Test Plan Document

For

Forty and Eight Bar

1 Introduction 3

2 Business Background 3

3 Test Objectives 3

4 Scope 3

5 Test types Identified 3

6 Problems Perceived 3

7 Architecture 3

8 Environment 3

9 Assumptions 3

10 Functionality 3

11 Security 4

12 Performance 4

13 Usability 5

14 Test Team Organization 6

15 Schedule 6

16 Defects Classification Mechanism 6

17 Configuration Management 6

18 Release Criteria 6

Test Plan

# Introduction

This documents serves the purpose to show our test cases and what we intend on testing within our program. We will go through front end and back end test cases to make sure that the website/program functions as it should to ensure there will be little to no maintenance to maintain this website. This website serves the purpose to the 40 &8 to inform its members and to also expand the community of the 40&8. The tests we will cover in this document will be related to maintaining this website to stay online and also testing the user experience will using the site. The tests will also make sure navigating the site and entering data into the site will be smooth and will allow the user to have a good experience.

# Business Background

The background of the 40&8 was that this was started as a veteran club but they did not want the club to just be vets they wanted to allow the vets to be able to invite their friends and family. So they started with just vets they invited their friends and family to become members and from their friends and family they were also able to invite other into the club. So this club is not just a veteran bar/club this is an organization that supports veterans and its community. They have a nursing scholarship and they do different charitable activities in the community.

# Test Objectives

1. Will the site work on mobile phones and tablets
2. If the url path the user uses is wrong what screen will come up?
3. The form they will fill out to try and become a member each textbox needs to have parameters to make sure those textboxes are all formatted the same way
4. If the mail server is unable to send the email what will happen to that form?
5. When an admin approves a user we need to check that that new member was added to the database and also if they were declined we need to check and make sure they were not added to the database
6. Along with that we need to make sure when they submit the form the form is being sent in an email to the admin and it is also available for the admin to see when they are logged in
7. We need to test and make sure that the admin site is locked only to the admins login and that no one else can make it in to that part of the application
8. We also need to check and make the sure the databse is always live and it never gets disconnected from the website because if someone applies on the site to become a member and the database is offline then we have lost that application
9. We will also make it available that every 7 days the mail server will email the admin saying to update the schedule and we will have a form for them to fill out so that that can be posted under the calendar part of the website so we will need to test that that email comes out every 7 days and that the form is available every 7 days.
10. We will also have to test whether the facebook links we put in there if they are valid and what happens if they click on the link and in the rarest instant facebook is offline what will happen.

# Scope

***Inclusions:*** Functionality tests, API tests, Database testing

***Exclusions:*** hardware testing, System test, Agile Testing

# Test types Identified

The testing types we identified as necessary are Functionality tests, API tests, and Database tests. Functional testing and database testing are one in the same so we will perform functional testing to make sure that the database is working how it should. API testing will test that the API’s that we are using are giving us the output that we are looking for. The database testing is a little more complicated we will have to find documentation for that but I am sure it will be fairly simple. For the database we just need to check for new entries and to make sure those entries are accurate from what the user submitted.

# Problems Perceived

I think one of the bigger problems that comes to mind is testing the mail server as a whole. One aspect of that is testing to make sure the form was submitted and the email was sent another aspect of that is making sure that what is in the form is accurate information and not the applicant before that’s information. Another aspect of the mail server concept is when the admin accepts or denies the application the system will either add them to the database or their information will be erased.

# Architecture

# Environment

# Assumptions

# Functionality

***Constraints and Resolutions***

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Customer Constraints** | **Infosys Limitations** |
| Constraint 1 |  |  |
| Constraint 2 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

***Risk Identified & Mitigation Planned***

***Test Strategy***

***Automation Plans***

***Deliverables***

# Security

***Constraints and Resolutions***

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Customer Constraints** | **Infosys Limitations** |
| Constraint 1 |  |  |
| Constraint 2 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

***Risk Identified & Mitigation Planned***

***Test Strategy***

***Automation Plans***

***Deliverables***

# Performance

***Constraints and Resolutions***

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Customer Constraints** | **Infosys Limitations** |
| Constraint 1 |  |  |
| Constraint 2 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

***Risk Identified & Mitigation Planned***

***Test Strategy***

***Automation Plans***

***Deliverables***

# Usability

Our plan to go about testing usability, is to allow the rest of the class to use our site. The site will be hosted on a cloud server, being able to be accessed by everyone that knows the URL. Having other people other than us testing this will give us good feedback as to how easy it is to navigate. The customer wants the site to be very simple, considering who this is being built for. There is no plan to automate this type of testing, as this type of testing is difficult to automate.

# Test Team Organization

Other groups with the P446 course.

# Schedule

**Perform Execution of Tests** - <5 mins

**Test Reporting** - <5 mins

# Defects Classification Mechanism

* **Functional Defects** – Experiencing no functional defects currently. The UI responds as it should, data is returning with the correct values.
* **Performance Defects** – Experiencing no performance defects currently. Everything is running and executing at the speed it should. Tests are not taking too long to run.
* **Usability Defects** - We experience a cosmetic defect with usability, since older people will now have to use a newer system to show interest in the club. This will not be a major issue, just may take a very short amount of time for them to figure out how it works. If the UI would need to be structured any different way, turnaround time should be around 1 to 2 days. These are minor things that can be fixed easily.
* **Compatibility Defects** – Currently, we are experiencing some minor compatibility defects. These defects are browser related, as the UI is displayed different depending on your chosen browser. This can be fixed with simple polyfills, turnaround time will be 1 to 2 days to fix this defect.
* **Security Defects** – Experience no security defects currently.

# Configuration Management

Our tests are being configured using Karma and Jasmine. These are what the angular CLI uses to test the UI. Karma is running the tests and Jasmine is the test reporter which is generated through HTML.

# Release Criteria

Tests should execute successfully and test all important features of the application. Also, ensure compatibility amongst all environments.