Construction 2

Testing Report

HexaTech

INFO3003A

Table of Contents: Construction 2

Executive Summary

This testing report for the Solarway Suppliers Repair Hub outlines the comprehensive testing approach used to validate the system's functionality, input validation systems, user interface design, and reporting accuracy and design. The purpose of this testing phase was to ensure that the system performs as expected and required, which has been outlined in the design and analysis phases. The system must also

support key business operations, have a functional reporting function, and deliver a user-friendly experience.

To begin, a high-level overview of the client, and the task at hand is given. This outlines what exactly is being tested, and why it is being tested. This is followed by objectives of the testing report, outlining what this document is attempting to achieve, and then the tasks, which are ways that these objectives are going to be achieved. A scope is then given, which lists specific functions of the system, that are to be tested to confirm the functionality and design of the system, the scope includes tactics for testing these specific functions – essentially specific tests to be run. A small risk analysis is then provided, detailing important vulnerabilities in the testing phase, and tactics to mitigate them. Justification for each type of test run is then given. This provides the reader further context as to why each test was chosen, detailing to important and useful information each test provides to the reader.

The actual testing phase follows, with each section linked to a specific test, detailing who undertook the test, what the methodology for the test is, and the actual testing results. This format follows for all the testing choices: functional testing, use-case testing, input field testing, user interface design testing, and report testing.

Additionally, a plan is given for what these testing results are going to be used for, going into the next and final iteration. This includes how the testing findings will be used to improve the system, through bug fixes and optimisations, and what tests are going to be run in the next iteration, Construction 2. This plan ensures a reliable, user-friendly system is delivered for final deployment.

This document also contains the second phase of the testing procedure, which took place after changes have been made to address the bug fixes found in the initial testing phase. This second phase documents exactly what happened with the results of the initial testing procedure and how they were used to improve the system – the purpose, objectives, and tasks thereof.

Justifications were provided for undertaking a re-testing of failed tests as well as regression testing to ensure the core functionality of the system remains intact following the changes made to the system. It also includes a methodology for these 2 methods of testing, and the documented results.

Last, a final roundup is provided which indicates that all tests now passed as the bugs have been adequately fixed without affecting the functionality of the system. This ends with the conclusion that the system is ready for the final deployment.

Name of the Product

Solarway Suppliers Repair Hub

Prepared By:

Yaaseen Hassim, Muhammad Bodhania, Lebohang Sebiloane – For 28 October 2024

Introduction

Purpose (Mission)

The mission of this test plan is to ensure that the Solarway Suppliers Repair Hub system functions as intended and set out as according to the analysis and design of the requirements in collaboration with the team at Solarway Suppliers. This will be achieved through identifying, describing and documenting inputs and results from several different types of tests. Through this comprehensive and structured testing approach, Hexatech aims to deliver a reliable, stable, functional, and user-friendly application that meets the requirements of the key stakeholders.

The purpose of this testing plan is to highlight and document the information gathered during the testing stage of this application. This document further details the different methods of testing, including a description, a justification of the respective tests, and a procedure of how they are undertaken. Results of the various tests are then documented, and a plan for how these results will be addressed then follows.

Overview

Business Overview

Solarway Suppliers is a solar wholesale business located in Greenside, Johannesburg. The business sells solar backup system components, such as inverters, lithium-ion batteries, solar panels, backup light systems, and more. The target market of the business is aimed more towards the wholesale market, supplying other smaller solar businesses, and installers of solar backup systems; they do, however, sell to individuals for personal use as well, although on a much smaller scale. The business consists of two departments: the sales department and the repairs department.

Application Overview

The Solarway Suppliers Repair Hub caters to the repairs department with the business. The current system being employed by Solarway Suppliers is inefficient, and at times, ineffective. The Repair Hub system aims to rectify these problems by digitising the repairs process, thereby making it efficient and effective.

Objectives and Tasks

Objectives

The objectives of this test plan are as follows:

- Ensure functionality of the system: The system must be able to perform all the high-level functions set out in the documentation and as specified by the requirements of Solarway Suppliers, the key stakeholder.
- Identify and eliminate defects: This test plan serves to detect and document any bugs, errors, or inconsistencies in the system and ensure that a plan is drafted to resolve these issues before final deployment.
- Ensure use case functions: The test plan has the objective to ensure that every applicable use case within the system functions as intended.
- Confirm usability: To validate that the system is intuitive and easy to use for users, ensuring that they can navigate the system seamlessly.

Tasks

The tasks of this test plan are as follows:

- Choice of testing ideas, including justification for the selection of tests.
- Test planning: designing tests around what needs to be tested.
- Test case design: developing detailed test cases to ensure tests are carried out in a consistent and accurate manner.
- Test execution and documentation: running the test cases and documenting the outcome of the tests.
- Plan for next iteration: detailed plan on how the test results will be used going into the next iteration and what tests will be run in said iteration.
- Final validation and sign off: conduct final check and sign-off to confirm that the test plan meets all its objectives, and that the system is one step closer to final deployment.

Scope of the Testing

General – What is being tested?

The following will be tested:

- Key functions of the 2 actors.
- All use cases specified.
- Input validation.
- Navigation and usability of the system.
- Consistent interface design.
- Aesthetic and consistent report design.

Tactics – How will this be tested?

- Functional Testing: this will be tested by ensuring that they key goals of the administrator and technician, specified in the documentation, can be achieved within the system.
- Use Case Testing: this will be tested by ensuring that all use cases specified in the documentation work and fulfil the expected and intended results – this also tests the database manipulation, ensuring that the system communicates with the database correctly and manipulates database records, or adds new ones in an accurate manner.
- Input Field Testing: testing to ensure that input fields within the system adequately validate data entered by the user to minimise input error.
- Interface testing: testing to ensure that the system and the navigation is designed in an intuitive manner and that the interface design is consistent and aesthetically pleasing. This will be tested by going through the interface across the pages.
- Report Testing: testing the reporting functionality of the system to ensure that is accurate, and also to ensure that the report design is aesthetically pleasing.

What will not be tested?

