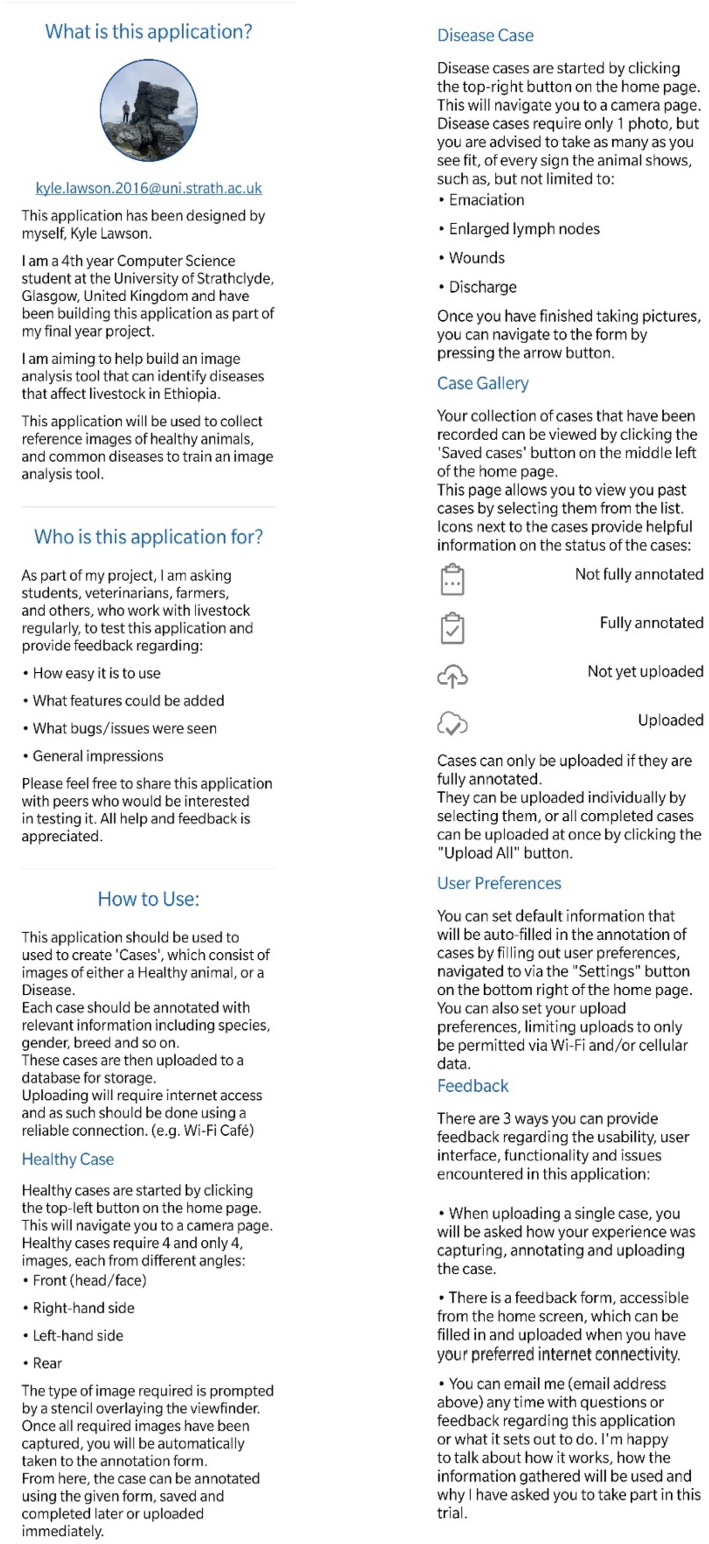
USER, MAINTENANCE & DEVELOPMENT GUIDE FOR THE IMAGE COLLECTION & ANNOTATION ANDROID APPLICATION

USER GUIDE



What is this application?

kyle.lawson.2016@uni.strath.ac.uk

This application has been designed by myself, Kyle Lawson.

