



College of Engineering

## CS CAPSTONE PROGRESS REPORT

MARCH 19, 2019

# APP TO SUPPORT FIELD DIAGNOSTICS IN VETERINARY MEDICINE

PREPARED FOR

## OREGON VETERINARY DIAGNOSTICS LABORATORY

PROFESSOR CHRISTIANE LOEHR

PREPARED BY

## GROUP 72 MALSANO

KATHERINE JEFFREY

BRANDON JOLLY

BRADFORD WONG

### Abstract

Currently, there are many difficulties for veterinary pathologists trying to perform remote diagnostics. There are not any effective ways for people out in the field collecting samples to communicate with specialized experts located in laboratories. As a result, this project will involve creating an Android mobile application that will be used as a bridge to connect the field personnel with the veterinary pathologists in laboratories. With this mobile application, the field personnel will be able to take pictures of the individual that is being analyzed and then send the pictures along with other data such as the patient, location, and time to a pathologist. The pathologist will then be able to use the provided information to perform a necropsy and send feedback to the field personnel. This project is intended to support remote field diagnostics in veterinary medicine.

## CONTENTS

<b>1</b>	<b>Project Purpose and Goals</b>	<b>2</b>
<b>2</b>	<b>Progress and Plans</b>	<b>2</b>
2.1	Completed Parts of Project . . . . .	2
2.1.1	Database . . . . .	2
2.1.2	Android . . . . .	2
2.1.3	Website . . . . .	3
2.2	Parts of Project Left to Complete . . . . .	3
2.2.1	Database . . . . .	3
2.2.2	Android . . . . .	3
2.2.3	Website . . . . .	4
<b>3</b>	<b>Problems</b>	<b>4</b>
<b>4</b>	<b>Code and Images</b>	<b>5</b>

## 1 PROJECT PURPOSE AND GOALS

The completed project components are intended to serve as a means of communication between personnel in the field and pathologists in the laboratory. The project's purpose is to improve a team's ability to perform remote diagnostics by providing a convenient way for teams to communicate information. The OVDL wants a native Android mobile application that collects field data and images, stores the information on a native SQLite database, sends the information to the lab's MySQL database using an API provided by the client, and gets real time feedback from the lab. The lab can interact with the database and user's submissions through a website, but the website is a stretch goal.

## 2 PROGRESS AND PLANS

### 2.1 Completed Parts of Project

#### 2.1.1 Database

We have completed the implementation of the MySQL database and the SQLite database on the Android app. In MySQL, we have added all the required tables. We have changed the image table to be renamed as pictures to avoid any confusion with the image object in Android Studio. All of the necessary primary key constraints have been included in each table. Each of the table has the required foreign key constraints for that table. For example, the PICTURE table has a foreign key from the SUBMISSION table. This code is shown in Figure 1.

The SQLite database also has the required tables implemented. They do not have any foreign key constraints because this is handled by the Android programming. It will then be further checked once the information is sent to the MySQL database through the use of the client's API. The android app has the required procedures to insert data into the SQLite database, and to grab information from the tables to be displayed to the user.

#### 2.1.2 Android

The home screen has been completely implemented. All of the navigation buttons have been added and they all take the user to the corresponding screens. Additionally, the toolbar has been implemented and can take the user to any of the relevant screens.

Clicking on the "Create Account" button successfully opens up a web browser. It takes the user to the registration page on the website.

For the create submissions screen, all of the input fields were added. The "Save Draft" and "Submit" buttons correctly store the submission data into the SQLite database. Additionally, the "Add Pictures" screen has been implemented. Users can upload up to five pictures using either the camera or the phone's gallery and can also delete the pictures they uploaded. Also, the application can track all of the meta data of the pictures, including the name of the picture, the date it was taken, and the GPS coordinates. Users can also navigate to the "Add Samples" screen where they can add information about their samples. They can add as many samples as they want and can delete them as well. Clicking on the check mark button on the "Add Samples" and "Add Pictures" screen will save the pictures and samples.

For the "View Drafts" screen, all of the drafts stored in the database can be seen in a list. Additionally, users can remove drafts from the database by long clicking on the entry in the list. Clicking on one of the drafts in the list will navigate the user to the "Create Submission" screen. The information that the user entered before saving the draft will be automatically filled out for the user.

