**CS 4153**
Mobile App Development
April 27th, 2021

# Quick LAS Viewer (QuickLAS)

Final Report

Roden, Jeff - A10015856
Roberson, Matthew - A20215085
Stott, Lucas - A11518481
Yeager, Jessey - A11773915
**Submitted By**: Jeff Roden

# Introduction

In the oil-and-gas industry,and even in the water well industry, the standard format for storing well log information is within a Long ASCII Standard file (an LAS file). Each LAS file is specific to a single well and is assigned a unique well identifier (UWI) number. Each LAS file has many pieces of information measured by various tools inserted down into the well which make contact with the borehole and are then slowly drug up the hole and back to the surface. The tools record measurements of the natural gamma ray, acoustic travel time (inferring rock density), resistivity and so on. The stored measurements of these various tools are referred to in the industry as “curves”. These curves are commonly graphed and used by those working in the industry to help understand how and where to drill. There are several desktop analysis programs out there that will take this data and calculate statistical and other useful information for display. However, an experienced worker need only look at the curves themselves to be able to interpret much of this information.

Our application, QuickLAS, is designed to do just that. The app reads in an LAS file and will display up to three of the different curves contained in the file in a “track” or graph. Displaying a maximum of two tracks/graphs at once. Using these curves, the user can make quick and easy decisions on what geologic formations a well penetrates and evaluate them for oil and gas bearing zones.

Many applications like this exist already, but few if any are as simple and efficient, and even fewer are accessible through a mobile device. This is what makes QuickLAS unique and useful: its simple efficiency. While computer programs require one to carry around a laptop or otherwise be tethered to a desk, it is rare that anyone, especially one in the field, will not be carrying a smartphone. Thus, this application allows quick and easy access to information that might have otherwise taken longer to obtain, slowing production and, ultimately, efficiency.

# Final Design/Components and Functions

See “Final Design and Components.pdf” in the project zip file.

# App Step-by-Step Instructions

All Testing was done on a Pixel 3a, targeting API 29: Android 10.0 (Q).

A Sample LAS file has been supplied with the project named “35073247520000.las”.

**This file will need to be copied into the emulator’s sdcard folder prior to launching the app!**

This is accomplished by navigating to the “Device File Explorer” inside Android Studio located at the lower right hand edge of the Android IDE shown in **Figure 1**.