The decision has been made not to test the following:

- The webpages that are used for Read Use Case, such as Read Staff, Read Part, etcetera. This is as these pages are simple pages built on the functionality of the Update Pages, therefore by testing the Update functionality of each entity, we also, in part, test the Read functionality.
- Browser compatibility will not be tested, this is because all devices in the Repair department of Solarway Suppliers are standardised and run Google Chrome.

Risk Analysis

It is important to consider the risks that are inherent in the testing of the system. Here, the risks will be listed, and a potential mitigation plan will be developed for each one:

- Incomplete test coverage: if certain functions of the system are not tested, bugs or potential issues could go unnoticed. To mitigate this, it is imperative that comprehensive test plans are created to cover key areas of the program, ensuring all functionality requirements are present and working.
- Time constraints: the system is being tested in a very limited period, this could result in rushed tests, could leave the system incomplete. To mitigate this, prioritisation of important, high value test cases is imperative, ensuring that the core functionality is working first and foremost.
- Miscommunication between testers and development team: if there is miscommunication between the testers and the development team, important bugs discovered in testing could be left unfixed, or misinterpreted. To mitigate this risk, constant, clear communication between the teams is required, including regular review meetings.
- Bias in testing: there is a significant risk, due to the development nature of this project, that testing results are biased, due to a conflict of interests. To mitigate this, it is imperative to have the developers separate from the testing team, and to ensure no overlap.

Justification for Testing Choices

Functional Testing

Functional Testing validates that the core, high-level and critical functions of the system are operating as intended, enabling the two actors, the administrator and the technician, to successfully achieve their goals using the system. If this was not tested, there is a risk that users may face barriers in performing critical tasks, leading to abandonment of the system before it even gets going.

The purpose for the development of the Repair Hub system is to replace the manual, ineffective, and inefficient systems being employed by Solarway Suppliers today, and testing to ensure that the system can achieve its high-level goals first and foremost is critical to the success of the system. By testing core functions early, any potential issues that can disrupt day-to-day operations of the repairs department of Solarway Suppliers can be detected and plans to rectify them can be drawn up.

Functional testing verifies that the system adheres to the main functional requirements set out by the key stakeholder and documented during the design phase. It is crucial to ensure that the system matches what was agreed upon in the project specifications and that the system delivers what is expected.

Use Case Testing

Use case testing builds on from the functional testing previously done, by testing as many use cases as possible. Each use case links to a webpage within the web application, and as such, by testing these use cases (therefore webpages), we ensure proper functionality of the entire system, including both the core functionality, which was explored in the functional testing, and the maintenance use cases.

Use case testing makes sure that the entire application, from end-to-end functions is verified and working as expected. It ensures that all elements of the system, throughout the entire repairs process, is functioning

correctly. Each use case represents some element of the process of repairs, and by testing these use cases, we can verify that the system can replace the current outdated, inefficient, and ineffective system.

Use case testing also mimics the real functioning of the system, with the tester taking on the role of an employee within Solarway Suppliers. By testing each use case, it is ensured that every interaction a user might have with the system leads to the expected outcome, confirming that the system meets the needs of its users in practical situations. Finally, use case testing helps catch critical errors on every tested webpage before the role out of the system, and therefore improves user experience and improves overall system quality.

Input Field Testing

The Repair Hub system for Solarway Suppliers is a complex, internal system that consists of many individual webpages, each with their own user inputs. Input field testing is crucial to regulate these inputs to minimise human error with regards to inputting data.

Input field testing ensures that only valid data is accepted to begin with, protecting the integrity of the system, and preventing processing errors due to human error. Without this testing, users could submit incorrect or incomplete information, leading to errors in the system's functionality. It also ensures that data stored in the database is clean and accurate, which is crucial for reporting functionality.

Input field testing also improves user experience, as users are shown clear error messages, triggered by input validation, which helps them understand their mistakes. Lastly, input field testing also ensures that user input aligns with specified requirements, for example, cell phone numbers being digits only, names only having letters, and other requirements for input.

Interface Design Testing

Solarway Suppliers prides itself on a good brand reputation and image, and as such, even though the Repair Hub application is an internal application, it is imperative that the system's interface is user-friendly. Successful and user-friendly interface design also means that it is consistent and intuitive.

Consistency across the system design, in elements such as font, colours, etc. enhances the user experience, making the system easier to navigate and reduces cognitive load. Having intuitive design also supports efficient task completion, this carries over to efficient navigation design, which is important to system functioning. Users being able to easily navigate the system enhances the efficiency of said users as they can get to their desired webpages easily due to the intuitiveness of the navigation system.

A major reason for using Interface Design Testing is stakeholder satisfaction. The design of the system is the first thing that a user experiences when they load up the webpages, and as such, by testing this, we can ensure that the user experience is a satisfactory one, and is polished, this also prevents risk of the system being abandoned early on due to ineffective and inefficient interface design.

Report Testing

One of the major requirements and specifications for the Repair Hub system is an in-depth and useful reporting functionality. With this important requirement in mind, it is imperative that a test is done on the reports in the system.

Report testing is critical to ensure that the reporting functionality within the system is both accurate and aesthetically pleasing. Reporting is a key output of the system, and one that the management at Solarway Suppliers will use to make key, informed decisions to better the business. It is therefore imperative that the data used in the reporting functionality is absolutely accurate, and that the business logic is correct. Testing will also ensure that the reports have valid filtering options, such as filtering by date.

Additionally, having aesthetically pleasing reports will improve usability, and ensure that the reporting functionality of the system is used, and not simply abandoned because of poor visual design. With well-designed, intuitive and accurate reports, stakeholders can confidently make informed decisions to optimise the business.

Environment Requirements

To ensure a constant and accurate testing environment, certain requirements with the regards are needed.

Facilities Required

The testing took place in the individual's own testing environment, this was achieved as the source code and execution files for the system were sent to the respective testers, this includes the PowerBI report file and MySQL database file. With these files, the testers could test the system anywhere they wished.

Hardware Required

Each tester used their own personal laptop device, and an internet connection was required.

