Roq has been hired to test both Credersi-Vend Admin and Credersi-Vend Customers in one two-week sprint. Roq testers will then present their work and quality findings in a presentation showcase on 2023-03-31.

* 1. Purpose

The purpose of testing is to provide sufficient evidence that details the level of confidence in software quality.

* 1. Test Objectives

The primary objectives of testing are:

* Ensure that the system meets the minimum requirements for a viable product.
* Minimise the presence of defects within the software (functional testing).

The secondary objectives of testing are:

* Create automated test cases to improve regression testing.
* Performance testing (add to).
* Save end user time.
* Evaluate usability of the products (non-functional testing).

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

|  |  |  |
| --- | --- | --- |
| **System** | **Features** | **Description** |
| Credersi-Vend Customers | Add | Allows for the entry of a customer in the database |
| Exit | Exits the command line program |
| List | Lists all previously entered customers |
| Query | Allows to find a customer from a ‘Company ID’ |
| Quit | Exits the command line program |
| Smoke | Creates 5 customers, lists them all then removes them from the Database. |
| Help/? | Lists the available commands for the console program. |
| Credersi-Vend Admin | Create Customer | Using the front end, this allows you to create a customer. |
| Create Site | This allows you to create a Site for an existing customer. |
| Create Machine | This allows for the creation of a machine at a specified site. |
| Routes | This allows for the user to see the next machine in the route. |
| ‘Breadcrumb’ Trail | This shows the user the path they have taken on a route. |

* 1. Test Scope – Exclusions

The scope of testing will not cover:

* User accounts: A single admin login has been provided to perform tests on. Testing on other types of user accounts is not within scope.
* With current team experience we would not be able to test security of the product in this sprint.

1. Approach

Test Management

* Completing daily stand ups:

As part of the team’s commitment to an agile development process, a daily stand up will be completed before commencing work on the project. This will cover the previous day’s activities, any blockers, planned activities for the day and any possible down time for the individual.

* There are currently no expected absences from the team. Any unexpected absences or emergencies will be reported as soon as possible to the team and product owner.

Test Levels

Component testing:

* Credersi-Vend Customers will be tested to confirm that all the create and query functions in the program.
* Credersi-Vend Admin will be tested to confirm that all CRUD functions work as intended.

Integration testing:

* Credersi-Vend Customers will be tested to ensure that the program is able to correctly interact with the SQL database and that information is added.
* Credersi-Vend Admin will be tested to ensure that the Svelte API, Neo4j Aura database and the Java code integrates with each other properly and interacts as intended.

System testing:

* Product requirements will be derived during the test process by the testing team. This will cover the primary functional objectives and some secondary non-functional objectives.
* Credersi-Vend Customers will utilise test data to simulate expected user activity and ensure that CRUD functions work correctly.
* Credersi-Vend Admin will have separate test data to simulate usage. Different test environments, such as different browsers and platforms, will be utilised.

Acceptance testing:

* Acceptance criteria will be derived from the product requirements generated by the testing team. Anticipated functional and non-functional requirements will be assessed against the software to see if it works adequately and that minimum viable product criteria has been met.

Test Types

After evaluating priorities of testing, this sprint will focus on functional testing of the software. Non-functional testing can be undertaken in a separate sprint at a later stage.

* White box testing:
  + Access to the code allows the creation of a wider variety of tests. Testing will primarily focus on functional testing.
* Black box testing:
  + Utilisation of user stories and user requirements to create tests that do not require knowledge of the code.
* Experience based on past projects:
  + Recent bootcamp experience will allow testers to utilise own experience.
* Backend Testing:
  + Postman, testing that the information comes from the API as expected. Specific testing around the API’s functionality with the code where there might be a higher concentration of defects.

Time Frame

* Test activities will be completed in one two-week sprint. Testers must be prepared to give a final presentation showcase of the work completed.

Legal and Contractual

* Credersi-Vend is a commercial supplier of vending machines. This is not sensitive in nature and does not have specific legal/contractual requirements. Standard data protection and contractual obligations will apply.