For the "View Submissions" screen, the user can view all of the submissions that they submitted to the database in a list. The user can also click on an entry and the app will successfully navigate to that submission's detailed page. The detailed page shows all of the information the user entered when creating the submission. This includes the title of the submission, group name, date the submission was created, the comment, the sick element, pictures, and samples. Like the View Drafts screen, the user can delete submissions from the database by long clicking on the submission in the list. Each entry in the list also shows the relevant information such as the submission title, the case ID, and when the submission was created.

Clicking on the "Instructions" button on the toolbar will take the user to the "Instructions" screen. This screen tells the user how to use the app and the contact information of the OVDL.

Clicking on the "Settings" option in the toolbar will navigate the user to the "Settings" screen. On this screen, users can log in and log out by clicking on the corresponding buttons. If the users try to create a submission without being logged in, the app will prompt the user to log in first.

### **2.1.3 Website**

Although the website is a stretch goal, the team had time to work on parts of the website. All the pages of the website have been created with basic layouts. The main page is the submissions page that has a table of all the submissions available to the user. The table shows the submission title, the name of the user that submitted it, its case ID and the date it was received. When clicked the individual submissions open a case detail page with the rest of the submission data. This is where messages will be processed. Users will be able to send messages about the submission to the submitter's phone and see any replies.

There is also an account page where users can see and change their username or password. An about page is also included to provide users with information about the site, the app, and how to use them or contact people who can assist them. The pages are all accessible to each other through a navigation bar at the top of the pages. The website is currently hosted on the client's server and the domain name is myvetpath. It will be connected to the database with an API provided by the client.

## **2.2 Parts of Project Left to Complete**

### **2.2.1 Database**

For both databases we will need final confirmation on any names of the table. We will also need to do some field test with their information to insure we are not missing any information during collection. We would need to create any views the client will need to make it easier for the API to query information once it has been completed. We also need to test to insure the API is properly sending information to the MySQL Database. This also means we need to test to the other direction as well. Those are the major things we need to add to the Database, with a few bug fixes along the way.

### **2.2.2 Android**

The Android app has all the required core functionality except for the tasks that use the API. These tasks aren't complete yet because the client promised the team that he would create the API, but he is still developing it at the moment. As a result, the team hasn't been able to work on any tasks that involve it. This includes authenticating log in credentials, sending data from the phone to the server, and receiving comments on submissions from the MySQL database. Lastly, there are some bugs involving drafts and the toolbar that need to be fixed.

### 2.2.3 Website

The website is still a stretch goal that the team would like to continue to work on in the future. The website has not been connected to the database because the API is still being developed by the client. Once it is ready to connect, some choices about the login and registration process will need to be made. The clients will need to decide how much security is necessary and specify the differences between user groups. We have discussed having levels of privileges for general users, OVDL affiliated users, and OVDL veterinary pathologist users. It is still unclear how security levels will be assigned and enforced.

Another important piece that is still unclear is messaging. We know pathologists will need to send messages to users about their submissions, but it is unclear if pathologists will also send messages to each other or how security levels will affect the messaging system. Unstable network connectivity is also a concern so we will need to implement a way of handling potential internet interruptions.

## 3 PROBLEMS

The first issue we had was one of our team members was unable to push anything to our Android repository. He was able to pull information but when he tried to push it would say he did not have authorization. We checked the repository to see if he was added as a collaborator. That was not the issue. We met as a group to see what was wrong. We had him reinstall Android studio and tried switching Github accounts. We eventually figured out that the issue was that his laptop kept autofilling one of his other Github accounts information.

An issue we had during testing the submissions table in our SQLite database was that any changes we made to the SQLite submission table would not be shown in the Android emulator. We tried using different emulators, but the same error would occur. The solution was that we needed to recreate the tables for the changes to occur. It did enforce our belief the database was not being created every time we opened the app though.

A permission issue we all had was connecting to the MySQL database from outside the server. If we used putty, we could get to the server and interact with the tables but we could not interact with the table without using putty. We tried direct connections but we kept getting errors stating we did not have permission. The solution this time was found out by our client, who discovered we needed to create an ssh tunnel on the server in order for us to connect outside of it. Once this was created we had no more problems connecting.