I am a 4th year Computer Science student at the University of Strathclyde, Glasgow, United Kingdom and have been building this application as part of my final year project.

I am aiming to help build an image analysis tool that can identify diseases that affect livestock in Ethiopia.

This application will be used to collect reference images of healthy animals, and common diseases to train an image analysis tool.

Who is this application for?

As part of my project, I am asking students, veterinarians, farmers, and others, who work with livestock regularly, to test this application and provide feedback regarding:

- How easy it is to use
- What features could be added
- What bugs/issues were seen
- General impressions

Please feel free to share this application with peers who would be interested in testing it. All help and feedback is appreciated.

How to Use:

This application should be used to used to create 'Cases', which consist of images of either a Healthy animal, or a Disease.
Each case should be annotated with relevant information including species, gender, breed and so on.
These cases are then uploaded to a database for storage.
Uploading will require internet access and as such should be done using a reliable connection. (e.g. Wi-Fi Café)

Healthy Case

Healthy cases are started by clicking the top-left button on the home page. This will navigate you to a camera page. Healthy cases require 4 and only 4, images, each from different angles:

- Front (head/face)
- Right-hand side
- Left-hand side
- Rear

The type of image required is prompted by a stencil overlaying the viewfinder. Once all required images have been captured, you will be automatically taken to the annotation form.
From here, the case can be annotated using the given form, saved and completed later or uploaded immediately.

Disease Case

Disease cases are started by clicking the top-right button on the home page. This will navigate you to a camera page. Disease cases require only 1 photo, but you are advised to take as many as you see fit, of every sign the animal shows, such as, but not limited to:

- Emaciation
- Enlarged lymph nodes
- Wounds
- Discharge

Once you have finished taking pictures, you can navigate to the form by pressing the arrow button.

Case Gallery

Your collection of cases that have been recorded can be viewed by clicking the 'Saved cases' button on the middle left of the home page.
This page allows you to view you past cases by selecting them from the list. Icons next to the cases provide helpful information on the status of the cases:

⋯	Not fully annotated
☑	Fully annotated
☁	Not yet uploaded
✓	Uploaded

Cases can only be uploaded if they are fully annotated.
They can be uploaded individually by selecting them, or all completed cases can be uploaded at once by clicking the "Upload All" button.

User Preferences

You can set default information that will be auto-filled in the annotation of cases by filling out user preferences, navigated to via the "Settings" button on the bottom right of the home page.
You can also set your upload preferences, limiting uploads to only be permitted via Wi-Fi and/or cellular data.

Feedback

There are 3 ways you can provide feedback regarding the usability, user interface, functionality and issues encountered in this application:

- When uploading a single case, you will be asked how your experience was capturing, annotating and uploading the case.
- There is a feedback form, accessible from the home screen, which can be filled in and uploaded when you have your preferred internet connectivity.
- You can email me (email address above) any time with questions or feedback regarding this application or what it sets out to do. I'm happy to talk about how it works, how the information gathered will be used and why I have asked you to take part in this trial.

MAINTENANCE GUIDE

This guide provides instructions for distributing the application and providing updates to users. It also details any potential issues that could arise from having multiple versions of the application installed on a device.

The final application is included in the repository that is detailed in Appendix D, named add-image-collection-4509db9f0bfa460791ad0f0510d1621e-signed.apk. This application can be installed to mobile Android devices running Android Marshmallow 6.0 in two ways.

Device managed installation

1. On an Android device, download the APK from the project repository using the credentials in Appendix D, dismissing any warnings regarding security for downloading a 3rd Party application.
2. Navigate to the device's file manager or downloads manager application.
3. From the manager, navigate to the download folder of the device, or browser used to download the application.
4. Select the downloaded APK and choose to install the APK when asked. Dismiss any warning about installed a 3rd party or unsigned application.
5. Once the installation has completed, it can be found in the application drawer of the device, identified by the icon shown in Figure C1.

