September 30, 2019 – June 12th, 2020

Western Oregon University

Computer Science Program

CS460-462 \ Software Engineering I, II, III

Senior Capstone Course

Tricia Holman + Corrin Moser + Joey Tavares

Faculty Advisor: Dr. Becka Morgan

https://github.com/Stormy9/wou-cs_team-nerdvana/tree/master/Petopia

The Reasons For Petopia

We decided to develop the Petopia app for our project for a few reasons:

1. Tricia had used a similar (*but very much narrower in scope than Petopia*) idea in one of the capstone courses for the Chemeketa CIS program, Database Development; then built on that database for another capstone course at Chemekta, Systems Analysis, devising the plans to develop that database into the core of a web app (*also very much narrower in scope than Petopia!*)

   So taking those earlier efforts to develop Petopia, our full-fledged and functional app, gave us a nice “jumping off” spot.

2. We were able to fairly easily figure out ways in which we could build Petopia so that it met all of the requirements for the Senior Capstone Project.

3. We thought that developing Petopia would give us some interesting aspects to develop, like logins/user accounts, fun and functional user & pet profiles (*which includes handling user photo uploads*), different “roles” or types of users (*Pet Owners & Pet Care Providers*), building a robust “transaction system”, incorporating search algorithms to make recommendations, and implementing an API for payments; just for starters.

4. We felt like this could be the basis for an actual business, based on models such as Care.com, and extending the scope of Wag.com and Rover.com – something that people might actually like having, that might actually fill a real need.

5. **Most importantly**, though, all three members of Nerdvana really love, enjoy, and appreciate pets and their people; and we thought Petopia would be an app that we could have a lot of fun developing and bringing into reality. We were correct!

Summary: Tech | Languages | Tools | SDLC

We used a team discord for team communication, brainstorming, and problem-solving.

We used GitHub for version control & team collaboration.

We used Visual Studio & Visual Studio Code to write the actual source code.

We used C# & LINQ, HTML/CSS & BootStrap, JavaScript/jQuery as our languages.

We used Microsoft SQL Server Management Studio to alter/update, query, and otherwise manage our database tables — and to generate our final ERD.

We used Azure & Kudu (*the latter does not work!*) for deployment and hosting of our app and it's connected database.

We used Google Drive for our non-GitHub team documents — and gmail for our Petopia site: for sending notification emails to our users about their appointments … and for users to contact us.

We used Pivotal Tracker to manage our user stories, sprints, timelines, which developer was doing what, what's done/what still needs doing — the overall development & progress of Petopia.

We used TOAD for the first three versions of our ERD design.

We used Mirro for our early diagrams — mind-maps, use case diagrams, etc.

We used Stripe API in the Care Transactions part of our Petopia app.

We used the simplezipcode API for our zipcode proximity finder (**https://github.com/alexmaris/simplezipcode**)

We followed Disciplined Agile Delivery practices in the engineering and development phases.

Our Favorite Parts

The really good, fun, and/or cool stuff, and the most-challenging stuff... parts we're particularly proud of, or really like, or anything we really want to say -- things we want to point out or highlight as being really freaking cool.

Corrin: I'm really proud of how pictures are uploaded using our app. It was a lot of work to get it right, and it wasn't something that a simple google search fixed. Took a lot of research and reading around to figure out how to do it. And finally getting it working the way that I wanted it to work felt amazing.

Joey: The overall feel and flow of the site turned out great. You are met with a really welcoming homepage and a functional one at that, being able to search up user profiles as a visitor lends so well to the initial experience on the webpage. The way a user has both their own profile to represent themselves but also individual pet profiles ended up working great and enabled us to build many of the other features from there.

I was happy that we were able to use the initial planning periods to get situated our design, in the end our decisions paid off and despite a few minor hiccups our design helped us and we were able to continue building and scaling the application through the sprints.

Tricia: I love the whole thing, but I think my favorite part is the user & pet profiles – bringing those to fruition from the wireframe Corrin drew as we all figured it out way back in the beginning. Also the Pet Photo Gallery.