Software Required

The software required was as follows:

- Microsoft Visual Studio Code – this application is used for the source code and is required to launch the code.
- Node.js – an add-on to Visual Studio Code that allows the system to run as a web-based application.
- MySQL – the database software was required to ensure the system is accurately updating the information in the backend storage. In actual running, the database system will not be required, but for the purpose of testing, it was employed.
- PowerBI – this reporting software was required as it was used to compare what the reports should look like, to what they actually look like in the system reporting page.

Actual Testing

Functional Testing

Participants

This test will be performed by Yaaseen Hassim across different functions of the system on 6 October 2024.

Methodology

Over the lifecycle of this development project, Hexatech has undertaken the analysis and design of the system for Solarway Suppliers. With this prior knowledge and experience, combined with the documentation of all previous milestones, a list was compiled of what each of the system's two main users', the administrator and the technician, main goals are within the system. This information was gathered mainly from the use cases and models documentation.

Following the creation of this list of high-level goals of the system's actors, each goal was mapped to a system functionality. This allows for comprehensive testing to take place by testing certain system functionalities. An example of this is as follows: one of the administrators' high-level goals is to check new

repairs into the process following a customer's request – the functionality that maps to this is the create repair use case.

With the knowledge of which system functionalities need to be tested, test cases were roughly drafted for each one. A test case is a set of steps to be undertaken that lead to an expected outcome, which if achieved, means the test is passed.

With the list of test cases to go through, the next step was to execute these test cases in the step-by-step manner documented with the test case. Each test case starts from the home page in the system, ensuring that the user can navigate to the required functionality and execute it, achieving the expected outcome.

The results of each functional test case were then documented. The documentation includes the following details: Test Case ID, the steps executed (as specified in the test case), the expected outcome, pass or fail status, and additional comments.

Functional Testing Documentation

Each of the user's requirements will be tested and documented, as specified in the table as follows:

Actor: Administrator

Test ID	Function	Testing steps	Expected outcome	Pass/fail	Comments
F001	Checking repairs into the system.	1. Navigate to the create repair page. 2. Enter and select the required data. 3. Submit data.	The system should accept the data and save the repair in the database to begin the repairs process.	☑Pass	This functional requirement passes, the administrator can save the repair in the system with no hiccups.
F002	Updates repair status according to the repairs process.	1. Navigate to the update repair page. 2. Select the repair to be updated. 3. Select a new status. 4. Submit data.	The system should update the selected repair in the repair table with the new selected status.	☑Pass	
F003	Record stock of newly ordered parts.	1. Navigate to the update part page. 2. Select the respective part. 3. Update the quantity to the new level. 4. Submit the data.	The system should update the chosen part with the new quantity in the database.	☑Pass	
F004-1	Add new customer.	1. Navigate to the create customer page. 2. Enter the required data such as customer name, email, contact number, etc. 3. Submit the data.	The system should create a new customer record in the database and save the required details provided.	☑Pass	
F004-2	Update customer.	1. Navigate to the update customer page. 2. Select the customer to update. 3. Enter the new data. 4. Submit the data.	The system should update the respective customer record in the customer table with the newly entered data.	☑Pass	
F005-1	Add new staff member.	1. Navigate to the create staff page.	The system should create a new staff	☑Pass	

		2. Enter the required data such as name, surname, role, etc. 3. Submit the data.	record in the staff table and save the required details provided in said record.		
F005-2	Update staff member.	1. Navigate to the update staff page. 2. Select the staff member to be updated. 3. Enter the new data. 4. Submit the data.	The system should update the correct staff record with the newly entered data.	☑Pass	
F006-1	Create new manufacturer.	1. Navigate to the create manufacturer page. 2. Enter the required data for the manufacturer. 3. Submit	The system should create a new manufacturer record in the manufacturer table and save the required details provided in said record.	☑Pass	
F006-2	Update manufacturer.	1. Navigate to the update manufacturer page. 2. Select the manufacturer to be updated. 3. Enter the new data. 4. Submit the updated data.	The system should update the correct manufacturer record with the newly entered data.	☑Pass	
F007-1	Add new unit.	1. Navigate to the create unit page. 2. Enter the required data for the new unit. 3. Submit data.	The system should create a new unit record in the unit table with the attributes reflecting the data entered.	☑Pass	
F007-2	Update unit details	1. Navigate to the update unit page. 2. Select the unit to be updated. 3. Enter the new data. 4. Submit.	The system should update the selected unit record with the newly entered data.	☑Pass	
F008	Assign Job.	1. Navigate to the create job page. 2. Enter and select the required data for the job. 3. Submit the data for the job.	The system should create a new job ticket in the job table and save the respective data in the record for that ticket.	☑Pass	

Note: F005, F006, and F007 reflect the administrators' tasks of maintaining customer, staff, manufacturer and unit details.

Actor: Technician

Test ID	Function	Testing steps	Expected outcome	Pass/fail	Comments
F003	Record stock of newly ordered parts.	1. Navigate to the update part page. 2. Select the respective part.	The system should update the chosen part with the new	☑Pass	

		3. Update the quantity to the new level. 4. Submit the data.	quantity in the database.		
F008	Assign job	1. Navigate to the create job page. 2. Enter and select the required data for the job. 3. Submit the data for the job.	The system should create a new job ticket in the job table and save the respective data in the record for that ticket.	☑Pass	
F009	Assign hours to job	1. Navigate to the update job page. 2. Select the job to add hours worked to. 3. Enter hours worked. 4. Submit.	The system should update the job record in the job table with the hours worked on said job.	☑Pass	
F010	Updates status of ongoing repairs.	1. Navigate to the update repair page. 2. Select the repair to change status of. 3. Select new status. 4. Submit.	The system should update the status of record in the repair table with the newly selected status.	☑Pass	
F011	Assign parts to a repair.	1. Navigate to create RepairPart 2. Select the part to be assigned. 3. Select the repair to be assigned to. 4. Submit.	The system should record the part and repair together in the RepairPart table.	☑Pass	

Note: F003 and F008 overlap as both technicians and administrators have that function.