Streamed Install using the Android Debug Bridge

1. On a Windows/Mac/Linux machine, download install the Android SDK Platform Tools package from the Android Debug Documentation. This will install the Android Debug Bridge.
2. Connect a device to the machine via a data cable (some cheap charging cables do not allow data transfer!).
3. USB debugging may need to be enabled on the device, which requires enabling developer options. This link provides instructions for enabling developer options on each Android version, follow the instructions relevant to the device.
4. Navigate to developer options via the settings application of the device.
5. Enable USB Debugging.
6. Keep the device unlocked. On a Windows/Mac/Linux machine, download the APK from the project repository using the credentials in Appendix D.

7. Open a terminal/command prompt on the machine and enter Command 1 to verify the device is connected. A device should be listed as a response in the terminal (Terminal Response 1).
8. If the response was not correct, repeat steps 2-7, and ensure the machine is running the terminal/command prompt in administrator mode.
9. Enter Command 2 in the terminal/command prompt to perform the streamed install of the application.
10. Once the installation has completed, the application can be found in the application drawer of the device, identified by the icon shown in Figure C1.
11. At any point between steps 2-10, the device may ask for you to allow files to be transferred/USB debugging/etc. Allow this as it is required for performing a streamed install.

When distributing a new version of the APK, saved information in the application storage and images in the device's photo album may conflict and/or cause bugs in the application. It is advisable to completely wipe the device of all data associated with a previous application version.

Installing a New Version of the Application

1. On the Android device, navigate to Settings > Apps & Notifications > ADD Image Collection App > Storage & Cache. The application settings may be found along a different path in the device settings depending on the Android version.
2. Select Clear Storage & Clear Cache.
3. The application can now be uninstalled by holding down on the application in the app drawer of the device and selecting uninstall.
4. Navigate to the photo album/gallery application on your device. Delete the folder containing application relevant images called Animal Disease Diagnosis.
5. The application is now fully installed, and the new version can be installed in either of the two ways documented in this guide.

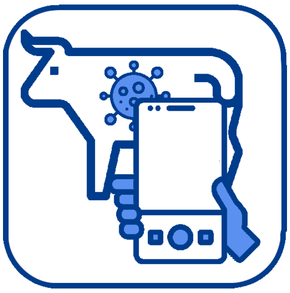


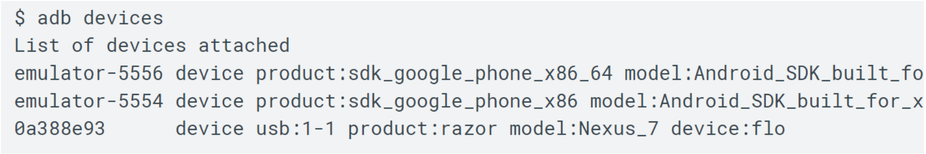
Figure C1 - Application Icon

adb devices -l

Command C1 - ADB Command to List Connected Devices

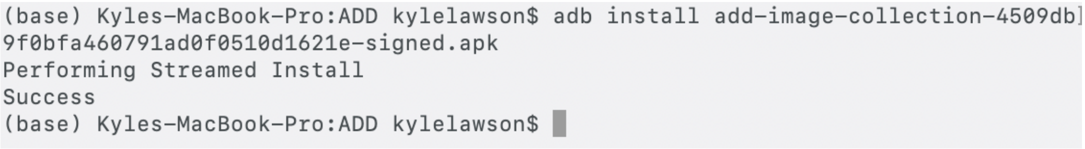
adb install <path-to-apk>

Command C2 - ADB to Perform Streamed APK Install



```
$ adb devices
List of devices attached
emulator-5556 device product:sdk_google_phone_x86_64 model:Android_SDK_built_fo
emulator-5554 device product:sdk_google_phone_x86 model:Android_SDK_built_for_x
0a388e93      device usb:1-1 product:razor model:Nexus_7 device:flo
```

