Test Summary Report

**Contents**

1. Purpose 2
2. Application Overview 2
3. Testing Scope 2
4. Metrics 3
5. Types of Testing Performed 4
6. Test Environment and Tools 5
7. Lessons Learnt 5
8. Recommendations 5
9. Best Practices 6
10. Exit Criteria 6
11. Conclusion 6
12. Purpose

The purpose of this document is to explain and show how we will be testing certain attributes and actions of our program/website. We also intend to show that with these tests we will be performing our website will be bulletproof and without any failures of our test to improve the longevity of our system.

1. Application Overview

Our program is intended to be used by the owners and the members of the 40 and 8 club this is primarily used to keep all the information of the organization in one place to make things super simple and streamlined. Members can come to our site to learn more about the organization, to check for upcoming events, be able to apply for membership, and to be able to connect with the organization through social media. Admins will be able to approve people for membership status and to search through a database of all members for information related to contacting them.

1. Testing Scope
2. In Scope(Functional Testing)

* Adding new members to the database accordingly and deleting information of people that were not approved for membership.
* Making sure events are displayed when form has been submitted by admin and displayed appropriately in the calendar.
* Testing that the appropriate members are displayed when searched by admins.
* We will also need to test and make sure the database is connected to the system at all times and if it is not then the website needs to be unserved from the web service.

1. Out of Scope

* No Performance testing

1. Items not Tested

* We will not be testing that the website is making its connection to the web service because as soon as it goes down so will the website.

1. Metrics

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Cases Planned** | **Test cases executed** | **Test Cases Passed** | **Test Cases Failed** |
| 15 | 13 | 10 | 3 |

1. Test Cases Planned vs. Executed:
2. Test Cases Passed vs Failed:
3. Types of Testing Performed
   1. Modular Unit Testing

Angular CLI affords us the ability to create new modules for our website easily with the command ‘ng generate’ which also creates template unit tests, called specs, for each component in a separate .spec.ts file. We can design our unit tests for each component in these files, and run them all with the ‘ng test’ command.

* 1. End-to-end Testing

Similarly, we can run end-to-end integration tests with the command ‘ng e2e’.

1. Test Environment and Tools
   1. Karma v5.0.9

When running unit tests in Angular CLI, a locally-hosted version of the app is built and served, and unit tests are run against this server. A component in Angular CLI called Karma opens a new controlled Chrome window which runs and details the results of the unit tests, as well as shows a version of each component or web-page currently being tested against.

* 1. Protractor v5.4.3

When running end-to-end tests, a testing framework for Angular apps called Protractor is used with the ‘ng e2e’ command.

1. Lessons Learnt

|  |  |  |
| --- | --- | --- |
| **S. No** | **Issues Faced** | **Solutions** |
| 1 | There was no original file with members information anywhere. | Manually Insert the data. Also provide a way to output that data into a file if needed. |
| 2 | Long wait for hear back on what we are doing with the domain | Put the site up on a temporary testing area where we can show them. |

8. Recommendations

* If any site issues arise, one of us should be able to go on and fix those issues for them in a timely fashion. We can set it up to allow this.
* Provide an easy way for someone to go in and view the tests are passing. It is viewed from a URL but, we may add a button to go to that page.

9. Best Practices

* Contacting our sponsors on the regular regarding our changes was beneficial. Allowed us to get changes geared towards their liking quickly.
* Using Jasmine and Karma for testing was a very good idea. These types of tests execute automatically once the Application is built and deployed.
* Allowing for output of the user data is beneficial for us, them and anyone else who ever has to work on this.

10. Exit Criteria

* All test cases should be executed – **Yes.**
* All defects in Critical, Major, Medium severity should be verified and closed – **Yes.**
* Any open defects in trivial severity – **Action plan should be prepared with an expected date of closure.**

11. Conclusion

Once the Application meets the criteria as mentioned in Section 10 above, it is suggested to ‘Go Live’ by the Testing Team (Which is us, the developers). Appropriate user testing should be done. Allow users to go on and use the site for a period of time, they will report back if there are any issues. These issues will then be addressed and will set us back to the ‘Go Live’ state.

Submitted by Drew Yellina, Jakob Markland, and Bryce Winnecke