Use Case Testing

Participants

This test was performed by Yaaseen Hassim on 6 October 2024.

Methodology

Throughout the life cycle of this development project, through analysis and design, various use cases have been drafted and edited through the different milestones based on the scope and requirements of the key stakeholder, Solarway Suppliers. This cycle has resulted in a solid set of use cases that make up the entire functionality of the Repair Hub system and Use Case Testing involves testing these use cases that make up the system.

Each webpage of the system represents a use case, and the methodology of this testing procedure seeks to test these webpages, ensuring that they are functioning correctly, and the expected outcome is the actual outcome. Firstly, a list of the webpages was drawn up, ensuring that said list is comprehensive. Following this, for each webpage, and therefore use case, the expected outcome is written down. The test then takes place, and the results are noted, if the actual outcome is the same as the expected outcome, then the test passes, if not, it fails, and a reason is noted down.

Steps have not been drafted for use case testing as every webpage reflects a use case test, and the testing aims to also check if the user can use a webpage intuitively with what is in front of them.

Input Field Testing

Participants

This test was performed by Yaaseen Hassim on #DATE#

Methodology

The Input Field Testing involves running tests across the different user input entry points across all the webpages, the Solarway Suppliers Repair Hub application consists of many different webpages, all serving important purposes, whether as core use cases, or maintenance-based use cases, and as such, keeping user input error to a minimum is vital. Input field testing helps in this regard.

The first step was to identity the input fields, a list was made of the different input fields across the application. Next, validation rules for each input were drafted, many input fields are limited already and have built in validation rules, such as comboboxes with values taken from the database. Next, the test was executed for each input where validation rules apply, two tests were undertaken, a negative and a positive test to ensure that the system accepts the correct type of input and rejects the incorrect type of input, negative testing includes edge testing and testing for empty inputs. Results were then documented down.

Use Case and Input Field Testing Documentation

The Use Case Testing and Input Field Testing are combined in the following documentation, for each webpage, and therefore use case. They have been combined for easy comprehension as tests have been done per webpage.

Create Repair Page

Use Case Test

Test ID	Test Case	Test Data	Expected Results	Pass/Fail	Comments
U001	User creates a new in-house repair.	• Customer: Bongani • Unit: 12KW Deye • Description: Voltage • Check-in Date: 2024/10/06	Repair is submitted and saved in the repairs table in the system database	☑Pass	
U002	User creates a new manufacturer repair.	• Customer: Bongani • Unit: 12KW Deye • Description: Voltage • Manufacturer: Trina Solar • Check-in Date: 2024/10/06	Repair is submitted and saved in the repairs table in the system database	☑Pass	

Input Field Test

Test ID	Field Name	Test Data	Expected Results	Pass/Fail	Comments
I0001	Customer	Taken From Combobox	Accurately load customers from the customer table	☑Pass	
I0002	Unit	Taken From Combobox	Accurately load units from the unit table	☑Pass	
I0003	Description	Taken From Combobox	Accurately load description from the set options	☑Pass	Set options are available for this entry.
I0004	Manufacturer	Taken From Combobox	Accurately load manufacturers from the manufacturer table	☑Pass	
I0005	Check in date	Date Time Picker	Accurately shows day, month, and year	☑Pass	Validation for empty input

Update Repair Page

Use Case Test

Test ID	Test Case	Test Data	Expected Results	Pass/Fail	Comments
U003	User updates a repair.	• Repair: 2 VOLTAGE problem for Rose Peterson • Status: COMPLETED	Accurately updates the selected repair with the new status change in the repair table	☑Pass	Validation for empty input

Input Field Test

Test ID	Field Name	Test Data	Expected Results	Pass/Fail	Comments
I0006	Repair	Taken From Combobox	Accurately load repairs from the repair table	☑Pass	
I0007	Status	Taken From Combobox	Accurately load description from the set options	☑Pass	Set options are available for this entry.

Create Job Page

Use Case Test

Test ID	Test Case	Test Data	Expected Results	Pass/Fail	Comments
U004	User creates a job.	• Staff: Michael Smith • 2 VOLTAGE problem for Rose Peterson • Job hours planned: 2	Job is submitted and saved in the job table in the system database	☑Pass	Validation for empty input

Input Field Test

Test ID	Field Name	Test Data	Expected Results	Pass/Fail	Comments
I0008	Repair	Taken From Combobox	Accurately load repairs from the repair table	☑Pass	
I0009	Staff	Taken From Combobox	Accurately load staff from the staff table	☑Pass	
I0010	Job Hours Planned	Spin Edit	Ability to pick number using arrows.	☑Pass	

Update Job Page

Use Case Test

Test ID	Test Case	Test Data	Expected Results	Pass/Fail	Comments
U005	User creates a job.	• Job: Michael fixing VOLTAGE problem for Rose Peterson	Job is updated and saved in the job table in the system database	☑Pass	Validation for empty input

		• Actual hours worked: 2			

Input Field Test

Test ID	Field Name	Test Data	Expected Results	Pass/Fail	Comments
I0011	Job	Taken From Combobox	Accurately load jobs from the job table	☑Pass	
I0012	Actual Hours Worked	Spin Edit	Ability to pick number using arrows.	☑Pass	

Create Part Page

Use Case Test

Test ID	Test Case	Test Data	Expected Results	Pass/Fail	Comments
U006	User creates a part	• Part Name: Anti-Reflective Coating • Quantity: 10	A new part is submitted and saved into the part table in the system database	☑Pass	

Input Field Test

Test ID	Field Name	Test Data	Expected Results	Pass/Fail	Comments
I0013	Part Name	“TestPart”	Save part in table.	☑Pass	
I0014	Part Name	“”	Display error as part is left empty	☑Pass	
I0016	Quantity	Number picker	Ability to pick number using arrows.	☑Pass	

Update Part Page

Use Case Test

Test ID	Test Case	Test Data	Expected Results	Pass/Fail	Comments
U007	User updates a part	• Anti-Reflective Coating • Quantity: 9	Part is updated and saved in the part table in the system database	☑Pass	

