Final Report

Team Anubis

May 11, 2017

Sponsor:
Cindy Brown and Ken Lamm 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

Introduction

This capstone project, the *NAZ Animal Welfare Task Force Pet Adoption Tool*, tackles the issue of abandoned and return animals in Northern Arizona. Too often, irresponsible students adopt dogs and cats without considering the responsibility, time, and cost involved in owning a pet. They also do not take the time to think about living restraints, pet deposits, or allergies. This has led to a large number of returned or even abandoned animals across Northern Arizona. Our goal is to decrease this number by educating and informing students on everything that goes into adopting an animal, so that fewer students make the decision to adopt before considering if it is the best choice for them.

Therefore, we, Team Anubis, have built a website designed to examine the students that desire to adopt an animal to determine if they are truly ready to do so. This website will also connect students to volunteering and fostering opportunities that they may not be aware of. It will also match students to an animal personality type, so that they can get a better idea of what kind of animal would be a good fit for them.

We have designed this website for the NAZ Animal Welfare Task Force, an organization dedicated to improving the lives of animals in Northern Arizona. Their members will use this website as administrators in order to monitor the website's activity via reports, add additional shelters to the system, and ensure that the shelters are getting all the necessary info from the website's questionnaires.

This document will discuss our process in creating this website, the requirements that we set out to fulfill, the architecture that used to design the system, how we tested the system, the project timeline that we followed, and the future work that will be coming for this project.

Process Overview

This section will overview the process we used to develop the system including all tools and artifacts used to aid in the development process. The first semester of this project was entirely used to plan the project. We initially split into the different roles of Frankie Berry the editor, Steven Gruenewald the recorder, Marjorie Hahn the team leader and customer communicator, Matthew Siewierski as the release manager and Riley Shelton as the architect. Then, we chose the software tools we were going to use. We started with the plan to use Django and MongoDB for our system, but ended up changing to a DNN content management system and ASP.NET frontend due to the NAU ITS servers requiring a Microsoft-based solution. We changed from MongoDB to Microsoft SQL because we needed a relational database.

During our development process we used Slack to keep in constant communication with each other. We used Trello to record the tasks that needed to be accomplished. The entire site was backed up using a GitHub repository. Throughout the first semester we only met with our mentor biweekly and the second semester we met weekly. We met with our sponsors monthly to keep them updated. Our team overall had perfect attendance (excluding emergencies).

Requirements

The following functional and non-functional requirements are summarized from the original Requirements Document created for this project.

Functional Requirements:

- The system will provide basic profile creation, login, and logout functionality for all users.
- The system will assess the user's lifestyle to determine if they are ready to adopt a pet.
- The system will match the user to animals in the system. (This was modified to animal personalities since we were not able to match to specific animals.)
- The system will use educational pop ups to inform students on the responsibilities involved in adopting an animal.
- The system will generate reports on website activity viewable by an administrator.
- The system will notify local shelters when a student has completed the questionnaire and is interesting in adopting an animal.
- Administrators will be able to modify the questionnaire in order to change the information that is sent to interested shelters.
- The system will display adoptable dogs and cats from Second Chance Center for Animals and Coconino Humane Association.

Non-Functional Requirements:

- The system will be maintainable for a non-technical administrator.
- The system will have a responsible interface.
- The system will be thoroughly documented to aid in maintainability.

Architecture and Implementation

This section will go over the architecture and implementation for the system. Below in figure [] is the overview of the high-level architecture. To being all users will connect to the system through the use of a web browser and an internet connection. Once a connection is established the