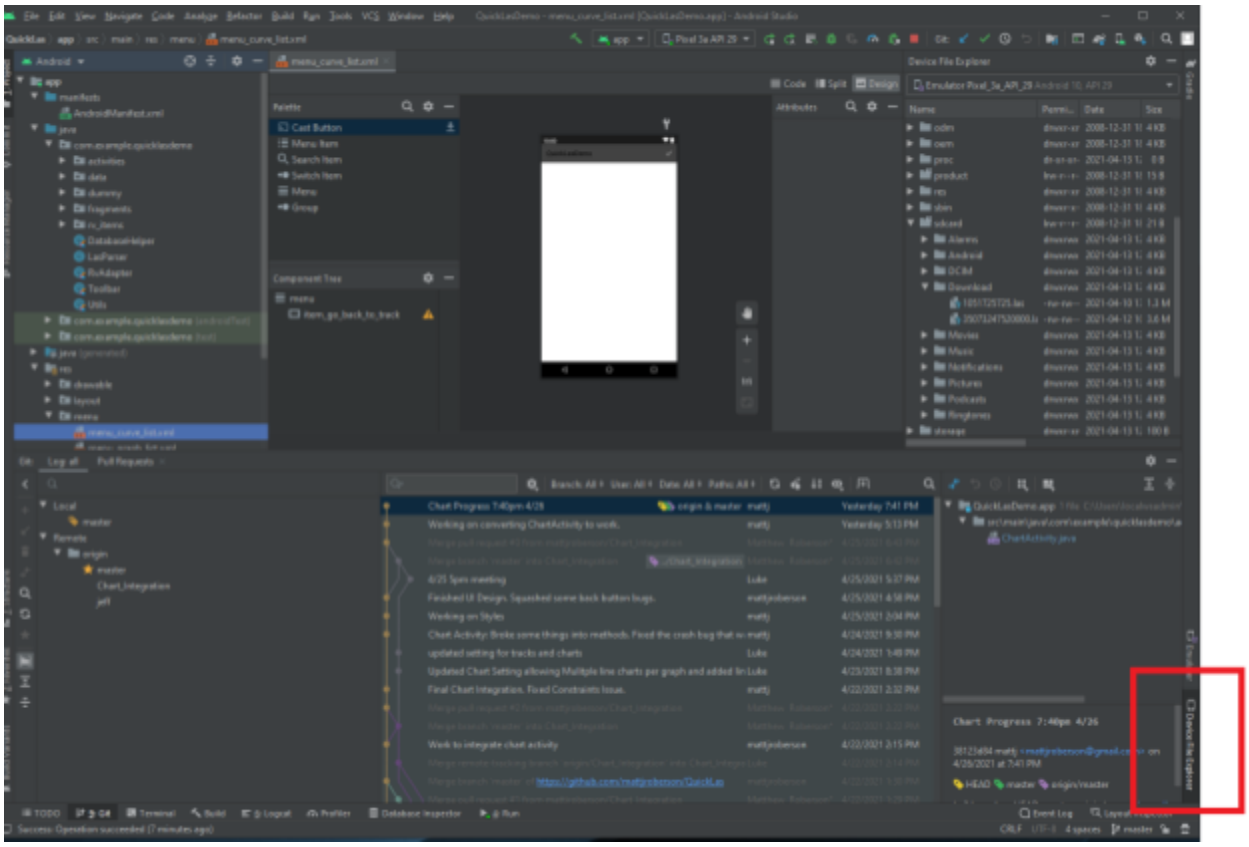


**Figure 1**

From here right click on the *sdcard/Download* folder and select “Upload” and navigate to the LAS file and hit ok. The device explorer should now look like **Figure 2**.

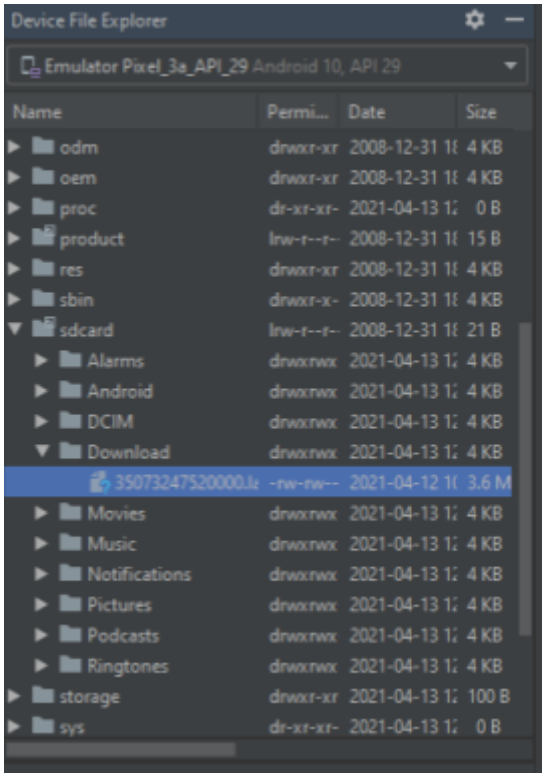



**Figure 2**

The app can now be started in the Pixel 3a emulator.

1. After starting the app, press the “Tap to Begin” button to move to the next screen.

2. You will now import the data from the previously mentioned LAS file into the app. Press the “+” button in the upper right hand corner of the screen as shown below in **Figure 3**.