Terminal Response C1 - Typical Response to Command 1, Listing all Connected Android Devices



```
(base) Kyles-MacBook-Pro:ADD kylelawson$ adb install add-image-collection-4509db
9f0bfa460791ad0f0510d1621e-signed.apk
Performing Streamed Install
Success
(base) Kyles-MacBook-Pro:ADD kylelawson$ █
```

Terminal Response C2 - Successful Response when Performing a Streamed Install

DEVELOPMENT GUIDE

This guide details how to run the development version of the application using React Native and Expo. These frameworks are built around JavaScript and involve coding to one maintainable codebase that is compiled down to native code if necessary. In the case of this project, the JavaScript is immediately compiled to a native APK, without having to work with Android Java. Libraries and modules that have been modified are also detailed in this section.

Setting up the Expo Project for Development

1. This section of the guide assumes you are running a Unix machine. More specifically Linux or macOS, the machine must have a BASH terminal, and that you have sudo permissions.
2. Verify that Node Package Manager installed on the machine by entering Command 3 in a terminal instance. The npm version should print if it is installed. Follow this link to install npm if it is not installed.
3. Create a new Project directory on the machine e.g. ~/ADD/
4. Open a terminal in this folder, and clone the project from the repository using Command 4, and Reporter credentials for the repository detailed in Appendix D.
5. Change directory to the project folder using Command 5.
6. Install the Expo Command Line Interface using Command 6.
7. Download the dependencies for the project by using Command 7. This may take a while.
8. Once downloaded, copy the module folder(s) from inside the folder ModifiedDependencies, into the node_modules folder, overwriting any changes.
9. Run the Expo client using Command 8. The expo instance should start in the terminal and in a web browser (Response 3).
10. Download the Expo application on an Android device from the Google play store.
11. Launch the application, you may have to create an expo account.
12. Choose to Scan the QR Code on the application and scan the QR code shown in the web browser or terminal. This will load the JavaScript development version of the application.
13. This development version will refresh when changes occur in the project directory. This allows the application to be previewed during development.

Executing Jest Unit Tests

1. The Jest tests are packaged with the project, contained in the folder 'tests'. These tests can be carried out once Command 7 has been run and all dependencies have been downloaded. The Jest testing suite is an included dependency.
2. The Jest tests included in the project validate the code that is independent of device functionality and React native Component States and Properties.
3. Ensure 'Setting up the Expo Project for Development' has been fully completed before running the Jest unit tests.
4. Open a terminal instance on the project folder.
5. Run Command 9 to begin the Jest Unit tests.
6. The test will run over the next few seconds and display the results of each test.

MODULE FIXES

The Component Swiper used from the react-native-swiper package contained bugs as of 01/04/20 and required fixes to be implemented. The fixed module is found in the folder ModifiedDependencies in the project directory and is added to the project in Step 7 above.

The bug in the module is highlighted in Figure C2. This code does not correctly handle the indexing of the image currently displaying the Swiper, if more than one is present, causing it to display the incorrect image, and not change the index indicator on the Swiper. Figure C3 outlines the fix that is implemented, where the index correctly updates on swipes. Until this module is updated (as of 01/04/20 it is not), this fixed module must be included in the project and manually added to the node_modules directory when the project is cloned.

EXPO VERSION

As of 05/04/20, building the project with Expo 37 will result in the View of the application not rendering correctly. When a keyboard appears, the View is treated as if it were wrapped in a KeyboardAvoidingView, moving up to avoid the keyboard, breaking the styling of the View. The developers of Expo are aware of the issue and are working on a fix.