When we were testing to see if we could add submissions to the SQLite database we could not figure out why any new additions could not be displayed. We used logs to see the flow of how a submission was processed in our app and determined it was being added correctly, but we simply could not retrieve the information that was stored. With the help of the logs we then determined to look deeper into the function used to retrieve the data from the SQLite server. Turns out the function had a logic error and was trying to display something that could not happen. Once we fixed the logic problem in the function, the submissions were properly displaying.

Another issue that we had was that we haven't been able to use the API yet because the client is still developing it. As such we have not been able to test the connection between the Android app and the database server and we haven't been able to complete any API related tasks. Since we finished the rest of the core functionality early, we tried to assist him with the API, but we are still having problems with retrieving data from the MySQL database.

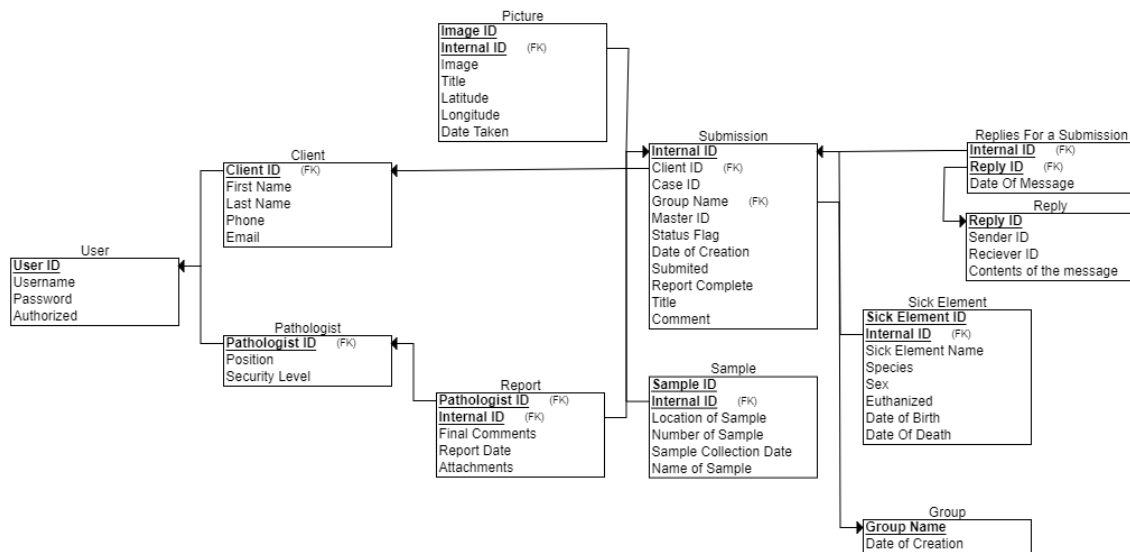
## 4 CODE AND IMAGES

```

CREATE TABLE 'PICTURE' (
  'ImageID' int(11) NOT NULL,
  'MasterID' int(11) NOT NULL,
  'ImagePath' varchar(255) NOT NULL,
  'Title' varchar(255) DEFAULT NULL,
  'Latitude' bigint(20) NOT NULL,
  'Longitude' bigint(20) NOT NULL,
  'DateTaken' date NOT NULL,
  PRIMARY KEY ('ImageID', 'MasterID'),
  KEY 'MasterID idx' ('MasterID'),
  CONSTRAINT 'MasterIDForPicture' FOREIGN KEY ('MasterID') REFERENCES 'SUBMISSION' ('MasterID')
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=latin1 ;

