



# Software Testing Plan Version 4

---

Team Anubis

March 30, 2017

Sponsor:

Cindy Brown on behalf of the  
NAZ Animal Welfare Task Force

Faculty Mentor:

Steven Jacobs, Lecturer SICCS

Frankie Berry  
Steven Gruenewald  
Marjorie Hahn  
Riley Shelton  
Matthew Siewierski

# Table of Contents

<b>Introduction</b>	<b>2</b>
<b>Unit Testing</b>	<b>2</b>
<b>Integration Testing</b>	<b>4</b>
<b>Usability Testing</b>	<b>5</b>
<b>Conclusion</b>	<b>9</b>
<b>References</b>	<b>9</b>

# Introduction

This document describes the software testing plan for the NAZ Animal Welfare Task Force Pet Adoption Tool. This web application will provide an interface for NAU students to think critically about whether or not they are ready to adopt a pet and, if they are ready, then the system will inform them on what kind of pet would be a good fit for them [5]. The primary goal of this application is to help prevent irresponsible students from adopting pets that they are not ready to have, which will hopefully lead to a decrease in the number of adoptable animals that are abandoned or return to shelters in Northern Arizona.

First, this document will describe the unit testing that will take place on the system. This will be done primarily through testing suites integrated within the DNN content management system [2], which will test specific modules in order to ensure that they are functioning as intended. Next, we will describe the integration testing that will be taking place in order to ensure that all of the separate parts of the system work together correctly. The final method of testing that we will deploy in order to ensure that this system will be useable for our sponsor and our target audience will be usability testing. This will help to ensure that target audience (students) will be able to use the system effectively and also that our sponsor (the NAZ Animal Welfare Task Force) will be able to maintain the system after we are gone.

## Unit Testing

This section describes the various unit test(s) that will be used to measure the completeness and accuracy of the functional capabilities of the system. Due to the nature of the system, our use of unit testing will be limited in comparison to usability and integration testing detailed later in the document. This is because of our use of a

content manage system for the front end of the system. Each of the modules will be tested using basic though a try/catch system and the debugging mode offered by DNN. The following is our unit testing plan of the different modules of our system.

- Unit Test Plan
  - Test Items
    - Questionnaire module (cshtml file)
    - Registration module (ascx file)
    - Modify Questionnaire module (cshtml file)
    - Reports module (cshtml file)
    - Educational Tooltips module (cshtml file)
    - Shelter Notifications module (cshtml file)
  - Features to be tested
    - The files described in the test items section will be tested based on their postconditions, postconditions meaning the final state once the module has pulled/pushed the information from the database.
  - Test deliverables
    - The following items will be delivered in the final report once testing is complete:
      - Unit testing report
      - DNN content management system Debugging Log
  - Testing tasks
    - Before each test, the test cases must be documented along with the input and expected behavior.
    - During the tests, the behavior is recorded for each input.
    - After the tests, a unit testing report will be written that contains the test case, input, expected behavior, and actual behavior observed during the test.
  - Environmental needs

- All tests will be run on all major web browsers (i.e. Chrome, Edge, Firefox, and Safari), along with mobile devices (Android and iOS).
- Tests will be done using basic try/catch functions in C#.
- Grading Criteria
  - A test case will pass when the output of the test matches the expected value and no error has occurred.
  - A test case will fail when the produced output of the test raises an error and/or the output is not the expected value.
- Resolution
  - Failed test cases will result in code fixes and be retested until the test case passes.

## Integration Testing

This section covers the integration testing of the system. Our system has taken the evolving prototype approach to software development and as a result, integration testing has been occurring throughout the development lifecycle. Due to the web environment of the system, all components must conform to HTTP format standards. Additionally, due to DNN architectural constraints, components must be integrated and developed according to DNN.

We have divided the integration testing of our system into key points in order to address the core components and determine whether or not they are working properly.

Integration points:

Database Functionality: Access to a database is a key aspect of the system, specifically for DNN, the front end and database content management system. NAU's IT department has assured us that there is plenty of room in the server's

database for the needs of the website. Therefore, running out of room should not be a concern.

Test harness used: DNN built-in system

Verification: To verify database functionality we make use of an SQL console [4] which handles queries as well as SQL management software in order to validate the pushing and pulling of test data.

Module Interaction: All custom modules used in the system are built inside the DNN content management system. Each of the different modules interact with both DNN and the backend database to achieve functionality.