Microsoft-IIS server will create a new instance of the application allowing the user to view the application. After the application has started ASP.NET and DNN content management system take over to display the content of the application back to the user. Allowing them to interact with the system. During all stages a constant connection with the MS SQL server is required to display any assets back to the user and to perform the necessary actions for functionality.
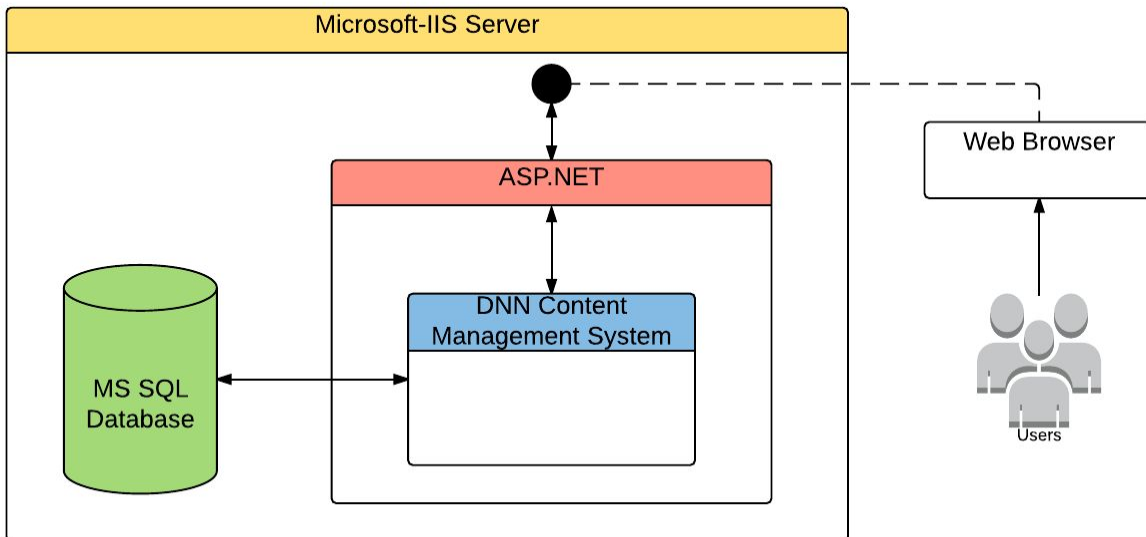

The following are a detailed description of the different modules that are utilized to achieve functionality in the system outlined in figure[].

- User Management
  - DNN built in module
  - This is a DNN module that allows for user registration, login, logout and user rolls. We used this to define the two different user groups registered, and admin. This allowed us to restrict the pages that are viewable by the different user groups. Password recovery is also provided by this module, allowing users to change their password if they were to forget it. Being a registered user is required to achieve functionality.
- Maintain Questionnaire
  - ASP.NET C# module
  - The maintain questionnaire is composed of three different modules one for each of the main functionalities: add, update, and remove. Each of the different modules query the database using an SQLDataReader to read and write. Admin users are only able to only maintain the Readiness Questionnaire, this is due to it being of our own design. The Meet Your Match® Canine and Feline-ality questionnaires are based off of existing forms and research so they are not able to modify them.
- Manage Partners
  - ASP.NET C# module
  - The manage partners is composed of three different modules one for each of the

4

main functionalities: add, update and remove. Each of the different modules query the database using an SQLDataReader to read and write. The partners are used to as the recipients of the notification from each of the questionnaires if the user wishes to notify the shelter. We allow the user to choose to notify, so that it would like the amount of notification that a shelter will receive. It also allows the user to retake the questionnaires as many time as they wish.

