## Final Project Report

Name: Joao C. Barbosa

Link: https://github.com/barbosajcarlos/TermProject-PetPatientPortal

**Project Description Document**

A pet contact app for a veterinarian clinic or animal hospital. The App will be able to maintain each Pet’s profile and the history of vet visit. It will require Database design, Searching/filtering functions, and UI design.

1. **Project Statement**

**Pet Patient Portal App** (PPPortal APP) is an application intended for a Veterinary clinic or animal hospital to add and maintain each Pet Patient profile and record the history of vet visit/s. This App is targeted for Android Smartphones.

Each Pet Patient will be either added to the database as a New Patient or looked up by entering their name and DOB (must enter both as it was entered in the database due to validation). Once a New Patient is entered a pet profile will be created and stored in the firebase database. Otherwise, if the Patient is a returning Patient, its information is stored in the database and can be retrieved.

Once a Patient Pet profile is created, it is possible to record the reason for current visit and to access Visit History of that individual pet. Each time a Pet shows up for a visit, a new Visit History will/can be recorded by entering info in the text view Enter Reason for Visit.

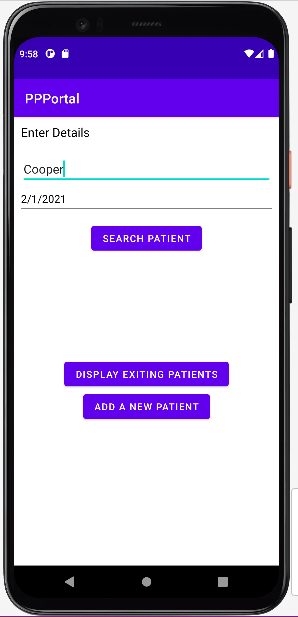
PPPortal APP will benefit Vet hospitals/clinics by storing and maintaining critical information about Pet Patients and their visit/s history. This will save Vets time as information will be readily available to access, so they can start treatment right away and skip all the unnecessary number of questions.

**2. Application Design and Functionality**

Graphical user interface, text, application, Teams

Description automatically generated

Design and functionality: The figure to the left is the main screen for the PPPortal App. It displays the main functions the App performs --> Search Patient, Display Existing Patients and Add New a Patient. Each function will take user to a new screen depending on user needs.

**Case scenario**: Search Patient 🡪 If user wishes to search for a Patient in the databases, user **must enter** Pet Name and DOB.

If user enters either Pet Name or DOB only, a toast message will be displayed: Please enter DOB if user forgets to enter DOB. Similarly, if user only enters DOB, a toast message will be displayed: Please enter Pet Name.

When user enters all information required there will be two scenarios: Patient exists in the Database or Patient doesn’t exist in the database yet.

Graphical user interface, application

Description automatically generated

If **Patient doesn’t exist**, a toast message will be displayed: No patient found. Please add new patient and user will be taken to a screen where they can Add New Patient.

If **Patient exists in the database**, a toast message will be displayed: Patient found.

User will be taken to a new screen showing Patient’s profile. Patient profile screen will have two buttons: Record Visit and a Visit History button.

User can Record Visit by entering a reason for the visit in the text field above the Record Visit button.

Graphical user interface, application, Teams

Description automatically generated

Graphical user interface, application, Teams

Description automatically generated

User can also access/view Visit History by clicking the button and a new screen will display all visits that have been recorded for that Patient. Visit History screen will show reason for Patient’s visit, date, and time.

Table

Description automatically generated with medium confidence

**Graphical user interface, application

Description automatically generated**

From the main screen, if user clicks on Display Existing Patients, a new screen will open showing all existing patients. User may view Patients from there as well instead of search them by name and DOB.

Now, if user clicks on Add a New Patient button on the main screen, a new screen will open, and user will have to input information about the new Patient so it can be added to the database.

Android components used for the project: Activity, Intents, toast messages, firebase database (Firebase automatically performs all the data input output operations in the background thread), Networking using firebase, permission to use camera and internet.

1. **Application Implementation and Evaluation**

App was tested well to see if it worked the way it is intended. The Firebase database isn’t behaving as it is supposed to all the time. There were times when new Patient profiles were created, but they did not get saved in the database right away, but sometime later when checked they were in there. It took some time for the info to become available. This happened a couple times. I suspect firebase dataset was going through a system maintenance, but not entirely sure as there were no alert messages on their end. Other times the app worked perfectly; data was saved in the database at real time.

**JAVA:**

The following are the classes that were defined and implemented in the App.

PatientListAdapter: This class is to bind the list of existing patients in the database to the UI component recycler view

ViewHolder: is a nested class in the patient list adapter that handles the binding of data of individual cells in the list

VisitHistoryAdapter: this class is to bind the list of patients visit history to the UI component recycler view

PetData: This is a data model class that contains all the required fields or properties of a Pet

VisitHistoryData: This is a data model class that contains required properties to represent visit history data of a particular patient

AddNewPatientActivity: the screen contains all the fields required to take input to create a new patient in the data base

MainActivity: Main screen that shows up when the app opens, it has different options to add a new user, display the list of existing patients and search for existing patients

PatientDetailsActivity: This activity displays the details information of a particular pet it also has a functionality of recording a visit of existing patient

ShowExistingPatientsActivity: Activity that displays list of all patients in the databases

VisitHistoryActivity: displays visit history of patient with the reason and time of visit.

**XML**

activity\_add\_new\_patient.xml: this is a UI part of adding a new patient screen

activity\_main.xml: this is a UI part of main screen

activity\_patient\_details.xml: this is a UI part of patient details screen

activity\_show\_existing\_patients.xml: this is a UI part of showing existing patient screen

activity\_visit\_history.xml: this is a UI part of visit history screen

item\_existing\_patient.xml: this refers to a single cell in existing patient list screen

item\_visit\_history.xml: this references to a single cell in the visit history screen

**4. References**

The Final project was completed by using class content, referencing sample code and Android Documentation and firebase online reference.

<https://developer.android.com/>

https://firebase.google.com/docs/firestore/quickstart

**5. Experiences and Thoughts**

The project was quite challenging, I got close to what I expected, but missed some functionalities I had in mind due to time constraint. There were some other features that I had envisioned to have on the App but ran out of time. I wanted to have a daily scheduled appointments button that would display Patients and time of appointments for the chosen date. I also wanted to have an option to have patient profile information updated, so in case Patient moves, changes contact info and so on we would be able to update it. Also, a functionality that allows sharing patient visit report through customer’s email or text message.