```

**Fig. 1:** Code to create PICTURE Table in the MySQL Database



**Fig. 2:** Beta ER Diagram

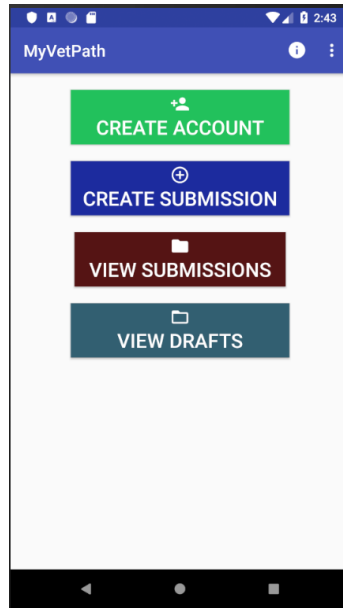


Fig. 3: Home Screen

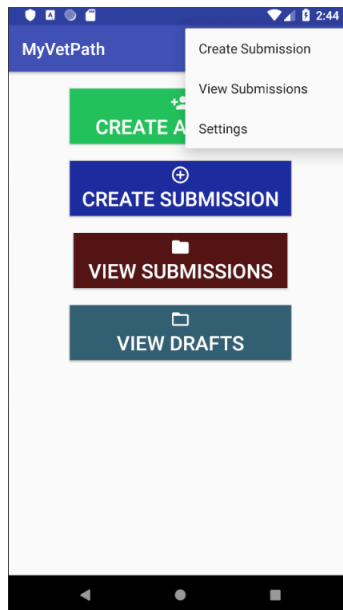


Fig. 4: Toolbar

**Create Submission**

**Submission Title\***  
Demo Submission

**Group Name**  
Group #72

**Sick Element Name**  
Sick Element Name Demo

**Species\***  
Dog

**Sex:** Female ▾

**Euthanized** ☒

**Date of Birth**  
Animal Born On Friday, March 1, 2019

**Date of Death**  
Animal Died On Monday, March 18, 2019

**Comment\***  
This is a demo comment that provides more information about this submission.


 **SAVE DRAFT** **SUBMIT**

Fig. 5: Create Submissions Screen

**Create Submission**

Demo Submission

**Group Name**  
Group #72

**Sick Element Name**  
Sick Element Name Demo

**Species\***  
Dog

**Sex:** Female ▾

**Euthanized** ☒

**Date of Birth**  
Animal Born On Friday, March 1, 2019

**Date of Death**  
Animal Died On Monday, March 18, 2019

**Comment\***  
This is a demo comment that provides more information about this submission.


 **ADD SAMPLES** **SAVE DRAFT** **SUBMIT**

Fig. 6: Create Submissions Screen (Part 2)



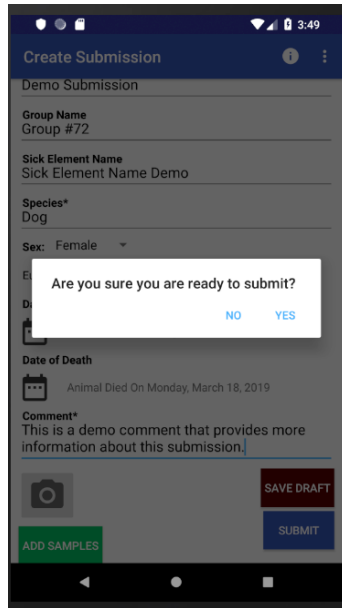


Fig. 7: Create Submissions Screen (Part 3)