Documentation

A formal requirements document for the software has not been provided. Testers will produce a functional requirements document as tests are performed and functionality is elaborated on.

1. Acceptance Criteria
   1. Entry Criteria

The software development lifecycle is complete and the code is stable.

The software development process has been completed and the software is ready for testing.

The test plan has been completed by the testing team.

The test cases are completed.

The required test data has been created and is available to the test team.

The necessary testing tools are ready to use.

The environment has been set up and correctly configured.

The build has been deployed and is ready for testing.

* 1. Exit Criteria

All the test cases have been executed with all their results recorded.

All defects found have been reported.

Any documentation is completed and reviewed.

* 1. Suspension Criteria

If the test environment becomes unstable or unavailable, testing activities will be suspended until the environment is able to be used again.

If the system under test crashes or hangs during testing, testing activities will be suspended until the system is stabilised and the issue is resolved.

If an error occurs within the test data, testing activities will be suspended until the data is in a usable state again.

If any required resources are unavailable, testing activities will be suspended until the resources are obtained.

**5. Tasks and Deliverables**

5.1. Test Project Plan

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Date** | 17 March 2023 | 20 March 2023 | 21 March 2023 | 22 March 2023 | 23 March 2023 | 24 March 2023 | 27 March 2023 | 28 March 2023 | 29 March 2023 | 30 March 2023 | 31 March 2023 |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Phase** |  |  |  |  |  |  |  |  |  |  |  |
| Planning | Planning | | |  |  |  |  |  |  |  |  |
| Testing |  |  |  | Testing | | | |  |  |  |  |
| Reporting |  |  |  |  |  |  | Reporting | |  |  |  |
| Presentation |  |  |  |  |  |  |  |  | Presentation | |  |
| Showcase |  |  |  |  |  |  |  |  |  |  | Showcase |

5.2. Test Milestones

|  |  |  |
| --- | --- | --- |
| **Task** | **Milestone** | **Planned End Date** |
| Planning | When Test Plan and User Stories are completed. | 2023/03/20 |
| Testing | When all test cases are completed. | 2023/03/24 |
| Reporting | When defect reports are written up and peer-reviewed by the team. | 2023/03/28 |
| Presentation | When we have all the information in a presentable format ready for the showcase. | 2023/03/30 |

5.3. Test Deliverables

|  |  |  |
| --- | --- | --- |
| **Deliverable** | **Description** | **Task** |
| Test Case document | This document will contain:   * A table of test cases to be performed on the software under test * Test data to be used * Test steps * Expected and actual results * Pass/Fail | Test Recording |
| Defect report | This will record any defects found, their details and the events leading to them. Will include screenshots or other evidence to support reproducing the defect. | Test Analysis |
| Trello board | This will record and plan all user stories for both programs under test. | Test Planning |

**6. Roles and Responsibilities**

Due to the inexperience of the team, we will be working collaboratively on most tasks or as pairs.

Our team will be sharing roles and responsibilities such as:

Planning, which will include the creation of test cases, user stories and a test plan;

Testing activities which involve unit testing, system testing and acceptance testing;

Defect reporting which will track any defects found and the process leading to them;

Presentation of the testing sprint to relay the information to stakeholders.

1. Test Environment Needs

Each tester will use their individual laptop which will run the test environment.

Credersi-Vend Customers

Visual Studio Code will be used to run the console app for manual testing and will also be used to run N-Unit tests for component testing.

|  |
| --- |
| Version: 1.76.2 (user setup) |
| Commit: ee2b180d582a7f601fa6ecfdad8d9fd269ab1884 |
| Date: 2023-03-14T17:55:54.936Z |
| Electron: 19.1.11 |
| Chromium: 102.0.5005.196 |
| Node.js: 16.14.2 |
| V8: 10.2.154.26-electron.0 |
| OS: Windows\_NT x64 10.0.22000 |
| Sandboxed: Yes |

An SQLite database will be used for this app.

