Test Report

Test Report Section 1

In this phase of the project we are concerned with testing the final product; the system that we have produced for our client. We are using all types of tests (unit, user acceptance, system, and integration testing) to make sure that the system fits the requirements detailed in the requirements specification, and just functions as a whole.

The main items that are going to be tested via unit testing are:

- Database Select **(Functional) –** Does the database return the correct data on executing a SELECT query?

- Database Delete **(Functional)** – Does the database delete the required record(s) on execution of a DELETE statement?

- Administrator's Page interface with Database **(Functional)** – Do the three above functions work from the administrator's page? This will be tested as a unit; using a stub instead of the actual database.

- Email **(Functional)** – Does the target email address receive the correct data via email when calling the method with requisite data from the command line? (for both Volunteer and Help emails)

The items that are going to be tested via system testing are:

- Blog Page Creating/Updating/Deleting **(Functional)** – Can the client create a new blog post, update the created blog post, and then delete it all from the administrator's page?

- Testimonials Creating/Updating/Deleting **(Functional)** - Can we create a new testimonial, update the created testimonial, and then delete it all using the administrator's page?

- Carousel Adding/Removing Images **(Functional)** – Can we add and remove images from the carousel using the administrator's page?

- Carousel Adding/Removing Videos **(Functional)** – Can we add and remove video from the carousel using the administrator's page?

The created system does not involve a large amount of integration but there are definitely some that needs to be achieved. Specifically, the programs/units that need to be integrated are the database with the back-end server, the back-end server with the administrator's page, and the administrator's page with the main web pages. To test the integration of these large components tests will be performed on the functions that require two (or more) components to work together, ensuring that the expected results are returned. Furthermore, the communication between these components will be monitored to ensure correctness for a greater number of situations. This will provide a greater confidence level that the results gleaned are not lone occurrences. An example might be testing that the administrator's page can send off a request to the database, via the back-end server, to SELECT some data; while the communication of each component is monitored to ensure the correctness of data that is to be returned.

Test Report Section 2

Test Case ID	unit_test-01
Description of test	Test that the back-end sends responses to SQL queries on the database in the correct format.
Related requirement specification details	AD3
Pre-requisites for test	The back-end server and the database are running, with the server awaiting requests.
Test procedure	1. Send a request to the server to SELECT some data from the database 2. Receive the data on the command line. 3. Check that the data is in JSON or similar format and each field in the database is a key with the value being the record's value for this key.
Test material used	None
Expected result	The request is sent to the server which responds with the data that has been asked for within 3 seconds. This data should be in the format described above.
Comments	None
Created by	JM
Test environment(s)	CMD, Terminal

Test Case ID	unit_test-02
Description of test	Test that when a user tries to send an email from either of the forms on the website, this is received correctly by the account connected to the server.
Related requirement specification details	U1
Pre-requisites for test	The back-end server should be able to accept data from the form and email this to the email which the server is authorised to send to.
Test procedure	1. Navigate to the volunteer page. 2. Fill in the details of the form on the appropriate tab. 3. Press the submit button. 4. Navigate to the email address which is set up to receive the emails. 5. Check that the email has been received. 6. Check that the email has the correct data.
Test material used	None
Expected result	The email should be sent to the correct email address and should contain the correct data.
Comments	After Step 3, an alert needs to notify the user on the success/failure of their submission. This should be done within a small amount of time.
Created by	JM
Test environment(s)	Google Chrome, Firefox, Safari

Test Case ID	**unit_test-03**
Description of test	Test the message prompt for high-risk actions such as deleting records from the database or changing the content of a page.
Related requirement specification details	AW8
Pre-requisites for test	The server is running and there are no errors while loading the pages or performing actions.
Test procedure	1. Navigate to the *About Us* tab in the Admin Page. 2. Change the content of the editor and Click Submit 3. If an alert appears click ok and check if the content is updated or not. 4. Repeat process but if an alert appears click cancel and check if the content is updated or not. 5. Repeat for other editors as well as the *Delete* function of the database tab.
Test material used	Web Browser
Expected result	A message prompt should appear. If Ok is selected, the action should continue as normal and if *Cancel* is selected then the action should not complete.
Comments	Check the alerts' information that they display about the actions. Moreover, check for correct use of grammar and spelling.
Created by	CCM
Test environment(s)	Google Chrome, Firefox, Internet Explorer, Microsoft Edge