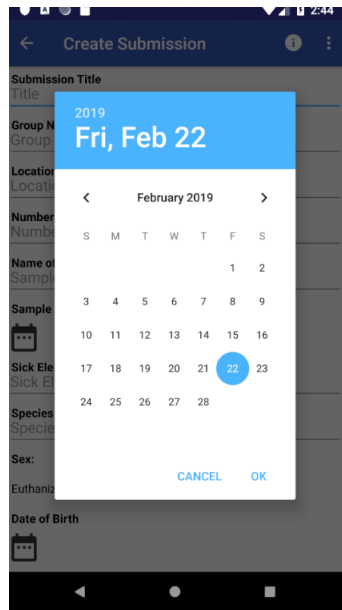


Fig. 8: Date Picker

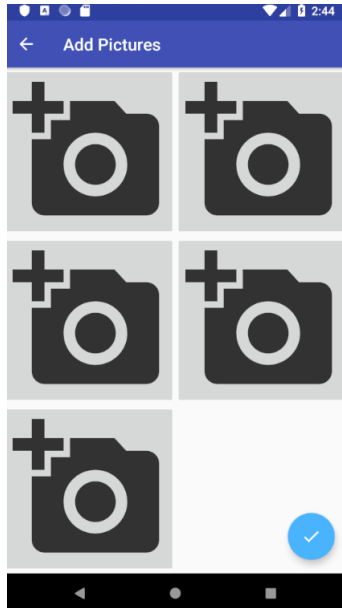


Fig. 9: Add Pictures Screen

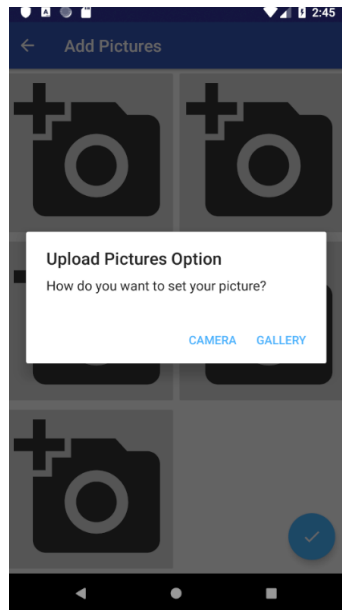
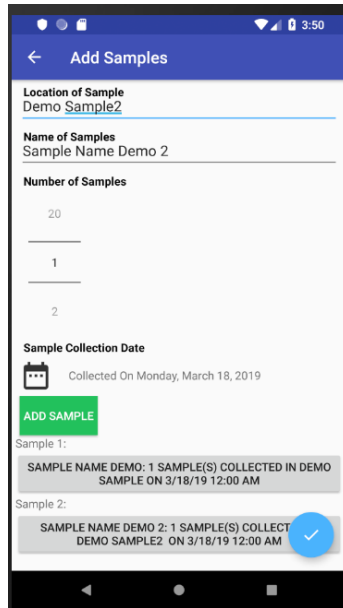


Fig. 10: Add Pictures Prompt



The screenshot shows the 'Add Samples' screen of a mobile application. At the top, there is a blue header bar with a back arrow and the title 'Add Samples'. Below the header, the form contains the following fields: 'Location of Sample' with the value 'Demo Sample2', 'Name of Samples' with the value 'Sample Name Demo 2', and 'Number of Samples' with a value of 20. A green 'ADD SAMPLE' button is located below the 'Number of Samples' field. Below the button, there are two sample entries. 'Sample 1' shows 'SAMPLE NAME DEMO: 1 SAMPLE(S) COLLECTED IN DEMO SAMPLE ON 3/18/19 12:00 AM'. 'Sample 2' shows 'SAMPLE NAME DEMO 2: 1 SAMPLE(S) COLLECTED IN DEMO SAMPLE2 ON 3/18/19 12:00 AM' with a blue checkmark icon to its right. The bottom of the screen shows the Android navigation bar.

← Add Samples

Location of Sample  
Demo Sample2

Name of Samples  
Sample Name Demo 2

Number of Samples  
20  
1  
2

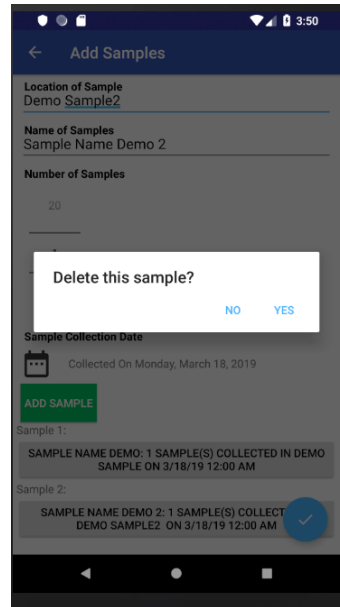
Sample Collection Date  
Collected On Monday, March 18, 2019

ADD SAMPLE

Sample 1:  
SAMPLE NAME DEMO: 1 SAMPLE(S) COLLECTED IN DEMO  
SAMPLE ON 3/18/19 12:00 AM

Sample 2:  
SAMPLE NAME DEMO 2: 1 SAMPLE(S) COLLECTED IN DEMO  
SAMPLE2 ON 3/18/19 12:00 AM

Fig. 11: Add Samples Screen



This screenshot shows the same 'Add Samples' screen as Figure 11, but with a modal dialog box overlaid in the center. The dialog box is titled 'Delete this sample?' and has two buttons: 'NO' and 'YES'. The background of the screen is dimmed, and the 'Sample 2' entry at the bottom has a blue checkmark icon next to it.