- Base Questionnaire
    - ASP.NET C# module
    - The main Readiness Questionnaire queries the database for the list of questions, this allows admin users to update and maintain the questionnaire. A user completes the Readiness Questionnaire to see if they are PetReady e.g. if they have a large enough budget, live off campus. After completing the questionnaire the results from the form are compared to stored values to assign a value to each answer. The answers are then summed together to get the readiness value. If the user opted into sending their results to the partnered shelter. An email will be sent to the shelters with the answers from the form and if they are ready or not. The following information is stored: User_ID, and Readiness Value.
- Meet Your Match® Canine-ality (mymdog)
    - ASP.NET C# module
    - A custom module based off the existing paper form created by the ASPCA. It queries the database for the list of questions, this was done to make it easier to change if the ASPCA changes any of the questions. It allows the user to complete the questionnaire to see what color of dog (Purple, Orange, Green) best fits them. It pulls the values from the form that the user selected and calculated the different values for green, purple, and orange. It then calculates the max of the three colors. If the user opted into sending their results to the partnered shelter. An email will be sent to the shelters with a the answers from the form and the color that they are matched with. The following information is stored: User_ID, Purple, Orange, Green, and the color.
- Meet Your Match® Feline-ality (mymcat)
    - ASP.NET C# module
    - A custom module based off the existing paper form created by the ASPCA. It queries the database for the list of questions, this was done to make it easier to change if the ASPCA changes any of the questions. It allows the user to complete the questionnaire to see what color of cat (Purple, Orange, Green) best fits them. It pulls the values from the form that the user selected and calculated the different values for green, purple, and orange. It then calculates the max of the three colors. If the user opted into sending their results to the partnered shelter. An email will be sent to the shelters with a the answers from the form and the color that they are matched with. The following information is stored: User_ID, Purple, Orange, Green, and the color.
- Readiness Display
    - ASP.NET C# module

- ○ The Readiness Display module queries the database to see if the user has completed the base questionnaire. If they have not it will display an icon to prompt them to complete it. Once they have completed the questionnaire it will display if they are PetReady or not, if they are not it will let them retake the questionnaire and prompt them to apply to foster/volunteer at the local shelters. Otherwise it will say they are PetReady and allow them to complete the mymcat and mymdog questionnaires to see what color of dog/cat would be the best fit form them. We also let the users that have been not to be PetReady to complete these questionnaires as a reward for taking the questionnaire. We leave the adoption up to the shelters, are primary objective is to educate the user on the cost and needs for adoption. Once they have completed either of the mymcat or mymdog the results are displayed back to them. All this functionally is by querying the database.
- ● Fostering/Volunteering
  - ○ DNN Form and List module
  - ○ This form was created using the preinstalled Form and List module. It allows admin users to add and change the form to meet what they need quickly and easily. We have CAPTCHA enabled for all un-registered users to prevent spam and bots. It also allows the admin to change how the emails are sent to.
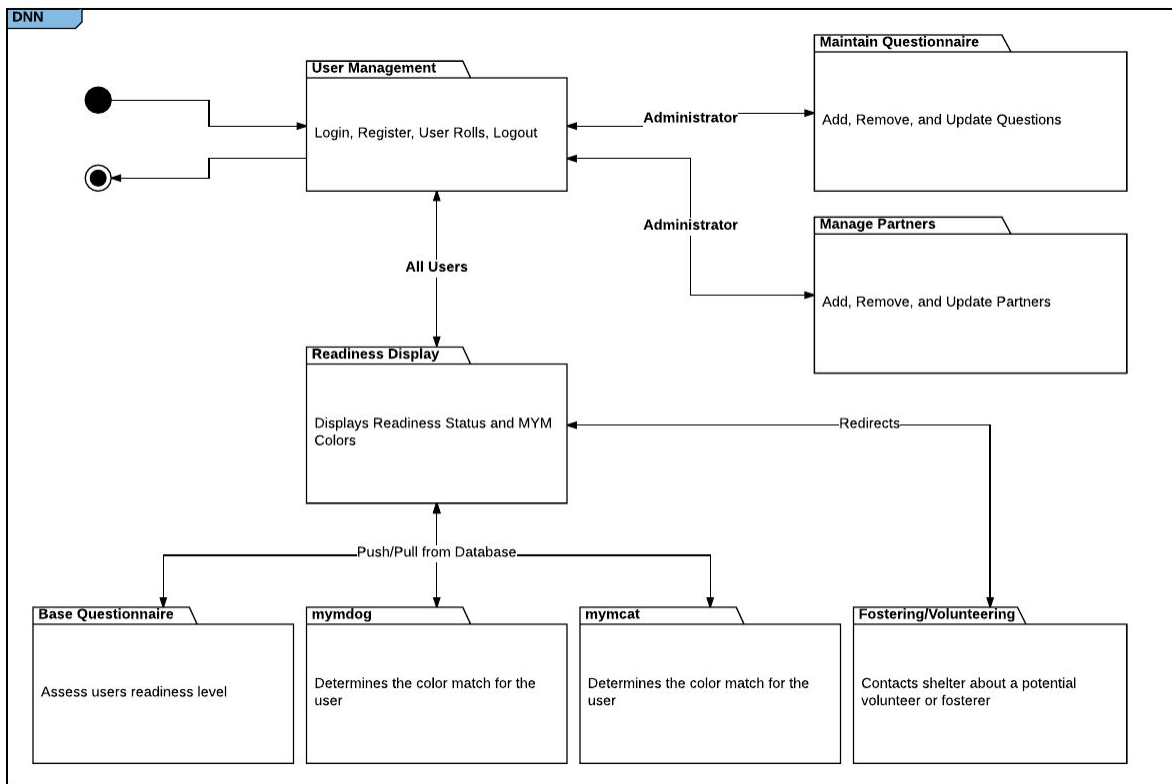
A change in architecture that happened during the development was a shift from attempting to match users to individual animals to the broader animal personalities defined by the ASPCA Meet Your Match® program. This change was due to the limitation on developing an API that

