Badger-Tech Timeline Test Plan

Table of Contents

1. Introduction

1.1 Purpose

A local museum located on Niagara-On-The-Lake has contacted Brock University, St. Catherines with the task of designing, developing, and implementing an online interactive timeline application. This application will depict the events surrounding the history of Niagara, with special attention given towards the war of 1812.

This document is designed to reference the testing plan, approach, and overall framework that will be used to address the testing process of https://badger-timeline.infinityfreeapp.com/src/ - site. Here, this document will introduce:

- Objectives
- Test Strategies
- Requirements
- Features
- Schedule
- Risks/assumptions
- Tools

Furthermore, this document will introduce the necessary steps taken in the following testing phases:

1. Unit testing
2. Component testing
3. Integration testing
4. System testing (Acceptance testing)

1.2 Project Overview

The https://badger-timeline.infinityfreeapp.com/src/ -site is a powerful web-application that will be responsible for facilitating the process of information passage from the client(Niagara-on-the-lake Museum) to the user(General public).

On the user side, this website will allow users to view information about the museum, the date of any events or workshops sponsored by the museum, create an account for email notifications, and will also have an interactive timeline application that will give the user an experience through Niagara's rich history. This web-app can be accessed and experienced through any device and in any orientation.

On the client side, this website will allow permitted staff members(admins) to edit content directly on the timeline such as texts and images. In addition, admins will also be allowed to edit content on the web such as its services, as well as adding any more additional information about the museum.

2. Objective and Tasks

2.1 Objectives

The objective of this test is to verify that the web-application - https://badger-timeline.infinityfreeapp.com/src/ is working as intended. To verify, all major components of the application must work as intended. The web-application can be considered and thought of as two major components: the webpage component and the timeline component. Both of these components will each be tested separately to check that they are working according to specifications, before finally testing the system as a whole.

The objective of testing the web-page component is to check and verify the various features offered in the homepage. These tests can include and are not limited to checking all links found on the homepage, checking that each button found works as intended, to verify the login/register system, and also to check that the webpages can be readily accessible by various devices such as PCs, Notebooks, and mobile devices. All the while having a reasonable amount of expectations towards the UI/UX associated with each device.

The object of testing the timeline component is to check and verify that the various features implemented in the timeline are all working in accordance to its specifications. These tests can include but are not limited to checking each individual point on the timeline spectrum, checking the various ways to navigate the timeline, checking how the information is presented, editing features for admin users, the sub-timeline itself, image view function, and etc. The timeline application must also be able to operate on various devices such as PCs, Notebooks, and mobile devices, and have a reasonable amount of expectations towards the UI/UX associated with each device.

The final product of this test plan will yield both a production-ready web-application, and also a set of test-cases and scenarios that can be reused whenever a new feature is to be added or changed in the future and/or for maintenance purposes.

2.2 Tasks

This section highlights the various tasks presented in this test plan document. The Tasks are separated and listed by its responsibility as a whole towards the testing plan. These tasks can be subject to change, but can include and not limited to:

- Testing phase
- Post-testing
- Problem reporting
- Retesting
- Acceptance Testing
- Regression Testing

3. Scope

3.1 General

This section will describe which components of the system will be tested, basic and/or added functions that are specific to a component, interface of each component, how the component will interact once integrated with one another, and the expectations of the integrated system.

There are three components that make up the overall web-application system. These components consist of the database, web-pages, and the timeline application.

The database is wholly responsible for storing and retrieving information, and is used by both the web-page component and the timeline component. Once integrated with the web-page component, the database will be responsible for providing information about any upcoming events and workshops hosted by the museum, along with storing and authenticating information for requests involving login/registration. Furthermore, the database is responsible for storing roles that are associated with each user. Once integrated with the timeline component, the database will have the responsibility of providing stored information for each historical time-period and event for the timeline to display. In addition, the database will need to support the editing feature by having the ability to delete old entries and store new entries. All interactions involving the database will be done through either the web-page component or timeline application component. The user cannot directly access the database itself, it will remain independent.