← Add Samples

Location of Sample  
Demo Sample2

Name of Samples  
Sample Name Demo 2

Number of Samples  
20  
1  
2

Sample Collection Date  
Collected On Monday, March 18, 2019

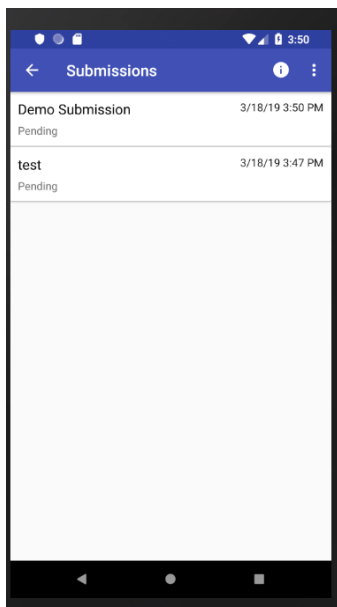
ADD SAMPLE

Sample 1:  
SAMPLE NAME DEMO: 1 SAMPLE(S) COLLECTED IN DEMO  
SAMPLE ON 3/18/19 12:00 AM

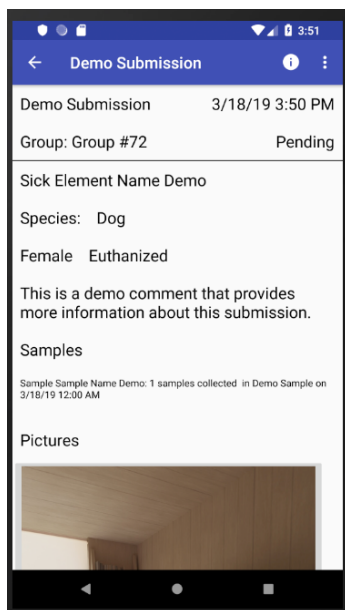
Sample 2:  
SAMPLE NAME DEMO 2: 1 SAMPLE(S) COLLECTED IN DEMO  
SAMPLE2 ON 3/18/19 12:00 AM

Delete this sample?  
NO YES

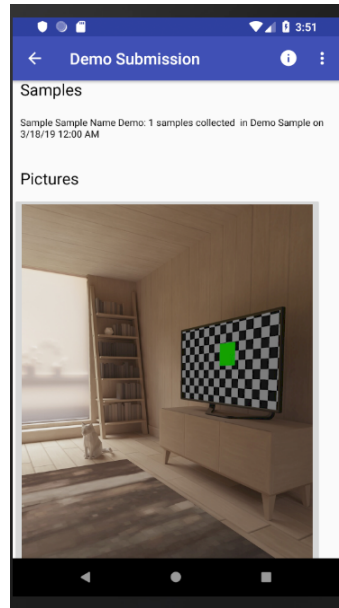
Fig. 12: Add Samples Screen (Part 2)