Test Case ID	unit_test-04
Description of test	Test the addition of a new record in any table of the database.
Related requirement specification details	AD1
Pre-requisites for test	The server is running alongside the database which should already be setup. The connection between the two should be tested without any problems.
Test procedure	1. Navigate to the *Database* tab in the admin page. 2. Select a table from the drop-down and select *Insert* from the other. 3. Fill out the fields with correct data. 4. Check if the database contains the newly added data.
Test material used	Web Browser, Database
Expected result	The test data inserted in the fields of the Admin Page, will appear in the database.
Comments	Step 4 of the procedure can either be achieved by checking the table or running an SQL command for the table to be shown: "SELECT * FROM *table.name"*
Created by	CCM
Test environment(s)	Google Chrome, Firefox, Safari, Microsoft Edge, MySQL

Test Case ID	unit_test-05
Description of test	Test the removal of a record from any table of the database.
Related requirement specification details	AD2
Pre-requisites for test	The server is running alongside the database which should already be setup. The connection between the two should be tested without any problems.
Test procedure	1. Navigate to the *Database* tab in the admin page. 2. Select a table from the drop-down and select *Delete* from the other. 3. Choose the correct column and fill out the fields pointing at a piece of data to be deleted. Choose the correct operators if the second field is also used. 4. Check if the database still contains said data.
Test material used	Web Browser, Database
Expected result	The data that was selected to be deleted must not show up in the database
Comments	Step 4 of the procedure can either be achieved by checking the table or running an SQL command for the table to be shown: "SELECT * FROM *table.name"*
Created by	CCM
Test environment(s)	Google Chrome, Firefox, Safari, Microsoft Edge, MySQL

Test Case ID	**unit_test-06**
Description of test	Test the display of selected data from the database.
Related requirement specification details	AD3
Pre-requisites for test	The server is running alongside the database which should already be setup. The connection between the two should be tested without any problems.
Test procedure	1. Navigate to the *Database* tab in the admin page. 2. Select a table from the drop-down and select *Select* from the other. 3. Choose the correct column and fill out the fields pointing at a piece of data to be selected/displayed. Choose the correct operators if the second field is also used. 4. Test on MySQL Workbench if a select script with the same variables, (columns,operators,equalities etc.) displays the same results as the webpage.
Test material used	Web Browser, Database
Expected result	The result displayed in the webpage must be the same with the result in MySQL Workbench.
Comments	Step 4 of the procedure can either be achieved by checking the table or running an SQL command for the table to be shown: "SELECT * FROM *table.name"*
Created by	CCM
Test environment(s)	Google Chrome, Firefox, Safari, Microsoft Edge, MySQL Workbench

Test Case ID	sys_test-01
Description of test	Test that the administrator can create, modify, and delete new blog posts.
Related requirement specification details	AW1
Pre-requisites for test	The website has a connection to the back-end via the administrator's page and the administrator page's blog and testimonial section can be accessed.
Test procedure	1. Create a new blog post on the admin page 2. Check the blog post appears on the main website 3. Modify this blog post on the admin page 4. Check that the modification alters the main website accordingly 5. Delete the blog post on the admin page 6. Check that the blog post is no longer on the main website
Test material used	Web Browser
Expected result	The blog post is created, modified, and then deleted with each action being displayed on the main website.
Comments	When deleting the user must confirm the action in the pop-up alert.
Created by	JM
Test environment(s)	Google Chrome, Firefox, Safari

Test Case ID	**sys_test-02**
Description of test	Test that the administrator can create, modify, and delete new testimonials.
Related requirement specification details	AW1
Pre-requisites for test	The website has a connection to the back-end via the administrator's page and the administrator page's blog and testimonial section can be accessed.
Test procedure	1. Create a new testimonial post on the admin page 2. Check the testimonial appears on the main website 3. Modify this testimonial on the admin page 4. Check that the modification alters the main website accordingly 5. Delete the testimonial on the admin page 6. Check that the testimonial is no longer on the main website
Test material used	Web Browser
Expected result	The testimonial is created, modified, and then deleted with each action being displayed on the main website.
Comments	When deleting the user must confirm the action in a pop-up alert.
Created by	JM
Test environment(s)	Google Chrome, Firefox, Safari