Input Field Test

Test ID	Field Name	Test Data	Expected Results	Pass/Fail	Comments
I0017	Part	Taken From Combobox	Accurately load parts from the part table	☑Pass	
I0018	Quantity	Spin Edit	Ability to pick number using arrows.	☑Pass	

Create RepairPart Page

Use Case Test

Test ID	Test Case	Test Data	Expected Results	Pass/Fail	Comments
U008	User creates a repair part	• Repair: 13 FUSE problem for Ben Benzino • Part: Diodes	Repair part submitted and saved in repairpart table in the system database	☑Pass	

Input Field Test

Test ID	Field Name	Test Data	Expected Results	Pass/Fail	Comments
I0019	Repair	Taken From Combobox	Accurately load repairs from the repair table	☑Pass	

I0020	Part	Taken From Combobox	Accurately load parts from the part table	☑Pass	

Create Customer Page

Use Case Test

Test ID	Test Case	Test Data	Expected Results	Pass/Fail	Comments
U009	User creates a customer	• First name: Steve • Last name: Minecraft • Email address: steveminecraft@gmail.com • Contact number: 081 254 6475 • Physical address: 123 Street Johannesburg 2109	Customer submitted and saved in customer database in the system database	☑Pass	

Input Field Test

Test ID	Field Name	Test Data	Expected Results	Pass/Fail	Comments
I0021	First Name	“TestName”	Accept first name	☑Pass	
I0022	First Name	“”	Display error as input is left empty	☑Pass	
I0023	First name	“Test123”	Display error as input includes numbers.	☒Fail	Input should be rejected as names should not contain numbers.
I0024	Last Name	“TestName”	Accept last name	☑Pass	
I0025	Last Name	“”	Display error as input is left empty	☑Pass	
I0026	Last Name	“Test123”	Display error as input includes numbers.	☒Fail	Input should be rejected as names should not contain numbers.
I0027	Email Address	“test@gmail.com”	Accept email address	☑Pass	
I0028	Email Address	“”	Display error as input is left empty	☑Pass	
I0029	Email Address	“Test.com”	Display error as email is invalid	☑Pass	@ is required
I0030	Email Address	Test@gmail	Display error as email is invalid	☑Pass	.com or. co.za, etc. is required for a valid email
I0031	Contact Number	0828001234	Accept contact number	☑Pass	
I0032	Contact Number	“”	Display error as input is left empty	☑Pass	
I0033	Contact Number	“ABC”	Display error as contact number is invalid	☑Pass	
I0034	Contact Number	“1234”	Display error as contact number is invalid	☒Fail	Input should be rejected as the number should be 10 or 11 digits long.
I0035	Physical Address	12 Sandton Drive	Accept physical address	☑Pass	

I0036	Physical Address	“”	Display error as input is left empty	☑Pass	

Update Customer Page

Use Case Test

Test ID	Test Case	Test Data	Expected Results	Pass/Fail	Comments
U009-1	User updates a customer’s first name	• Customer: Steve Minecraft • Select what to update: First name • Enter new details: Terraria	Customer update submitted and saved in customer table in the system database	☑Pass	Validation for empty input
U009-2	User updates a customer’s last name	• Customer: Steve Minecraft • Select what to update: Last name • Enter new details: Terraria	Customer update submitted and saved in customer table in the system database	☑Pass	Validation for empty input
U009-3	User updates a customer’s email address	• Customer: Steve Minecraft • Select what to update: Email address • Enter new details: Terraria@test.com	Customer update submitted and saved in customer table in the system database	☑Pass	Validation for whether there is @ and a valid domain before and after @ and a “.”
U009-4	User updates a customer’s contact number	• Customer: Steve Minecraft • Select what to update: Contact No • Enter new details: 081 546 1672	Customer update submitted and saved in customer table in the system database	☑Pass	Validation rules exist for phone number format.
U009-5	User updates a customer’s physical address	• Customer: Steve Minecraft • Select what to update: Physical address • Enter new details: 321 Street Johannesburg 2109	Customer update submitted and saved in customer table in the system database	☑Pass	Validation for empty input

Input Field Test

Test ID	Field Name	Test Data	Expected Results	Pass/Fail	Comments
I0037	Customer ID	Taken From Combobox	Accurately load customers from the customer table	☑Pass	
I0038	Update Option	Taken from Combobox	Accurately load pre-determined options available.	☑Pass	Options available are the attributes of a customer minus ID

Note: No testing was done for the individual attributes relevant to the customer as the testing is the same as the customer create option. The validation rules are taken straight from the Create Customer page.

Create Manufacturer Page

Use Case Test

Test ID	Test Case	Test Data	Expected Results	Pass/Fail	Comments
U010	User creates a manufacturer	• Name: SolaraTech Solutions • Email address: sales@solaratechsolutions.com • Contact number: 011 536 6272 • Physical address: 123 Street Johannesburg 2109	New manufacturer submitted and saved in manufacturer table in the system database	☑Pass	

Input Field Test

Test ID	Field Name	Test Data	Expected Results	Pass/Fail	Comments
I0039	Name	"Test"	Accept Manufacturer name	☑Pass	
I0040	Name	""	Display error as input is left empty	☑Pass	

Note: Input tests for Email Address, Contact Number, and Physical Address are not carried out as the validation is the same as in the create customer page. The validation rules are taken straight from the Create Customer page.