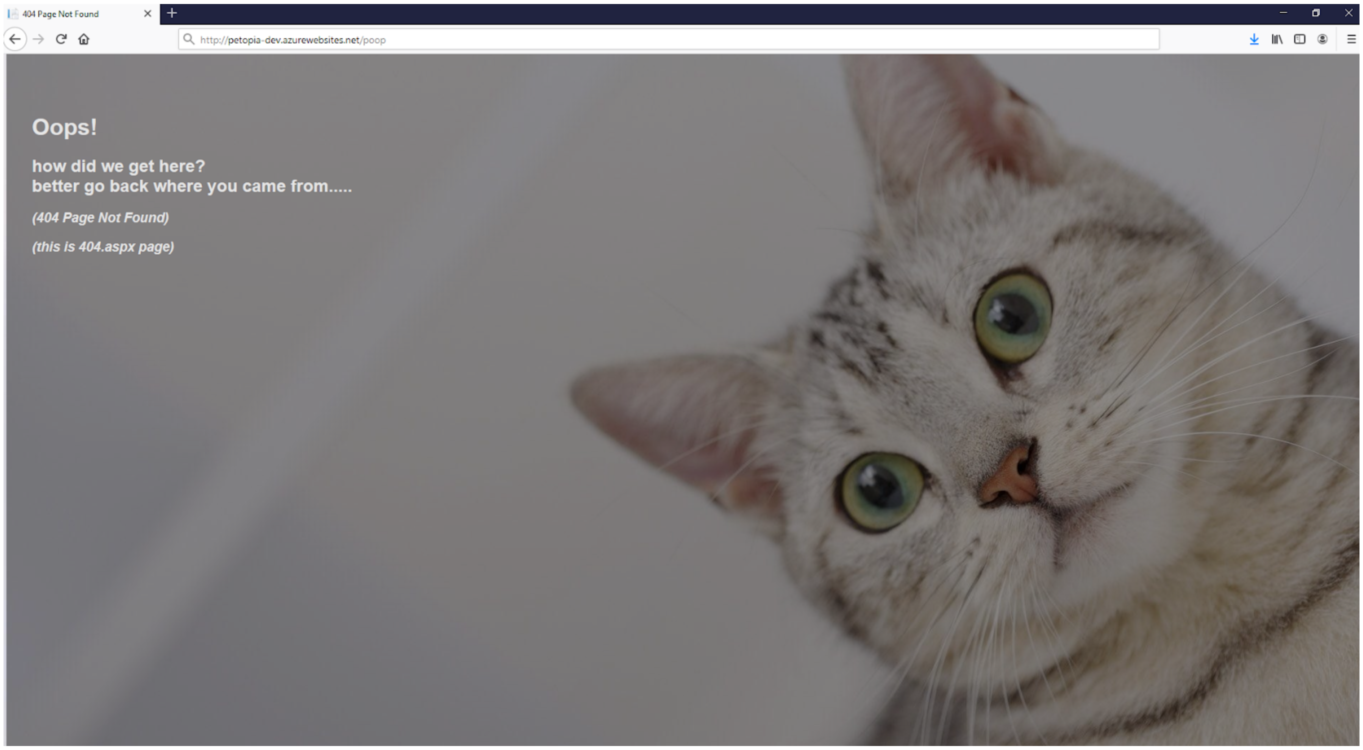
I also had a lot of fun figuring out the flow of, and then building, the "Care Transactions" system. Getting all of those interactive parts working correctly and smoothly was a bit of a challenge – but a fun one!

The very best part of Petopia though, was getting to work with Joey, Corrin, and our faculty advisor, Dr. Becka Morgan..... That is what made this one of the best experiences of my life so far!

Final Technical Requirements Checklist

❑ **No Yellow Screens of Death -- at least have a catch-all error page -- specific ones (*404, 500, 400, 401, etc*) are preferred.**

➔ Here’s our 404 page, as an example (obviously we have no controller or view called “poop”):



Oops!

how did we get here?
better go back where you came from.....

(404 Page Not Found)

(this is 404.aspx page)

❑ **Not susceptible to XSS attacks (reflected or stored) ... Not susceptible to XSRF attacks**

➔ When three other teams performed alpha-testing for our site, they tried to implement these attacks — and were unable to succeed.

❑ **All input forms (POST || PUT || DELETE) must be protected by either a login -*OR*- captcha**

➔ Our only user input form fields that are NOT behind a login, are the two search boxes on our front/splash page, where unregistered visiting users can search to see if there are either Pet Care Providers in their area, if they are a Pet Owner … or, to see if there are Pet Owners in their area, if they are thinking of being a Pet Care Provider.

Those two search boxes on the front page have strict validation on them — using the `**pattern**` attribute on the `**Html.SearchBox**` — so that ONLY a five-digit number (zip code) can be entered -- the form/search will not fire unless input matches that pattern.

- ❑ **No unprotected forms that modify our database, that could be abused by scripts -- including registration page (*Selenium?*)**
  - ➔ We use Captcha on our registration page, and other than that, all forms are behind a login.

    We also use validation on all of our form inputs, at the `**Model**` level where we can, and otherwise at the level of whatever `**Razor Html Helper**` is being used for that form input. (sometimes a combination)

❑ **No passwords, API keys, or other private data can appear anywhere in Git repo .....
If one ever WAS committed to the repo at ANY time, track it down & kill it (*plus change the password/key/whatever*) so no one can ever snoop through past commits and get a password!**

- ➔ Our team never got an email telling us that Azure detected a key on a repository somewhere so we can assume that our keys/private data has not been committed anywhere.

  Plus we are not, after all, *complete* idiots, and retained the information on how to hide our application secrets from CS460.