Test harness used: DNN built-in system

Verification: Module interaction verification is accomplished automatically through the DNN content management system.

Performance Testing:

We will run a performance test before releasing the system of 20 test users in order to ensure that the website runs efficiently. If not, we will consult with NAU's IT department to get the resources that the website needs.

## Usability Testing

This section describes the usability testing of the system. Due to the focus of our system on interactions between the software system and the end user, usability testing is critically important and our goal is to account for this by examining the overall quality and accessibility of the user interface exposed by the system.

In order to conduct usability testing, we will use three groups of NAU students who will be able to give detailed feedback on our user interface design choices. In order to

collect feedback during usability testing, we will create a Google Form [3] that will allow each student to rate their experience with the website, and what they would improve upon given the opportunity, see Figure 1 below for details. This will make the process of collecting feedback from these large groups of students much easier. After feedback collection, we will reevaluate the system in order to see how we can incorporate that feedback into the next iteration of the system. The three groups of students are described in further detail below:

1. CS477/577 Advanced User Interfaces Class Students:

This group of NAU students has technical user interface and usability experience [1] and thus will be able to give reasonable feedback based on what they know to be feasible for the system. They may also have suggestions on helpful miscellaneous tools that we might not be aware of, as well as possible solutions for any bugs in the system.

TESTING DATE/TIME: TBD

2. School of Communications Capstone Team:

This group of NAU students is not as technically-savvy as the previous group, and thus should be able to provide helpful feedback on how we present information on the website and how we should expect users to respond to different cues. Due to their background in communications, they will also be able to provide feedback on how effectively we communicate our ideas, hopefully allowing us to perfect the user's experience on the website.

TESTING DATE/TIME: March 23rd 2:20PM

3. NAU Chain Gang Club:

The Chain Gang is very involved with NAU events and traditions. They will be an excellent source of feedback for us regarding what will help attract NAU students

to this website and what will encourage them to use the system.

TESTING DATE/TIME: April 17th 8:00PM

It will also be important for us to give a demonstration of how to use the product to the sponsor and allow them to get comfortable with using the system. This way, the sponsor can see if there is anything unclear on how they are expected to maintain the system, and then we will be able to adjust accordingly before delivering the final product.

TESTING DATE/TIME: April 18<sup>th</sup> 7:00PM



## PetReady? Usability Survey

\* Required

How would you rate the website's overall layout and design? \*

1 2 3 4 5

Poor      Excellent

How would you rate the website's navigation? \*

1 2 3 4 5

Poor      Excellent

On a scale of 1-5, was the educational information on the website easy to understand? \*

1 2 3 4 5

Poor      Excellent

On a scale of 1-5, was questionnaire on the website easy-to-follow and straightforward? \*

1 2 3 4 5

Poor      Excellent

Do you believe this website will attract potential pet adopters? \*

Yes

No

Maybe

Please leave any miscellaneous feedback and observations you had while browsing the website and completing the questionnaire.

Your answer

**SUBMIT**

Never submit passwords through Google Forms.

Fig 1. Screenshot of Feedback Survey

## Conclusion

We will thoroughly test the system in order to ensure that it works as intended, will be acceptable for use by our sponsor, and will be utilized by NAU students. Specifically, we will use unit testing in order to ensure that each module works as intended, integration testing to ensure that each of the modules will work together, and usability testing to ultimately ensure that the system is user-friendly for the target audience, NAU students.

After we have run through each iteration of testing, we will evaluate the test results and see what we can do to incorporate those results into our final software design. Ideally, this iterative process of testing, evaluating, and incorporating would occur many times before the final product delivery, in order to ensure that the product is as robust as possible. However, given the time remaining, it is likely that we will only be able to go through this process a few times before delivering the system to the sponsor.

## References

1. *Designing the User Interface, 6th edition*, Shneiderman et al, Chapter 5: "Evaluation and the User Experience", (2017), pp. 139-179.
2. DNN, A CMS Equally Loved by Marketing and IT. Retrieved February 5, 2017 from: <http://www.dnnsoftware.com/>
3. Google Forms. Retrieved March 27, 2017 from: <https://www.google.com/forms/about/>
4. Microsoft SQL Server. Retrieved March 28, 2017 from: <https://www.microsoft.com/en-us/sql-server/sql-server-2016>
5. PetReady? from: <http://petready.org.nau.edu/>