Update Manufacturer Page

Use Case Test

Test ID	Test Case	Test Data	Expected Results	Pass/Fail	Comments
U011-1	User updates a manufacturer's name.	• Manufacturer: SolaraTech Solutions • Select what to update: name • Enter new details: SolaraTech	New manufacturer update submitted and saved in manufacturer table in the system database	☑Pass	Validation for empty input
U011-2	User updates a manufacturer's email address.	• Manufacturer: SolaraTech Solutions • Select what to update: email address • Enter new details: yesy@yes.com	New manufacturer update submitted and saved in manufacturer table in the system database	☑Pass	Validation for empty input Validation for whether there is @ and a valid domain before and after @ and a "."
U011-3	User updates a manufacturer's contact number.	• Manufacturer: SolaraTech Solutions • Select what to update: contact number • Enter new details: 011 647 8218	New manufacturer update submitted and saved in manufacturer table in the system database	☑Pass	Validation rules exist for phone number format.
U011-4	User updates a manufacturer's	• Manufacturer: SolaraTech Solutions	New manufacturer update submitted and saved in	☑Pass	Validation for empty input

	physical address.	• Select what to update: physical address • Enter new details: 123 Street Johannesburg 2109	manufacturer table in the system database		

Input Field Test

Test ID	Field Name	Test Data	Expected Results	Pass/Fail	Comments
I0041	Manufacturer ID	Taken From Combobox	Accurately load manufacturers from the manufacturer table	☑Pass	
I0042	Update Option	Taken from Combobox	Accurately load pre-determined options available.	☑Pass	Options available are the attributes of a manufacturer minus ID

Note: No testing was done for the individual attributes relevant to the manufacturer as the testing is the same as the customer create option. The validation rules are taken straight from the Create Customer page.

Create Staff Page

Use Case Test

Test ID	Test Case	Test Data	Expected Results	Pass/Fail	Comments
U012	User creates a staff member	• First name: John • Last name: Constantine • Email address: johnconstantine@gmail.com • Contact number: 081 526 4921 • StaffRole: Admin	Staff member submitted and saved to staff table in the system database	☑Pass	Validation for empty input Validation for whether there is @ and a valid domain before and after @ and a “.” Validation for phone number format

Input Field Test

Test ID	Field Name	Test Data	Expected Results	Pass/Fail	Comments
I0043	Staff Role	Taken From Combobox	Accurately load the 2 available options for a staff role.	☑Pass	

Note: Input tests for First Name, Last Name, Email Address, Contact Number, and Physical Address are not carried out as the validation is the same as in the create customer page. The validation rules are taken straight from the Create Customer page.

Update Staff Page

Use Case Test

Test ID	Test Case	Test Data	Expected Results	Pass/Fail	Comments
U013-1	User updates staff	• Staff: John Constantine	Staff member update submitted and saved	☑Pass	Validation for empty input

	member's first name.	• Select what to update: first name • Enter new details: Johnn Constantine	in the staff table in the system database		Validation for whether there is @ and a valid domain before and after @ and a "." Validation for phone number format
U013-2	User updates staff member's last name.	• Staff: Johnn Constantine • Select what to update: Last name • Enter new details: John Constantinee	Staff member update submitted and saved in the staff table in the system database	☑Pass	Validation for empty input Validation for whether there is @ and a valid domain before and after @ and a "." Validation for phone number format
U013-3	User updates staff member's email address.	• Staff: Johnn Constantinee • Select what to update: email address • Enter new details: yes@yes.com	Staff member update submitted and saved in the staff table in the system database	☑Pass	Validation for whether there is @ and a valid domain before and after @ and a "."
U013-4	User updates staff member's contact number.	• Staff: Johnn Constantinee • Select what to update: Contact number • Enter new details: 081 217 9189	Staff member update submitted and saved in the staff table in the system database	☑Pass	Validation rules exist for phone number format.
U013-5	User updates staff member's roles.	• Staff: Johnn Constantinee • Select what to update: staff role • Enter new details: Technician	Staff member update submitted and saved in the staff table in the system database	☑Pass	Validation for empty input

Input Field Test

Test ID	Field Name	Test Data	Expected Results	Pass/Fail	Comments
I0044	Staff ID	Taken From Combobox	Accurately load staff members from the staff table	☑Pass	
I0045	Update Option	Taken from Combobox	Accurately load pre-determined options available.	☑Pass	Options available are the attributes of a staff member minus ID

Note: No testing was done for the individual attributes relevant to the staff as the testing is the same as the customer create option. The validation rules are taken straight from the Create Customer page.

Create Unit Page

Use Case Test

Test ID	Test Case	Test Data	Expected Results	Pass/Fail	Comments
U014	User creates a job.	• Unit name: SolarMax Pro 3000 • Unit brand: EcoSun Technologies • Unit type: Solar panel Unit Size: 165cm	Unit submitted and saved to the unit table in the system database	☑Pass	Validation for empty input

Input Field Test

Test ID	Field Name	Test Data	Expected Results	Pass/Fail	Comments
I0046	Unit Name	“TestName”	Accept unit name	☑Pass	
I0047	Unit Name	“”	Display error as input is left empty	☑Pass	
I0048	Unit Brand	“TestName”	Accept unit brand	☑Pass	
I0049	Unit Brand	“”	Display error as input is left empty	☑Pass	
I0050	Unit Type	“TestName”	Accept unit type	☑Pass	
I0051	Unit Type	“”	Display error as input is left empty	☑Pass	
I0052	Unit Size	100cm * 100cm	Accept unit size	☑Pass	
I0053	Unit Size	“”	Display error as input is left empty	☑Pass	

Update Unit Page

Use Case Test

Test ID	Test Case	Test Data	Expected Results	Pass/Fail	Comments
U015	User updates a unit	• UnitID: 3 • Unit name: CANADIAN SOLAR 455W PANEL MONO 1 • Unit brand: Canadian Solar • Unit type: Solar Panel • Unit Size: height:78cm, width:39cm	Unit update submitted and saved to unit table in the system database	☑Pass	Validation for empty input

Input Field Test

Test ID	Field Name	Test Data	Expected Results	Pass/Fail	Comments

I0054	Unit ID	Taken From Combobox	Accurately load units from the unit table	☑Pass	

No testing was done for the individual attributes relevant to the unit as the testing is the same as the unit create option. The validation rules are taken straight from the Create Unit page.

Interface Design Testing

Participants

The test was performed by Yaaseen Hassim on 6 October 2024.

Methodology

Interface Design Testing involves testing several different aspects of the design of the interface. Overall, it focuses on evaluating the usability, consistency, and functionality of the system's user interface (UI). First, UI standards and guidelines are defined, including font chosen, size of font, colours, navigation design choice, etc. There are aligned with user experience best practices, and match Solarway Suppliers' branding design.