Issue 7589 Issue 7666 Issue 6506 Issue 7635

The APK included in the project repository was built with Expo 36, and as such does not have this issue.

```
npm -v
```

Command C3 – Command to Verify npm Installation

```
clone https://gitlab.cis.strath.ac.uk/xsb16116/animal-disease-diagnosis-app.git
```

Command C4 – Clone Project Repository

```
cd animal-disease-diagnosis-app/ADD_37
```

Command C5 – Change to Project Directory

```
npm i -g expo-cli
```

Command C6 – Globally Install the Expo Command Line Interface

```
npm install
```

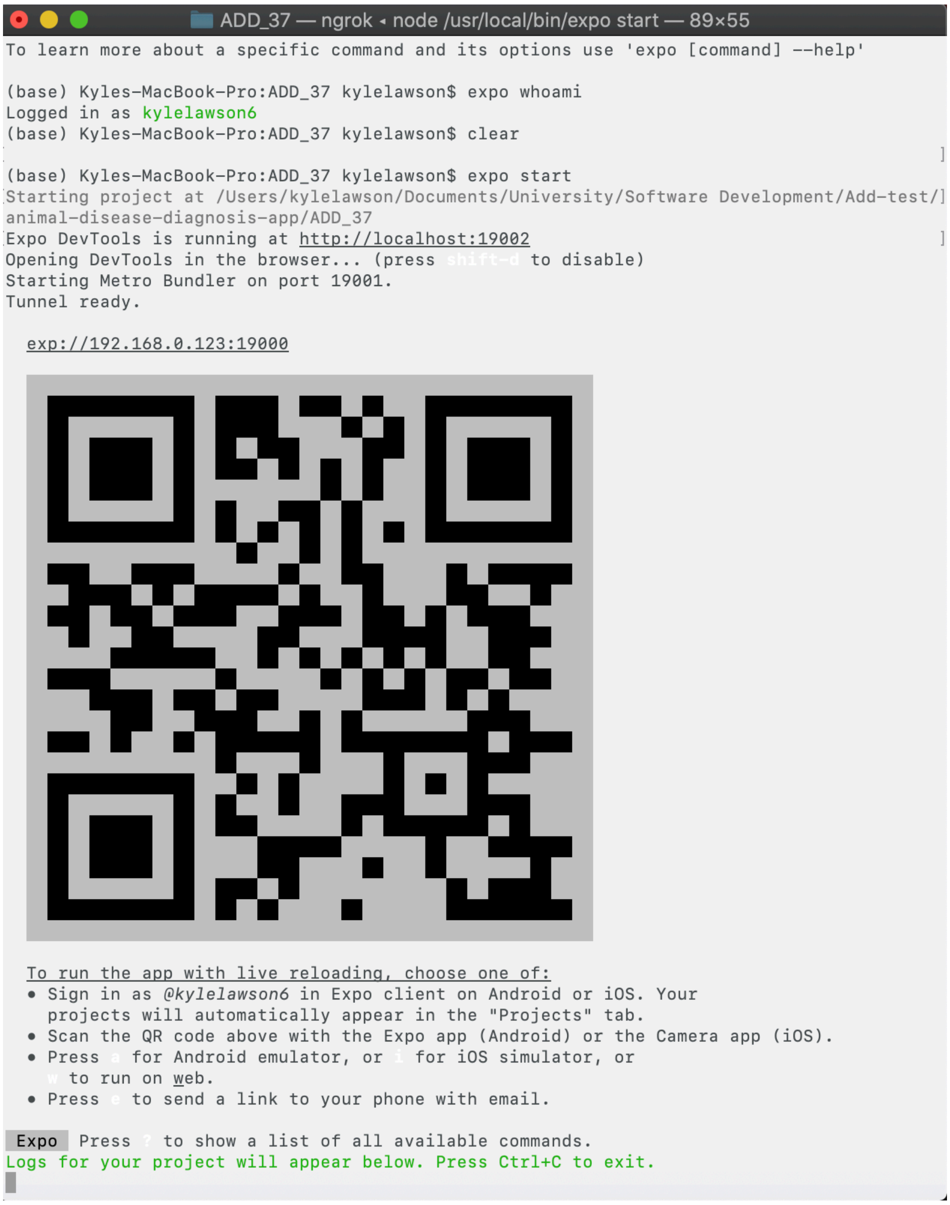
Command C7 – Download Project Dependencies

```
expo start
```

