Daniel Goncalves

05/20/2021

CS 443 – Pets App Document

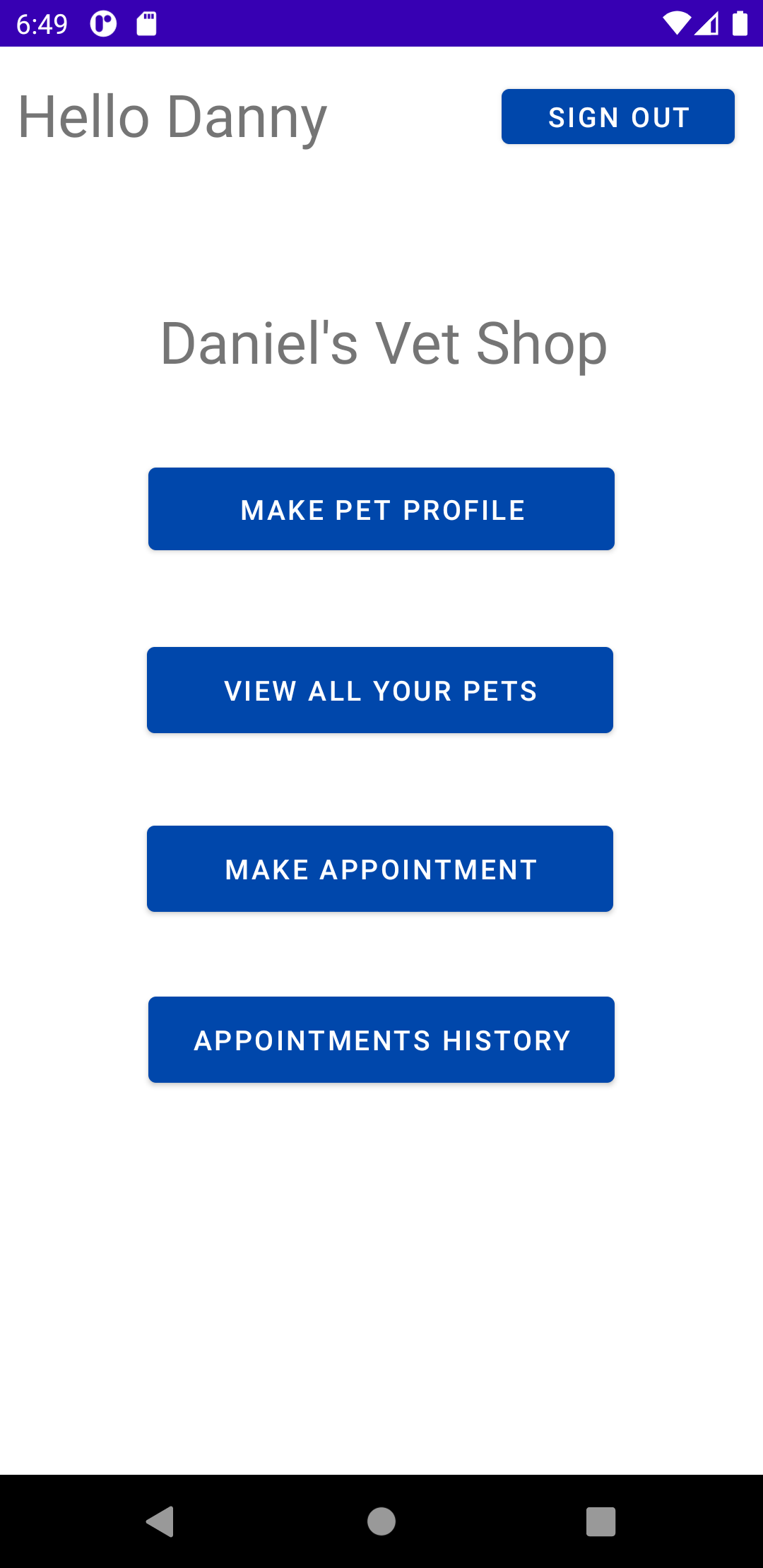
1. **Project Statement**

The app that I decided to build for my final project is a pet contact app. The purpose of this app is for users to make appointments with a vet shop. It’s not for emergency appointment such as for example, your dog ate chocolate or is severely injured, and you need an emergency appointment. I would’ve liked to implement a feature that would happen those types of situations. The app is built for regular/ non-emergency appointments. For example, your pet needs a follow-up, need a vaccination, or needs a normal examination. The reason I wanted to build this app is because I wanted the experience of building somewhat of a real-world application. The typical user for this app is pet owners who need to schedule an appointment their pets. A similar app to mines does exist in the Android Market, which is the Petco app. My pet app does have a special requirement. When the app initially loads up, it prompts the users to login. If the user does not have an account, then they can register which only takes their name, an email address and password.

2. **Application Design**

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, application, Teams

Description automatically generated

Calendar

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

Login Page Register Page Home Page Create Pet Page

Make Appointment Page Appointment History Page

In this project I used the activity, notification, thread, and network modules. I used the activity to switch between each activity. When the app loads and were in the login/register activity. There’s a way switch between the two activities in case the user needs to register. Once the user registers or logs in, then it triggers a new activity, which is the home activity. The register and login activity are now removed from the back stack. The only way back to the login activity is by signing out. I use the notification module to notify the user if there were any problems. If the user is trying to register but the password strength is weak, then it will notify that the password is weak. Similarly with the registration. There can not be blank field in the registration/login activity, and I also use toast to notify the user. I briefly use the thread module within the create pet and create appointment activity. If a pet/appointment was successfully created, then I use the “thread.sleep” method to wait 2 seconds then remove current activity from back stack and send user to home activity. In this application, I use Firebase Firestore to store data inputted by user. It stores all the appointments and pets created by the user. I frequently had to access the database to retrieve necessary information to display on the UI. To achieve this, I used an asynchronous class Task to retrieve and parse the data. The app mainly targets smartphones, and I don’t use external services.

3. **Application Implementation and Evaluation**

I used firebase authentication to handle the sign in/ signup functionally. It also handles the encryption of user’s password. With firebase authentication, there are various ways to sign in. For example, you can use your Google, Facebook, or GitHub accounts to sign in. For this project I choose to sign in with email and password. I used firebase firestore as the database to store the user’s data. I have previous experience using firebase in the past with Flutter, so the configuration and set up was like before. I tested the functionally of each feature soon after that feature was completed. I create multiple account to test the different scenarios for that feature. The app should usually always work as intended if the firebase functions I used in the app don’t change or get deprecated. An issue that I notice during testing is that the error messages for login/register isn’t always correct. For example, I can type correct email address but wrong password. There will be a toast notification of an error but instead of “wrong password”, it will say “bad email formatting”. I used firebase authentication error message as the output error message.

4. **References**

[Firebase FireStore Documentation](https://firebase.google.com/docs/firestore)

[Firebase Authentication Documentation](https://firebase.google.com/docs/auth)

[Android Spinners Documentation](https://developer.android.com/guide/topics/ui/controls/spinner)

[Android Task Documentation](https://developer.android.com/reference/com/google/android/play/core/tasks/Task#addOnSuccessListener(java.util.concurrent.Executor,%20com.google.android.play.core.tasks.OnSuccessListener%3C?%20super%20ResultT%3E))

5. **Experiences and Thoughts**

This project was fun to create, and I learned a lot about the Android SDK. As expressed earlier, I wish I could have built an emergency appointment feature. Also, while making an appointment, you only choose a date and not a time.