Firstly, the interface was tested for consistency, to ensure a cohesive user experience across the entire system. This includes tests of consistency with regards to font, colour, spacing, etc. Following this, the menu design and navigation system was tested, ensuring that every page can be easily accessed, and is labelled appropriately.

The results of these tests were then documented down, and whether it passed to an expected level, or failed and changes must be made to the interface design of the system.

Interface Testing Documentation

Colours

Test ID	Test questions	Pass/fail	Comments
IC01	Does the site use standard colours?	☑Pass	
IC02	Are the input field backgrounds the correct colour?	☑Pass	
IC03	Is the submit/update button consistent across pages?	☑Pass	
IC04	Is the main heading consistent?	☑Pass	
IC05	Are the subheadings consistent?	☑Pass	
IC05	Is the general screen background consistent?	☑Pass	

Text

Test ID	Test questions	Pass/fail	Comments
IT01	Are all the fonts consistent across the pages?	☑Pass	
IT02	Is all text properly aligned?	☑Pass	

IT03	Is the font consistent for the error messages	☑Pass	The error message is a system error for the specific browser, not chose by our development team.
IT04	Is the text readable across the different pages?	☑Pass	
IT05	Is there a clear distinction between different font types (e.g., bold for headings, regular for body text)?	☑Pass	
IT06	Are there no spelling errors?	☑Pass	

Images

Test ID	Test questions	Pass/fail	Comments
II01	Is the SolarWay Suppliers logo on the top left, large and prominent?	☑Pass	
II02	Is the Solarway Suppliers logo consistent?	☑Pass	
II03	Is the background image on all pages and consistent?	☑Pass	
II04	Is the Hexatech logo in the bottom left of every page and consistent	☒Fail	Logo is missing on several pages.

Navigation

Test ID	Test questions	Pass/fail	Comments
IN01	Is the navigation order logical and easy to navigate?	☑Pass	
IN02	Can all pages be accessed via buttons without issues?	☑Pass	Crash occurred once when navigating to update error but has not happened since, so pass.
IN03	Is the tab order on all pages, correct?	☑Pass	
IN04	Is there a clear and visible navigation menu on all pages?	☑Pass	
IN05	Are all buttons/inputs easily clickable and responsive?	☑Pass	

Report Testing

Participants

The test was performed by Yaaseen Hassim on 6 October 2024.

Methodology

Using the supplementary specification documentation through the analysis and design, the correct report structure was determined. The reporting functionality was then tested within the web application, checking to see if the report is using the correct tables from the correct database, and if it has an adequate filtering system.

Along these tests, testing was done to ensure that the report is designed in an intuitive and aesthetically pleasing manner. This includes tests of consistency with regards to font, colour, etc. The methodology for the report testing is simple as the reports are designed in PowerBI, a third-party app and not a direct web application-based design. Following these tests, the results are reported, and any additional comments are added where required.

Report Testing Documentation

Data for Reports

Test ID	Test questions	Pass/fail	Comments
RT01	Are the reports using the correct data from the correct table in the applicable database?	☑Pass	There is a delay in report updating, this is due to PowerBI in javascript restrictions.
RT02	Is there a data slicer to narrow down information, and does it work?	☑Pass	
RT03	Is there the option to export reports to email?	☑Pass	Option available due to PowerBI embed.

Design of Reports

Test ID	Test questions	Pass/fail	Comments
RT04	Is the font choice consistent across reports?	☑Pass	
RT05	Are percentages visible on bars in reports?	☑Pass	
RT06	Is the font size readable?	☑Pass	

Plan for Construction 2 Development

Following the completion of this testing phase, presented in this document, these results will directly inform and influence the development plan for the next, and final phase of this development project. The goal is using the findings from this testing plan to refine and further develop the system, in line with user functionality.

The first step is to analyse the results of the tests undertaken and assemble a list of the tests that were failed. With this list of failed tests, a plan will be drafted to adjust the system to resolve these issues and bugs. With this drafted plan, the development team will ensure that the bugs and issues are resolved. In addition to these functionality affecting bugs, improvements and enhancements will be made to areas within the system identified during testing that can be improved.

Code based bugs will be fixed in the javascript environment, and user interface bugs will be addressed in the html and css aspects of the system.

The final step is testing to ensure these bugs are resolved, this will be done using regression testing. Regression testing is a type of testing that ensures that new code changes or enhancements to the system do not negatively affect the rest of the system that is already established. Regression testing is the only new type of testing that will occur, in addition, a rerun of all failed tests will be undertaken, this is to ensure that the bug fixes are documented and verified. Any additional input tests that were not considered will be tested in the final testing report, such as password creation input rules.

Construction 2 – Testing Report

Purpose (Mission)

The second part of this testing report focusing on Construction 2 – in the aftermath of the initial testing report. It is undertaken after changes have been made to the system to fix and improve it with regards to the bugs and issues found following the completion of the initial testing phase.

The mission of this part is to ensure that the Solarway Suppliers Repair Hub system functions as intended and set out to as according to the analysis and design of the requirements following the changes made to the system to fix bugs found in the first round of testing. In addition, it is also to ensure that the bugs discovered in the first round of testing are eliminated.

Objectives and Tasks

Objectives

The objectives of this part of the test plan are as follows:

- Ensure the continued functioning of the system: The system should still be working in the expected manner even after the alterations made to address the bugs discovered in the first round of testing.
- Ensure bugs are eliminated: This test plan serves to document the elimination of any bugs discovered during the initial testing process by documenting the results following the alterations made to the system to eliminate said bugs.

Tasks

The tasks for this part of the test plan are as follows:

- Justification of choice of regression testing and retesting of certain test cases.
- Test planning for regression testing.
- Test case design for regression testing.
- Test execution and documentation.
- Final validation and sign-off.

Scope of the Testing

General – What is being tested?

- Regression testing – testing to ensure that functionality on forms with changed elements is still working as expected.
- Re-testing of previously failed tests.

What will not be tested?

Any previously passed tests on forms that are bug-free in the initial testing phase will not be repeated as they have already passed, and no changes have been made to these forms that are bug-free.