Test Case ID	sys_test-02
Description of test	Test that the administrator can create, modify, and delete new events.
Related requirement specification details	AW1, AW2
Pre-requisites for test	The website has a connection to the back-end via the administrator's page and the administrator page's events section can be accessed.
Test procedure	1. Create a new event post on the admin page 2. Check the event appears on the main website 3. Modify this event on the admin page 4. Check that the modification alters the main website accordingly 5. Delete the event on the admin page 6. Check that the event is no longer on the main website 7. Check that the deleted event is in the events archive.
Test material used	Web Browser
Expected result	The event is created, modified, and then deleted with each action being displayed on the main website.
Comments	When deleting the user must confirm the action in a pop-up alert.
Created by	CMM
Test environment(s)	Google Chrome, Firefox, Safari

Test Case ID	sys_test-03
Description of test	Test that the administrator can create, modify, and delete new testimonials.
Related requirement specification details	AW1
Pre-requisites for test	The website has a connection to the back-end via the administrator’s page and the administrator page’s blog and testimonial section can be accessed.
Test procedure	7. Create a new testimonial post on the admin page 8. Check the testimonial appears on the main website 9. Modify this testimonial on the admin page 10. Check that the modification alters the main website accordingly 11. Delete the testimonial on the admin page 12. Check that the testimonial is no longer on the main website
Test material used	Web Browser
Expected result	The testimonial is created, modified, and then deleted with each action being displayed on the main website.
Comments	When deleting the user must confirm the action in a pop-up alert.
Created by	JM
Test environment(s)	Google Chrome, Firefox, Safari

Test Case ID	sys_test-04
Description of test	Test that the administrator can upload and delete images and videos from the carousel.
Related requirement specification details	AC1, AC2
Pre-requisites for test	The website has a connection to the back-end via the administrator's page and the administrator page's home section can be accessed.
Test procedure	1. Upload new image to carousel. 2. Check that the new image is uploaded and shown in the public home page (in the correct position). 3. Delete the image/video. 4. Check that the image/video no longer appears on the website.
Test material used	Web Browser
Expected result	The image is uploaded and deleted with each action being displayed on the main website.
Comments	When deleting the user must confirm the action in a pop-up alert.
Created by	CCM
Test environment(s)	Google Chrome, Firefox, Safari

Test Case ID	uat_test-01
Description of test	Test that the user can navigate around the main website.
Related requirement specification details	U1-4
Pre-requisites for test	The website must be fully functional and have undergone all sys_test cases prior to this.
Test procedure	1. User navigates around the main website 2. User can find all of the features necessary with little to no hassle.
Test material used	Web Browser
Expected result	The user should be able to carry out the above tasks within a few minutes.
Comments	None
Created by	JM
Test environment(s)	Google Chrome, Firefox, Safari

Test Case ID	uat_test-02
Description of test	Test that the user can make use of the administrator’s webpage
Related requirement specification details	AW1-8
Pre-requisites for test	The website must be fully functional and have undergone all sys_test cases prior to this.
Test procedure	1. User types in the URL for the admin page and logs in 2. User changes password for the administrator’s page 3. User makes a change to the main website on the admin page 4. User logs out of the admin page
Test material used	Web Browser
Expected result	The user should be able to carry out the above tasks within 8-10 minutes going at a moderate pace.
Comments	The choice of what to change on the main website is left to the user to decide when they reach this section, because all of the routes through the system should be fully functional.
Created by	JM
Test environment(s)	Google Chrome, Firefox, Safari

Test Case ID	uat_test-03
Description of test	Test that the loading and reloading times of each webpage is within a reasonable amount of time. Furthermore, the navigation to each webpage through the navbar will be tested.
Related requirement specification details	NFR-Performance, FR-U2
Pre-requisites for test	The server is running and there are no errors while loading the pages or performing actions.
Test procedure	1. Access each page by both navigating to it via the navbar, and changing the URL. 2. Refresh each page after accessing it. 3. Perform actions on each page. 4. Measure the time taken for each of these actions to complete.
Test material used	Web Browser
Expected result	The webpages complete these actions (load, reload and other actions) in under 2 seconds, preferably in under .5 seconds.
Comments	Actions will test both the Javascript elements of the pages as well as the back-end javascript. To measure the time taken, browser Developer Tools will be used
Created by	CCM
Test environment(s)	Google Chrome, Firefox, Safari, Microsoft Edge