could get the animal data from PetPoint a 3rd party software that is utilized by a large percentage of shelters. Primary the ability to develop it in a way that would be maintainable for a non-technical user, was the main reason for the shift in architecture. A possible solution was to have the animals entered individually but this was decided to be to labor intensive to make the system usable.

Another change in architecture is the information that we stored in the database, such as animal information and responses. For the animal information this change came from the decision to match to the personalty groups rather than the individual animal. It was to labor intensive to add them individually, and not maintainable to develop an API to pull the information from the 3rd party application. Regarding the responses, it was a tradeoff between the dynamic generated questionnaire that allow the questions to be easily modified and being able to easily store the information for later retrieval. We came to the conclusion that a dynamically generated questionnaire and limited response storage would still offer the ability to generate usable reports. Another factor was that the information is sent to the shelter on completion if they oppt to sent it, so all the information is still being sent in a more useful way.

Testing

This section covers the testing strategy which was used to validate our implementation. The testing process consisted of unit testing, integration testing, and usability testing with a focus on usability testing specifically given the nature of the system.

Unit testing used in the validation of our implementation was fairly limited due to the fact that we used a content management system for the front end of the system. Each of the modules that comprise the system were tested through a basic try and catch system as well as the debugging mode offered by DNN. Specifically, using the DNN built-in functionality, the unit test plan involved analyzing a series of preconditions and postconditions for key modules and verifying the results against expected output.

Integration testing of our system has been occurring throughout the development lifecycle due to the evolving prototype approach we have taken. Like unit testing, much of the integration testing was handled by the built-in functionality of the DNN content management system. Specifically, we broke integration testing into two key integration points, which were database functionality and module interaction. These aspects of the system were both tested using the DNN built-in testing harness and were verified either through the use of an SQL console in the case of database functionality, or automatically for module interaction.

Usability testing was the focal point of our overall testing plan; due to the focus of our system on interactions between the software system and the end user, usability testing was regarded as

critically important and our goal was to account for this fact by examining the overall quality and accessibility of the user interface exposed by the system. Specifically, we conducted usability testing by using three groups of NAU students who were able to give detailed feedback on our user interface design choices. We made use of a Google Form to collect feedback that allowed each student to rate their experience with the website, giving suggestions on improvements or issues they encountered.

After conducting the series of tests, there were some key results which we reviewed and incorporated into our final iteration of the system. Both unit testing and integration testing were conducted without any issues which allowed us to focus on our usability testing feedback. The key feedback brought to our attention by users was the length of the questionnaire in the system. After considering the fact that a number of users suggested that the questionnaire was too long, we decided to break the questionnaire down into three separate forms, which was well-received and implemented as a part of our final iteration of the system. Usability testing also resulted in a number of minor interface changes such as button colors or text clarification. Ultimately, the usability testing step of the testing process was the most beneficial as it allowed us to refine the key functionality of the system.