Justification of Testing Choices

Re-Testing of Failed Tests

Re-testing of failed tests from stage 1 of the testing report is critical to ensure that the identified issues and bugs have truly and fully been resolved and the system now functions as expected. During the initial round of testing undertake, a few tests failed due to bugs, or incorrect functionality, and without re-testing these previous failures, there is a risk that the system could still contain the same defects.

All in all, the main justification for undertaking the re-testing process is to verify issue resolution – confirmation that the bugs or issues identified in the previous round of testing have been properly addressed and that the system now behaves as expected. We also ensure that the issues do not recur which in turn ensures the stability and reliability of the system. Re-testing also ensures that the system complies with functional and business requirements that were initially unmet due to the failures.

Regression Testing

Regression testing is essential to ensure that recent changes to the system, with regards to resolving issues or bug fixes, have not unintentionally impacted the existing, already functional elements of the systems. It provides a safeguard against the introduction of new issues due to system fixes.

Whenever there are changes made to the system, there is a risk that these changes could inadvertently affect already existing other parts of the system. Regression testing ensures that the system remains stable even with the new changes considered.

Overall, regression testing ensure that the system integrity is maintained. Regression testing guarantees that the system's foundational functional requirements remain intact despite changes to the system. It provides confidence that the system is ready for deployment as it is fully functional after the modifications made following the first round of testing.

Actual Testing

Re-testing

Participants

This test was performed by Yaaseen Hassim on 24 October 2024.

Methodology

The re-testing process follows on from the initial round of testing undertaken and involves re-executing test cases that failed in the previous testing cycle to confirm that the issues have been resolved and that the system features function as expected.

The methodology goes as follows: Firstly, a list of all failed tests from the previous testing cycle are compiled. These failures are then further evaluated to understand the nature of the failure. This list is passed to the development team who then address these failures in the form of bug fixes and system enhancements. Following this, the same exact test cases are executed to validate that the issues have been fixed. The results are then documented down.

Re-Testing Documentation

Each failed test will be rerun here, with clear labelling to ensure convenience when comparing to the first round of testing.

Failed Input Field Tests

Create Customer Page

Test ID	Field Name	Test Data	Expected Results	Pass/Fail	Comments
I0023	First Name	“Test123”	Reject as name contains numbers.	☑Pass	
I0026	Last Name	“Test123”	Display error as input includes numbers.	☑Pass	
I0034	Contact Number	“1234”	Display error as contact number is invalid	☑Pass	

Additional Notes:

These validation rules for phone number format, and for peoples’ names not having numbers exist on every page where these inputs are required – they will not be individually tested as the validation rules are reused for each page, and as such, testing each input is redundant.

Failed Interface Design Tests

Note: The only failed interface design test was the missing Hexatech logo on each webpage, following feedback from stakeholders, this design element has been deemed redundant, and as such, will not be re-tested.

Regression Testing

Participants

This test was performed by Yaaseen Hassim on 24 October 2024.

Methodology

Regression testing is performed to ensure that the change to the system, such as bug fixes, do not affect the existing functionality. The goal is to confirm that previously functioning aspects of the system still work as expected.

Regarding the methodology of regression testing, it goes as follows: The areas that have been modified due to bug fixes are identified first, this involves identifying exactly which pages have had changes done to them. With this in mind, the functionalities of these webpages and their expected outcomes are noted down. A test is then carried out to ensure that this expected outcome is still reached, even with the modifications. The results of this process are then documented down, if the functionality still operates as intended, then the test passes.

Regression Testing Documentation

The below are tests to ensure that every webpage's, where changes were made, key functionality still works as expected.

Create Customer Page

Test ID	Test Case	Expected Results	Pass/Fail	Comments
R01	User creates a customer	Customer submitted and saved in customer table in the system database	☑Pass	Functionality still works, even with changes.

Update Customer Page

Test ID	Test Case	Expected Results	Pass/Fail	Comments
R02	User updates a customer (any attribute)	Customer record is updated with new information.	☑Pass	Functionality still works. The ability to change any attribute passes.

Create Manufacturer Page

Test ID	Test Case	Expected Results	Pass/Fail	Comments
R03	User creates a new manufacturer.	Manufacturer record submitted and saved in the manufacturer table in the system database.	☑Pass	Functionality still works, even with changes in validation rules.

Update Manufacturer Page

Test ID	Test Case	Expected Results	Pass/Fail	Comments
R04	User updates a manufacturer (any attribute)	Manufacturer record is updated with new information.	☑Pass	Functionality still works. The ability to change any attribute passes.

Create Staff Page

Test ID	Test Case	Expected Results	Pass/Fail	Comments
R05	User adds a new staff member to the system.	Staff member submitted and saved in staff table in the system database	☑Pass	Functionality still works, even with changes.

Update Staff

Test ID	Test Case	Expected Results	Pass/Fail	Comments

R06	User updates a staff member's attributes.	Staff record is updated with new information.	☑Pass	Functionality still works. The ability to change any attribute passes.

Analysis of Results and Roundup

The purpose of this second part of the testing report was to ensure that the bugs and issues discovered in the initial round of testing were eliminated, and that the system continues to function to the expected level. Following completion of this phase of testing, including the re-testing and regression testing processes, the results indicate that all failed tests have now passed successfully, and that the system's functionality is not comprised. The system is ready for final deployment.

Firstly, the re-testing process undertaken, that specifically focused on the previously failed test cases, to verify that bug fixes took place shows that all tests that previously failed have no passed and confirms that the applied bug fixes and adjustments have resolved the underlying issues. Each failed test case was re-executed in the same manner, and now performs as expected and required.

Additionally, the regression testing results show that the recent bug fixes and changes to the system have not compromised the existing functionality of the system – that is, the core functions of the webpages with modifications still work as intended.

The successful outcome of both re-testing and regression testing demonstrate that the system is now in a much-improved state. The fixes applied have addressed the underlying bugs found in the previous testing phase while also maintaining the integrity of the system. We can confidently say that the system is ready for final deployment for Solarway Suppliers.