Test Case ID	uat_test-04
Description of test	Test that a user can fill out the form for emailing our client in a short amount of time; suggesting that the form is intuitive to fill out.
Related requirement specification details	U1
Pre-requisites for test	The pages which include the forms for contacting our client should be functional, and the email system should also be functional.
Test procedure	1. Load up the web page containing the form for emailing. 2. User should fill in the form with appropriate data. 3. The user should be able to do this in an appropriate amount of time. 4. The correct data should be received by the test email.
Test material used	None
Expected result	The email should be sent to the correct email address and should contain the correct data. The form should be filled out in a small amount of time.
Comments	None
Created by	JM
Test environment(s)	Google Chrome, Firefox, Safari

Test Case ID	uat_test-05
Description of test	Test that the donation button works and the money would be sent to the correct account via VirginMoneyGiving.
Related requirement specification details	U7
Pre-requisites for test	The donation button should be included on the Volunteer page under more information.
Test procedure	1. Load up the volunteer page. 2. Navigate to the more information section of the website. 3. Click on the donation button on the website.
Test material used	None
Expected result	The VirginMoneyGiving account is correct when the button is clicked.
Comments	None
Created by	JM
Test environment(s)	Google Chrome, Firefox, Safari

Test Case ID	int_test-01
Description of test	Test that the AJAX requests from the front-end for the blog and testimonials page get the data it requires from the back-end server and in the correct format.
Related requirement specification details	AD3, AW2
Pre-requisites for test	The back-end server should be able to get the correct posts and testimonials from the database and return these values. The front-end JavaScript should have been tested with some dummy data which has the correct format. These can then be linked via AJAX.
Test procedure	1. Place some data in the blog posts table 2. Force the AJAX request by calling the Javascript function on the front-end 3. Print out the response 4. Check that the data returned is in the correct format.
Test material used	None
Expected result	The data that is returned from the back-end server is in the correct format and contains the data placed into the database.
Comments	None
Created by	JM
Test environment(s)	CMD, Terminal

Test Case ID	int_test-02
Description of test	Test that the carousel can be edited from the admin page. New videos and photos should be able to be added and removed.
Related requirement specification details	AC1-AC3
Pre-requisites for test	The back-end server should be able to upload and delete the files appropriately and these should be served to the front-end when the home page is loaded.
Test procedure	1. Access the admin page and log in with correct credentials. 2. Open new tab and navigate to the home page of the website to check the carousel photos currently on display. 3. Go back to admin tab and add a new photo; removing one of the old ones. 4. Go back to the home page tab and refresh. Check that the carousel has been changed appropriately. 5. Repeat this process for a video. 6. Check that the file structure on the server is correct; files have been removed/added correctly.
Test material used	Test images for the carousel. Included is a video.
Expected result	Regardless of the changes made to the pictures on the carousel, when the home page is loaded the images/videos should be displayed correctly and with minimal loading time. File structure should be correct.
Comments	None
Created by	JM
Test environment(s)	Firefox

Test Report Section 3

Our tests will be performed in a variety of environments. As it is primarily a website, we have aimed to perform our tests over a few of web browsers, namely Google Chrome, Microsoft Edge, Mozilla Firefox, Opera and Safari. We have chosen these browsers, on both a Windows 10 and Ubuntu Linux Operating System as this will cover the most likely browsers that a User or Admin will use to access the site. Safari was specifically chosen to accommodate for our clients personal experience as we know that they mainly operate through a MacBook and Safari browser. These tests will help us ensure that the system built has the longevity that our client desires, as well as the ease of access and better user experience.

Some of our systems testing will also be using Command Line Interfaces such as Windows Command Prompt and Linux's Terminal Window. These will allow both simple access to the system in order to perform the tests but will also be testing the path a future web-developer might take in order to update/fix the system, should that need be.

We have a series of classifications ready for any faults that the tests may produce, ordered in ascending severity. These will help us catalogue which faults are in desperate need of fixing and how they affect the user experience.

No.	Level of Severity	Classification
1	Mild	Data is correct and everything is processed, with a few minor unexpected alterations (e.g. spelling errors, buttons out of line) that change nothing functionally.
2	Annoying	Date is presented in way that increases the difficulty of using the system but does not change anything functionally.
3	Serious	An important piece of data for either the system or the user is either lost or processed incorrectly.
4	Extreme	An active series of disruptions in the running of the system or the viewing of the web-pages.
5	Intolerable	Data that is outside the bounds of what should be processed being deleted or altered.
6	Catastrophic	A full shutdown of the server or database. Also, if parts of the Admin Page and confidential information is accessed by unauthorised users.