❑ **Solitary UP & DOWN sql scripts that will build entire db outta nothing and wipe it totally out -- ONE for each**

- ➔ **Our up.sql:**
  https://github.com/Stormy9/wou-cs_team-nerdvana/blob/master/Petopia/Petopia/Petopia/App_Data/userUp.sql
- ➔ **Our down.sql:**
  https://github.com/Stormy9/wou-cs_team-nerdvana/blob/master/Petopia/Petopia/Petopia/App_Data/userDown.sql

❑ **Generate an ERD for our final database schema -- save it as an image file -- put it in our repo -- make a link to it from our front page repo README**

- ➔ **https://github.com/Stormy9/wou-cs_team-nerdvana/blob/master/Petopia/Petopia_models-diagrams/FINAL_PETOPIA_ERD.png**
  It is also linked from our repo front-page README

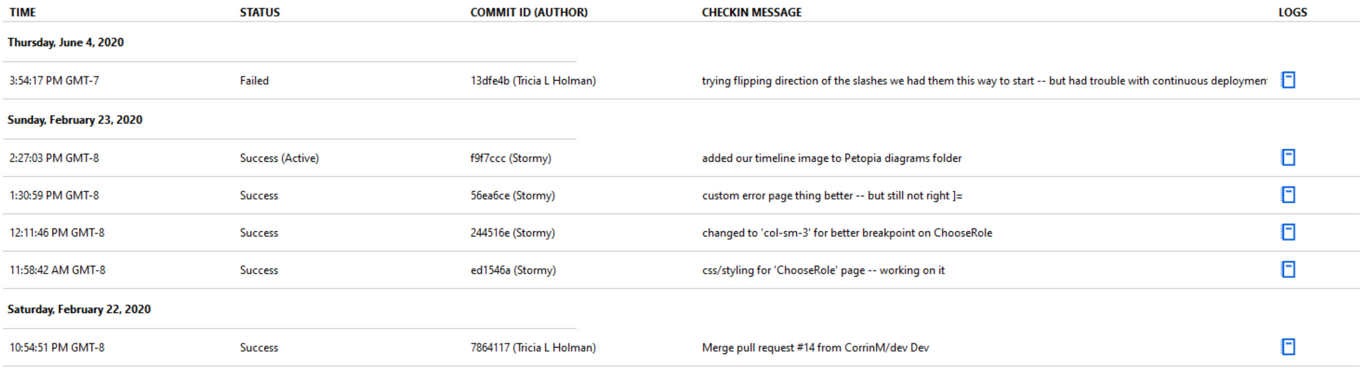
❑ **Demonstrate Continuous Deployment ➔ screenshot of Azure portal showing that we currently have -- or have recently had -- Continuous Deployment**

➔ From the beginning, we had continuous issues with Continuous Deployment. It was only ever “continuous” for two or three pushes, and then would consistently fail to deploy with subsequent pushes.

So... we took down our CD (disconnected it in Azure), and re-set-it-up from scratch, several dozen times. We double-checked our `.deployment` file a couple dozen times — and tried it with the slashes in both directions (since we found examples with them going different directions)... nothing would work. The “continuous” part of “Continuous Deployment” just would not work consistently — and not at all since late February.

It seemed to be a Kudu issue, but we were never able to glean anything useful from searching on any of the info we found in the Azure log files when CD failed.

Here’s a screenshot showing one of the brief moments it worked:



TIME	STATUS	COMMIT ID (AUTHOR)	CHECKIN MESSAGE	LOGS
Thursday, June 4, 2020				
3:54:17 PM GMT-7	Failed	13dfe4b (Tricia L Holman)	trying flipping direction of the slashes we had them this way to start -- but had trouble with continuous deploymen	🗎
Sunday, February 23, 2020				
2:27:03 PM GMT-8	Success (Active)	f9f7ccc (Stormy)	added our timeline image to Petopia diagrams folder	🗎
1:30:59 PM GMT-8	Success	56ea6ce (Stormy)	custom error page thing better -- but still not right]=	🗎
12:11:46 PM GMT-8	Success	244516e (Stormy)	changed to 'col-sm-3' for better breakpoint on ChooseRole	🗎
11:58:42 AM GMT-8	Success	ed1546a (Stormy)	css/styling for 'ChooseRole' page -- working on it	🗎
Saturday, February 22, 2020				
10:54:51 PM GMT-8	Success	7864117 (Tricia L Holman)	Merge pull request #14 from CorrinM/dev Dev	🗎

❑ **Professional-looking source code -- no sloppy code! All code not “trivially obvious” (to an ASP.NET MVC5 & JavaScript programmer) needs to be commented.**

➔ Not sure how to detail or prove this here..... but our actual Petopia source code resides here — so check it out:
https://github.com/Stormy9/wou-cs_team-nerdvana/tree/master/Petopia/Petopia/Petopia

❑ **Don’t store any user-generated files *(images, pdf’s, etc)* on your web server -- i.e., the App_Data folder**

➔ **Of course we don’t!**
https://github.com/Stormy9/wou-cs_team-nerdvana/tree/master/Petopia/Petopia/Petopia/App_Data

❑ **Page linked to from GitHub repo home page (the readme) that explains all major steps needed to install our app and get it up & running**

- ➔ There is no need to install our app in the traditional sense — not simply to use it, anyway — it is a web-based app and not desktop.
  - ➔ All you need to do, to make use of Petopia, is:
    - ➔ Install a web browser -- preferably Chrome, Opera, or Edge
    - ➔ Go to: **http://petopia.azurewebsites.net/** and look around, or better yet check out our “How to use Petopia” page: **http://petopia.azurewebsites.net/Home/UsingPetopia**