The web-page component will be responsible for greeting the user and will act as the main hub of interaction for information. The web-page will not contain any information about the history of Niagara, but will instead contain information and content about the client(Niagara-On-The-Lake Museum). Content and features presented on the web-page component include information about services offered by the museum, information about the museum itself, up-coming events/workshops that are hosted by the museum, and a login/register feature. Once integrated with the database, the web-page component will retrieve login information, as well as store registration information from the database. The web-page will then get the role of the user, and display the correct actions associated with the role of the user. Once integrated with the timeline component, the web-page will be responsible for providing a user a path to the timeline application either through a link/button, or through the search feature.

The timeline application component is responsible for giving the user a rich walkthrough experience on the events and historical periods surrounding the Niagara region. Unlike the web-page component, this timeline component will be solely responsible for displaying any and all information regarding historical periods and any events surrounding those periods to the user. Once integrated with the database, the timeline component will retrieve the necessary information regarding history periods and events and display the information for

the user to view. Any/all interactions involving the database will be done through the timeline component, such as when an admin wants to make an edit request for a historical event. Once integrated with the web-page application, the timeline component will have very minimal interaction with the web-page, since the timeline component will not need to rely on the web-page for anything. The two components will mostly be responsible for providing a link to each other.

3.2 Tactics

This section will describe how the testing process of the scope will be carried out in regards to the exploratory, functional, and acceptance level of testing.

Exploratory level will be responsible for ensuring that unit level defects are removed before the next level of testing can start. This level of testing will be carried out in the web-application with the use of test scripts. This testing can be done by analyzing the software code and determining whether or not the unit will work as intended. Some examples include basic navigation and modules.

Functional level will be responsible for testing the function of each feature. This level of testing will be carried out in the web-application with the use of test-scripts. This testing can be done by feeding in an input for a given feature and then validating the output of the feature. Some examples include seeing whether the database has saved an entry, whether the user is able to log in using the correct credentials, and etc.

Acceptance level will be responsible for testing the overall functionality of the system. This level of testing will be carried out in the web-application without the use of test-scripts, as the goal is to test from a user point of view. This testing can be done by taking a user story and attempting to validate the user story.

4. Testing Strategy

4.2 System and Integration Testing

First, we will do *Integration testing* which tests independent components and models of our system as a group. The testing is done after unit testing and before system testing. We will make use of integration testing for our Login page. When a user enters their username and password and clicks login this will direct them towards the home page for the niagara museum website. This is an example of integration testing because we test the overall login page functionality, not each individual component such as username and password. Therefore, combinations will need to be checked such as:

Invalid username invalid password (correct error message)
Invalid username valid password (correct error message)
Valid username invalid password (correct error message)
Valid username valid password (no error message and directs user to home page)

Then, we will perform *system testing* which tests the system as a whole. The Niagara museum website with all the functionalities/components of the application interact together

5. Requirements.

6. Test Schedules

- April 5 - 18

7. Control Procedures

8. Features to be tested

Here, the database will be tested for its ability to store and retrieve system generated information such as number of visitors, cookies, and etc. Furthermore, the database will also be tested for its ability to store and retrieve information generated by the admin/user such as any edits created by the admin, register requests, login requests, contacts requests, information on events and workshops, and etc.

Here, the web-page component will be tested on its ability to direct users to intended page destinations, showcase up-coming events and workshops along with correctly showing any information associated, displaying relevant information from the search function, and correctly displaying any necessary actions permitted for each user. Furthermore, the

web-page component will need to have the ability to perform on multiple devices and screen resolutions adequately.

Out of all the components listed above, this component will by far contain the most features, and so, the testing phase on this component will be very feature-driven. Here, the timeline-component will be tested on features such as the scrolling feature, content-display feature, sub-timeline feature, filter feature, search feature, photocard feature, image-scroll feature, calendar feature, device detection, and edit/add content feature for admins. In addition, the timeline will also be tested on its accessibility features such as text-to-speech, speak-to-text, a color-blind feature, and signifiers.

Dependencies

Risks and assumptions

Tools