Stable release: 3.41.1 (10 March 2023;)

Credersi-Vend Admin

Eclipse will be used to start the backend Spring Boot server that will allow use of the app via the user interface for manual testing. It will also be used to run Junit tests for component testing.

|  |  |
| --- | --- |
| Version: 2022-12 (4.26.0) | Build id: 20221201-1913 |

|  |  |  |  |
| --- | --- | --- | --- |
| Eclipse.org - m2e | M2E Maven Integration for Eclipse Core | 2.1.0.20221015-0744 | org.eclipse.m2e.logback.feature |
| Eclipse.org - m2e | M2E Maven Integration for Eclipse Core | 2.1.2.20221130-2239 | org.eclipse.m2e.feature |
| Spring Tool Suite 4 Main Feature | VMware, Inc. | 4.18.0.202303131633 | org.springframework.boot.ide.main.feature.feature.group |
| Spring IDE Boot Microservices Dash | Spring IDE Developers | 4.18.0.202303131633 | org.springframework.ide.eclipse.boot.dash.feature.feature.group |
| DBViewer Plugin | ZIGEN | 1.2.2.v20101009 | zigen.plugin.future.feature.group |

A Neo4j database will be used for the Admin app.

|  |  |
| --- | --- |
| Stable release | 5.5 / 16 February 2023 |

Postman will be used for testing the API endpoints.

|  |
| --- |
| Postman v10.11.1 |

Browsers

|  |  |
| --- | --- |
| Google Chrome | Version 111.0.5563.65 (Official Build) (64-bit) |

1. Test Data Needs

* Credersi-Vend Customers:
  + A dataset containing only one customer
  + A dataset containing customers with a range of numbers of machines which will be used to test the category of a company
* Credersi-Vend Admin:

Test data will be created by the testing team. For Credersi-Vend Customers, this will be in the form of a separate SQLite database file which can be put into the appropriate folder. For Credersi-Vend Admin, separate domain nodes can be created in the Neo4J database which can be connected to via the backend component of the app.

1. Staffing and Training Needs

We are inexperienced team, meaning our knowledge is limited. This is part of our training.

Training on the application/system.  
  
Training for any test tools to be used.

1. Test and Defect Management
   1. Test Management

Testing activities will be managed using Trello. This allows the creation of cards for tasks and the ability to assign cards to individual testers.

We will be completing stand ups daily and will report to the product owner with any technical issues.

Teams will be used to share ideas and keep in contact throughout the sprint.

* 1. Defect Management

Defect management will be done using GitHub’s issue tracking tools.

1. Assumptions

| Description | Impact |
| --- | --- |
| All programs available to test. | Helps with speed of testing. |
| All equipment available for testing | Helps access programs and helps testing. |
| Internet connection strong | Helps with access |
|  |  |

1. Constraints

| Description | Impact |
| --- | --- |
| Developer of VendRoutes left company | Difficult to contact to ask questions. |
| Lack of documentation | Information on routes program lacking. Meaning difficult to understand all code. |
| Unable to download dependencies due to I.T security protocols. | Downloading certain dependencies blocked or very slow, slowing down testing process. |
|  |  |

1. Risks

.

|  |  |  |  |
| --- | --- | --- | --- |
| Description | Impact | Likelihood | Risk Factor |
| Internet going down | 3 | 1 | 3 |
| Illness | 2 | 1 | 2 |
| Holidays | 2 | 1 | 2 |
| Pregnancy | 1 | 1 | 1 |
| Too much activity on website | 3 | 2 | 6 |
| Lack of documentation | 2 | 4 | 8 |

1. Templates

The Roq test plan will be used to plan the project. We have also devised our own test case spreadsheet, which will be used to document test cases to complete. Trello will be used to make user stories and distribute the project caseload to participants.

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 |
| --- | --- | --- |
| **Reviewer** | **Role** | **Date last reviewed** |
| Mark Berryman | Tester |  |
| Carl Dunn | Tester |  |
| Jo Greenhalgh | Tester |  |
| Cat Orr | Tester |  |
| Laurence Martin | Product Owner |  |

* 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 |
| Version |  |  |  |
| 0.01 |  |  |  |
|  |  |  |  |
|  |  |  |  |

* 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: |  |