Reynerio Sarmiento

CS443 Project

Pet Clinic Application

**Project Statement:**

When I was deciding what to do for this project, I was thinking of a project idea that would not take as much time based on the deadline requirement for the course, but would also be fun while making it. I tried to come up with some ideas, but then I just ended up looking at the Professors suggestions. I came upon an idea that intrigued me, a pet related app. I was interested in because I have a pet at home and it would be cool if I had an app where I could check for his vet clinic. People who have pets, such as I do, would really benefit from apps, like the one I made. I would like to make an app for a vet clinic one day to help customers get a better access to them. I do believe there are similar apps that do something like this, like PetCo or PetSmart.

In my app, I named it “Rey’s Pet Clinic.” In the beginning, it asks for username and password. However, before you can login, you need to register. It asks for your name, email, username, and to type your password in twice. If a username already exists, then it would ask you to change it. Also if you typed in the password wrong the second time, it’ll ask you to redo it. Once you have registered, you will be able to login and it will direct you to the menu page. From the menu page, you can select three options: Vets, Adopt a Pet, and Win Free Prizes.

In Vets, you will be able to see all the Vets that work for the Pet Clinic, with their respective information. You can also add them to your contacts, and the information will be passed over. From Adopt a Pet, you are able to switch between activities, to check out the Cats and Dogs that are available. You can also select the dog or cat that you want, and you will get a notification that you have adopted the pet, and the pet’s status will change from “Adopt Me” to “Has already been Adopted.” Finally, I added the Win Free Prizes section, which you can pick from two games to win prizes. The two games I implemented was a True or False Quiz, and a guessing numbers game. For the True or False, it asks you 20 questions, true or false, and you have to decide if the question given is true or false. If you get it correct, your points are increased by 1. If you get to 17/20 points or more, you are directed to the prize page. However, if you are not able to get 17/20 points, then you are given a chance to try again. The other game I implemented was the guessing the numbers game. You are to choose 5 numbers from 0 to 10. Once you have chosen, the application randomly generates 5 numbers from 0-10. If you guess them correctly, then you are directed to the prize section. The prize section gives you an image of your prize and asks for your home address, city, and state. Once you have entered the information, it gives you a notification that the prize is on its way.

**Application Design**

In my module, I added resource files and java files. The Android components I have used are Intenets, Bundles, SQLiteOpenHelper, setting up TextViews, Buttons, and EditTexts. The components are organized within their respective java files. Some of my Activities have a great basis of layout because I took my time to make it look like how I wanted. This was basically bu using Linear Layouts (horizontal/vertical orientations), TextViews (sifted to the right or left), EditTexts(making it password sensitive), and orientation stuff around the screen to where I wanted it place. My application targets smartphones, because it easier to use on a smartphone than Ipad.

**Application Implementation and Design**

I defined various classes for this project. Most of the classes correlate to one of the XML files. I felt like it was easier for me to organize which belonged to a certain XML file. For example, I have the activity\_dogs class that correlated to the activity\_dogs XML file, since it opens the layout onCreate().

For the start of the app, I defined DataBase.java, MainActivity.java, activity\_register.java, and registerInfo.java. All these connect with one another. To connect from the MainActivity to the activity\_register.java I used a method that reads in the information from the XML file from the “onClick” and decided if the Button was clicked or not. From the registerInfo.java, I made a DataBase that helps store the information typed onto the register activity with the help of SQLiteOpenHelper. However, before doing that, I had to initialize each data entry, as you can see in the registerInfo.java file. Finally, in the MainActivity, I had to check if the username entered is in the database, and that if the password given matches any username. If they do, an Intent is used to go to the Menu Activity.

Menu Activity was just using Intents based on the Button Click, from activity\_home.java. First Activity, you could open is Vets.java. In Vets.java, all I did was pass information from the given Activity, such as the image, firstname, lastname, phone, and email. This information gets passed to the activity\_addascontact.java file. From there I used a method that looks for a “onClick” on the XML file. If found, an intent is done where it inserts the vet’s information onto the contact app on the phone to where it’s supposed to go. Activity\_dogs and cats is a series of Intents with Toast messages, when a certain Button is clicked.

The fun part was working with the games that would direct you to a prize. I used many onClickListeners in this part of the application because I was able to add certain cases for each click. For example, in the guessing the numbers game, you can increase the number by clicking the “+” button and decreasing the number with the “-”button. However, in my code you’ll see that the Button have various cases, depending on what the random numbers generated are. For this game I used GuesstheNumbers.java and congrats.java.

The True or False game was harder. I had to create 20 different strings with their respective Boolean answers, done with some indexing, as you can see in quizQuestions.java. In trueorFalse.java I used the methods from quizQuestions and referred to them to see if the question outputted on the screen of activity\_pickagame XML file was a true or false question. Based on the Button you clicked, using an onClickListener, you had to check if the answer chosen paired up with the actual answer. If they did, then you increased the points by 1 and generated a new question by going to the next string. Else, you just generated a new question and went on to the next String, but you didn’t add a point to the score. At the end you just check if you reached the end of the array of strings, and check if the points you got the end were greater than or equal to 17. If they were, then you are directed to prize page, if not then you can try again, by selecting the Button that is made visible.

Getting errors was very frustrating for this project. I had simple errors like a misspell on my code, that took hours trying to find out why my application was not building. When I got errors, I tried to comment out parts that I thought was disrupting the program, part by part and rebuilding. Other errors I have encountered were problems with XML ids were not correlated correctly. I also had some problems with connecting activities together because I later noticed I had some problems with the Activity it was trying to get to. To fix this, I tried connecting the activity with a blank activity, and it worked, so I figured out that it was the other Activity. So I started debugging my code to find the errors. As of right now, I think my project works perfectly fine, and there are no crashes or errors.

**References**

* Google
  + Finding general java information, setting up strings, numbers, etc
* Stack OverFlow
  + Some errors that I came across with SQLiteOpenHelper
* Cs443 WebPage
  + Basic Concepts like Intents, Toast Messages, etc.

**Experience and Thoughts**

I really enjoyed this project. I feel that I learned a lot from this project on how Android Stuidos work between classes and XML files. I understand how to setup Toast Messages, Buttons, TextViews, onClickListeners, SQLiteOpenHelper, etc. I understand more on how java can help us implement Buttons and other key elements of Android Studios with certain cases.

**Some ideas I had**

* Adding a shopping cart where you are able to add the pets you have adopted and prizes you have won
* Add a random Image Generator for the prizes you could have won, besides the pee pads that are automatically set
* Have some previous data that would get implemented onto the adoption Activity, where certain dogs/cats would have been already adopted.