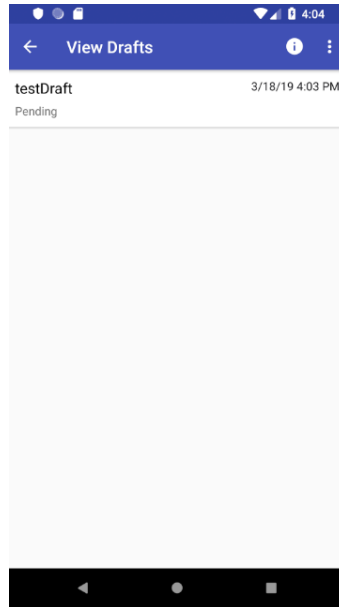
**Fig. 13:** View Submissions Screen



**Fig. 14:** Detailed Submissions Screen



**Fig. 15:** Detailed Submissions Screen (Part 2)



**Fig. 16:** View Drafts Screen

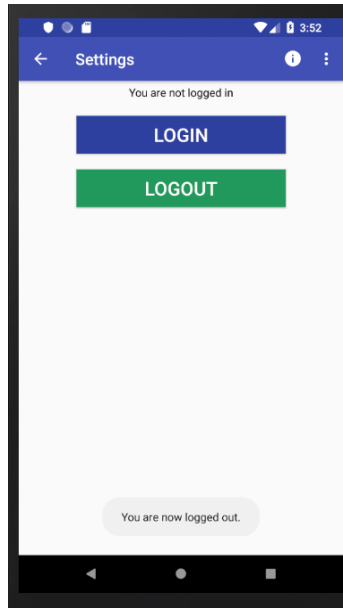


Fig. 17: Settings Screen (Logged out)

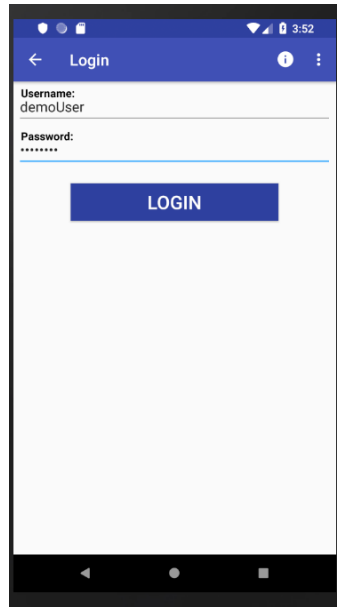


Fig. 18: Log In Screen

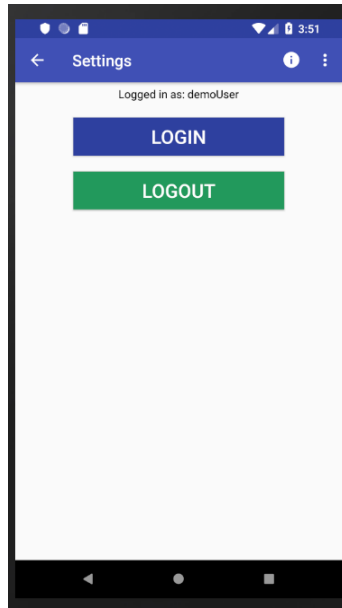


Fig. 19: Settings Screen (Logged In)

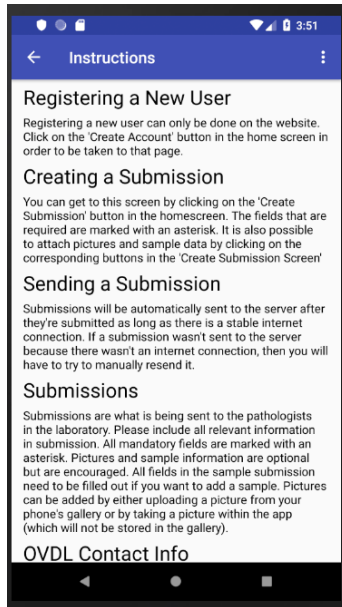


Fig. 20: Instructions Screen

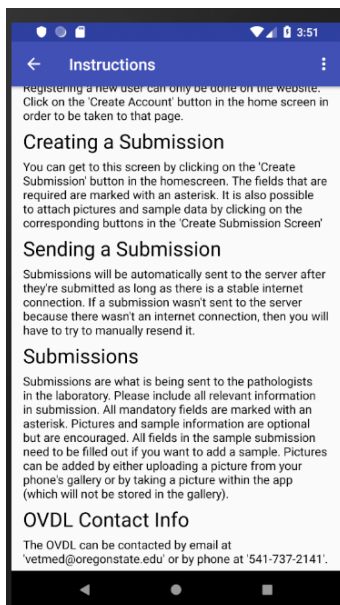


Fig. 21: Instructions Screen (Part 2)

## Oregon Veterinary Diagnostics Laboratory

Portal for the OVDL remote diagnostics app

Email address:

Password:

☐ Remember me

Login

Don't have an account?

Register New Account

Fig. 22: Website Login Screen

## Register New Account

Create a new account to send remote reports to the lab.

Email address:

Password:

Confirm Password:

Position:

Register

Fig. 23: Website Registration Screen