Project Timeline

- Build profile creation/manipulation modules (2/10 - 2/17)
- Build user assessment module (2/20 - 2/27)
- Build educational tooltips (2/27 - 3/1)
- Database integration (2/22 - 3/10)
- Implement matching algorithm (4/4 - 4/11)
- Implement report generation (4/4 - 4/17)
- User Testing (4/17)
- Implement shelter notifications (4/17 - 4/21)
- Turn over to sponsors / sponsor training (5/5)

Implementation started with a basic user interface that allows you to create a new account, login, and navigate to the other parts of the site. After this, the readiness questionnaire and the Feline-ality and Canine-ality questionnaires were made. Educational tooltips were added with the questionnaires. Once the questionnaires were finished the readiness values for the readiness questionnaire and the color values for the Feline-ality and Canine-ality questionnaires were assigned. Report generation was implemented after the questionnaires were finished, then the shelter notifications were implemented. Modules were tested after they were completed, and will continue to be until the project is delivered. As of right now, the project is complete, so it just needs to be given to the sponsor.

Future Work

This project will likely have an exciting future ahead of it. NAU ITS plans on migrating the site to its own .org web space in the future. There is also the possibility of migrating the site to an Azure cloud-server at some point. The

There is a Public Relations capstone team in the School of Communications planning a release for the website as well. This will include social media exposure, flyers, events, etc. The official release of the website will take place early next Fall.

Conclusion

Adopting a pet while at school without realizing the commitment needed to properly take care of the animal is a commonly seen and growing issue with students. This project aims to specifically target NAU students--a group especially guilty of adopting then returning/abandoning pets, and help reduce the number of pets adopted by irresponsible students, thereby reducing the number of pets that end up back in a shelter.

Through the use of a responsive web app, potential adopters will be screened via a questionnaire in order to make sure they are in a position to properly care for an animal. It will also educate them about the responsibilities involved in adopting so as to avoid any misinformation which may lead to returning or abandoning their potential future pet.

This document is focused on reporting the final results that this capstone project produced. Specifically, the document covers the project's processes, requirements, architecture, timeline, and future work.

We are confident that our sponsor can use this system to educate students on the responsibilities of adopting an animal, as well as contribute in helping more animals find their forever homes in Northern Arizona.

Glossary

- Admin User

    - A user that oversees the system by updating and maintaining the website.

- Content Management System

    - A Content Management System (CMS) is a computer application or program that supports the creation and modification of digital content using a simple interface to abstract away low level details unless required. A CMS usually supports multiple users working in a collaborative environment.

- Default User

    - A typical/normal/generic user (pet adopter) that interacts with the system.

- Digital Content

    - Digital content refers to the website and backend database for the system.

- Framework

    - The underlying structure of the system. More specifically, the framework is the software that the system is designed in and around.

- Readiness (Base) Questionnaire

    - A questionnaire given to the user to assess their lifestyle and readiness in order to determine if they are ready to adopt a pet.

- Canine-ality Questionnaire

    - A questionnaire based off the paper form designed by the ASPCA, to match a user to a type of animal personality for Dogs.

- Feline-ality Questionnaire

- ○ A questionnaire based off the paper form designed by the ASPCA, to match a user to a type of animal personality for Cats.

- ● PetPoint

  - ○ A shelter management service used by the larger shelters in northern Arizona. E.g. Coconino Humane Society, Second Chance Animal Shelter.

- ● PetReady?

  - ○ Current name for the pet adoption website tool.

- ● Responsive Web Application

  - ○ A web application that has an interface that changes according to the screen of the user. e.g. runs on a desktop, laptop, tablet, or smartphone.

- ● Shelter/Partner

  - ○ An animal shelter or shelter is a place consisting of stray, lost, abandoned, or surrendered animals which are mostly dogs and cats.

- ● System

  - ○ The Northern Arizona Animal Welfare Task Force PetReady tool.