# **Android Application**

## Introduction

The purpose of the android application is to replicate all the functionality of the PieChecker website. The application is designed to function on smart phones running the android operating system and the architecture is designed in such a way that the GUI will be adaptable to tablet size screens.

All testing was carried out using a Samsung Galaxy S4 running Android version 4.4.2 (Kitkat). All code was created using Eclipse with the Android SDK.

## GUI Architecture

The UI framework uses a Model-View-Controller (MVC) pattern. The view is responsible for what is seen on the screen. The controller is responsible for reacting to external events such as keystrokes, screen taps or phone calls. The model is the core of the application. The controller updates the model, the model invalidates the view and causes it to be redrawn if necessary.

As the application should run on smart phones and tablets it utilises fragments within the view.

|  |
| --- |
| Activity |

|  |
| --- |
| Fragment |

The activity represents the view of the screen and has no components. The activity is needed to hold the fragment. Each view of the GUI has its own fragment and they are swapped as needed. For example the home and video pages on the website are reproduced in two separate fragments for the android app.

## Splash Screen

The splash screen is the first thing the user sees when starting the app. It is run in its own thread to allow other processes to be performed at the same time. The original intention was to allow the app to retrieve any necessary data at this point.

## Login Screen

The login screen in its current state was designed to work with a version of the website no longer in use. The code takes the entered email address and posts it to the URL. This would trigger the account system to send an email to the user with instructions on registering.

## Home Screen

The home screen demonstrates the use of a list view containing more than one object. The images shown are held in the resources of the app. In a real app the images would be held in a remote database. To improve performance of the app the images would be fetched once, scaled to an appropriate size and held in a database on the smart phone.

## Gallery Screen

The gallery screen is used to demonstrate how multiple images can be shown on a small device. The view can be scrolled if there are more images than will fit on the screen. As with the home page the images are held on the device but would normally be on a remote database. The app would need to be modified to perform scaling on any downloaded images before attempting to display them.

## My Video Screen

The video screen use a video view in the top half of the screen and place holders for the temperature, timer and a button that would be used for changing the device being viewed. When the video view is pressed a controller appears in the bottom of the screen. The video in the images is a sample m3u8 stream from Apple. For the first demonstration the app was shown playing an mp4 file that was stored locally on the phones SD card.

## Navigation

The user navigates through the app using a drawer layout with a list. When the user selects an item in the list the app displays the appropriate fragment. The menu is displayed by either swiping from the left across the screen or pressing the arrow key in the top left of the image.

## Streaming from Twitch TV to Android

Streaming from Twitch is buggy at best and unworkable at worst. Twitch uses SWF file format which is only playable using Flash player. Flash is no longer supported by Android and Adobe no longer have support to allow Flash to play on Android. Twitch also uses HLS, or HTTP Live Stream, for most of there streams. While HLS is technically supported on Android it is extremely limited. It should, in theory, be possible to show a HLS stream in a web view. When this was attempted the image below would be shown.

The app was also testing showing the whole page in a web view.

The image above shows that the page in a web view appears to work. The audio started at 0 seconds but the video does not proceed beyond the initial frame. The image below shows the same stream at a later time. The video has still not changed.

It should be noted that at no point did the stream play any advertisements and when the app did not have focus the audio continued to play.

## Twitch TV App

Twitch TV has its own app that works on Android phones. It is currently free to use and allows the user to login and watch streams.

## Streaming from Android to Twitch

Given the lack of experience with Android and streaming in general this was always going to be a difficult task. Twitch TV is in the process of creating an SDK for mobile users but this has yet to cover Android devices. There also exist several apps for streaming across a Wi-Fi network. The most commonly mentioned one, SpyDroid, was tried but it failed to stream any video but this is more likely to be bug related to hardware or using the latest Android version. The end result is that no progress was made here. The only real solution would be to create something completely from scratch and this is far beyond the current capabilities of the team member assigned to this.

## Conclusion

The biggest obstacle to overcome for this part of the project was the lack of experience with Android development. At least 50% of the time spent was on research and tutorials to understand the basics. Late changes to the back end and no database of images have blocked development of correctly implementing the getting and processing of images for display. The change to Twitch as a streaming service has made it difficult to stream to the app without the Twitch app being installed, even with the app the performance is extremely buggy when attempting to watch in a web view. Streaming from a smart phone was also not achieved. At the time of writing there is still no SDK or API for Android from Twitch and despite several days effort no solution was found.

In conclusion, streaming to and from Android is the very first thing that should have been researched. It is extremely unlikely that the app would have been started had the group understood just how little information there was available and how little support there was for Android streaming. On the positive side there was significant knowledge gained while learning about Android development that can be utilised in future projects.