Command C8 – Start an Instance of the Expo Development Environment

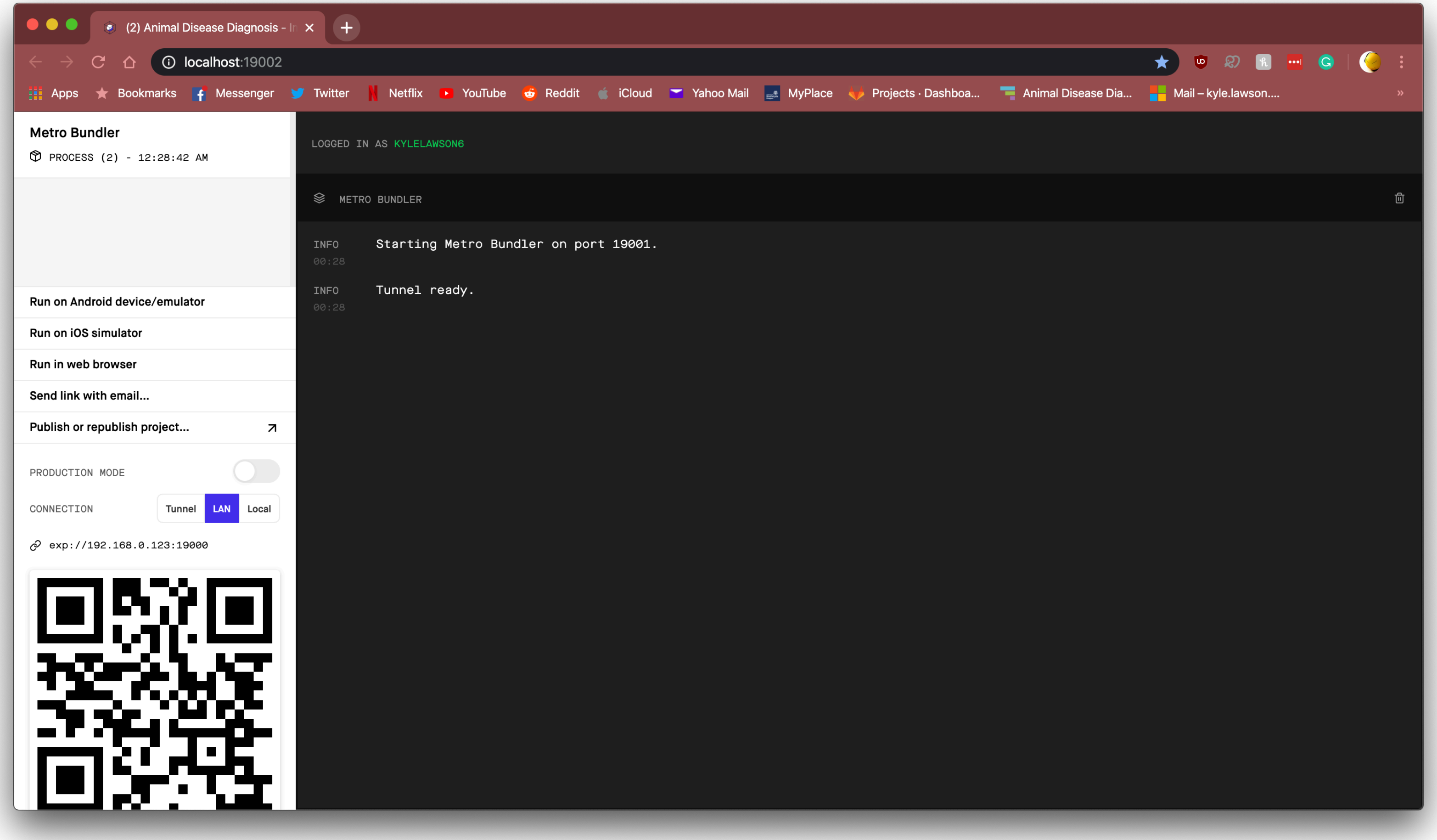
```
npm test
```

Command C9 – Command to Run Jest Unit Tests



```
To learn more about a specific command and its options use 'expo [command] --help'

(base) Kyles-MacBook-Pro:ADD_37 kylelawson$ expo whoami
Logged in as kylelawson6
(base) Kyles-MacBook-Pro:ADD_37 kylelawson$ clear
                                                                                        ]
(base) Kyles-MacBook-Pro:ADD_37 kylelawson$ expo start
Starting project at /Users/kylelawson/Documents/University/Software Development/Add-test/]
animal-disease-diagnosis-app/ADD_37
Expo DevTools is running at http://localhost:19002                                      ]
Opening DevTools in the browser... (press shift-d to disable)
Starting Metro Bundler on port 19001.
Tunnel ready.

  exp://192.168.0.123:19000



  To run the app with live reloading, choose one of:
  • Sign in as @kylelawson6 in Expo client on Android or iOS. Your
    projects will automatically appear in the "Projects" tab.
  • Scan the QR code above with the Expo app (Android) or the Camera app (iOS).
  • Press a for Android emulator, or i for iOS simulator, or
    w to run on web.
  • Press e to send a link to your phone with email.

 Expo  Press ? to show a list of all available commands.
Logs for your project will appear below. Press Ctrl+C to exit.
```

Terminal Response C3 - Expo Client in the Terminal



Metro Bundler

PROCESS (2) - 12:28:42 AM

PRODUCTION MODE

CONNECTION: Tunnel [✓ LAN] Local

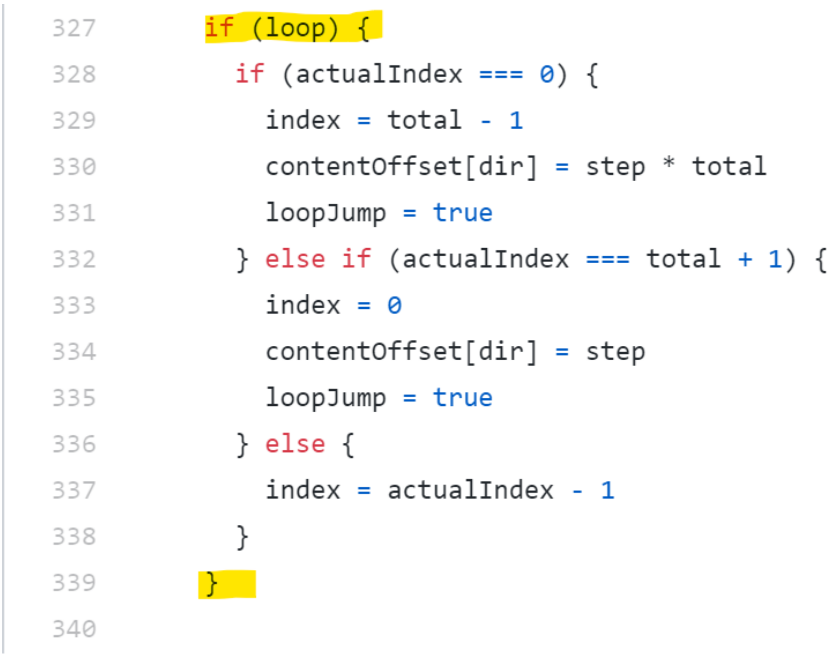
exp://192.168.0.123:19000

LOGGED IN AS KYLELAWSON6

METRO BUNDLER

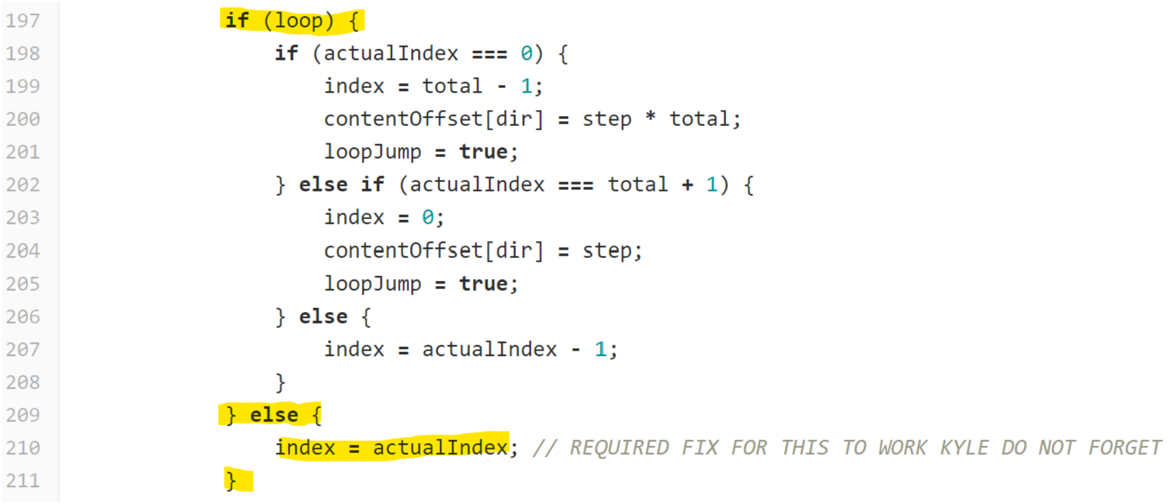
```
INFO    Starting Metro Bundler on port 19001.
00:28

INFO    Tunnel ready.
00:28
```

Terminal Response C4 - Expo Client in the Web Browser



```
if (loop) {
  if (actualIndex === 0) {
    index = total - 1
    contentOffset[dir] = step * total
    loopJump = true
  } else if (actualIndex === total + 1) {
    index = 0
    contentOffset[dir] = step
    loopJump = true
  } else {
    index = actualIndex - 1
  }
}

```

Figure C2 - Unhandled Condition in Swiper Module



```
if (loop) {
    if (actualIndex === 0) {
        index = total - 1;
        contentOffset[dir] = step * total;
        loopJump = true;
    } else if (actualIndex === total + 1) {
        index = 0;
        contentOffset[dir] = step;
        loopJump = true;
    } else {
        index = actualIndex - 1;
    }
} else {
    index = actualIndex; // REQUIRED FIX FOR THIS TO WORK KYLE DO NOT FORGET
}
```

Figure C3 - Implemented Fix for Swiper Module

APPENDIX D (SOURCE CODE AND DELIVERABLES)

The APK and source code for the application developed in this project is hosted on a University of Strathclyde Gitlab repository. Additionally, the Jest tests and PHP scripts are included in this repository. These resources can be cloned run as outlined in Appendix C – User, Maintenance and Development Guides. The following are the credentials to access a Gitlab account with Reporter permissions for this project so all code can be downloaded and viewed, but not modified.

Username: kylelawson

Password: eek6aQueing1

Repository: https://gitlab.cis.strath.ac.uk/xsb16116/animal-disease-diagnosis-app

The database is hosted on the University of Strathclyde server space DEVWEB and can be maintained and inspected through the phpMyAdmin dashboard. The following credentials provide access to this service where data collected from the application is stored.

Username: xsb16116

Password: kiaf1Eifoo2e

phpMyAdmin: https://devweb2019.cis.strath.ac.uk/phpmyadmin/