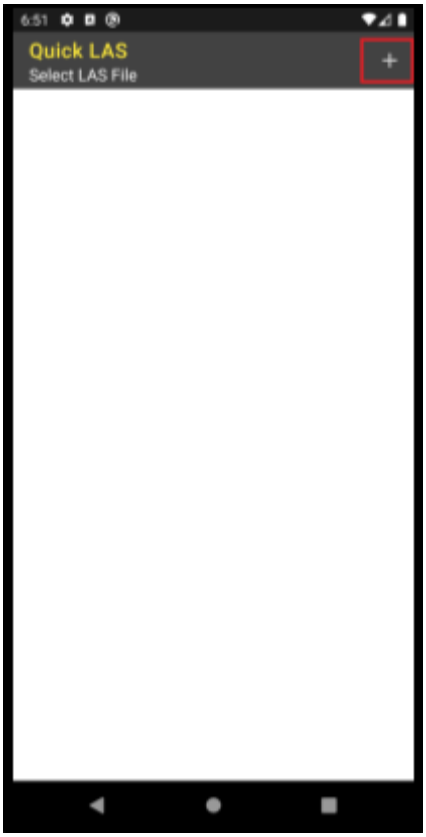



**Figure 3**

3. Select the navigation “stack” button in the upper left then select “Android SDK built for x86” and navigate to the Downloads folder. Double click the LAS file as shown in **Figure 4**.

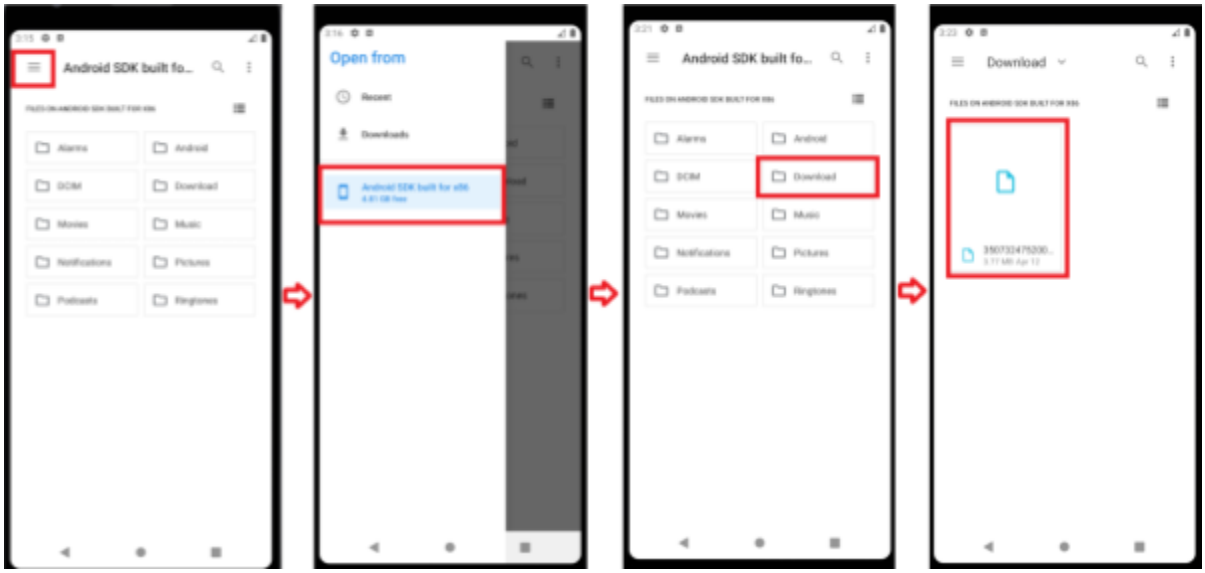



**Figure 4**

4. The well data has now been loaded into the app for this particular well! Subsequent wells could be imported following this same process and are saved into the app’s database for future viewing. There are two buttons located to the right of the well’s Unique Well Identifier (UWI) number. A trashcan button to delete the well from the database, and a pencil to access and set up the well’s data for graphing.

5. Select the Pencil button to the right of the well’s UWI number to advance to the next screen.

6. You are now at the “track” setup screen, where you will tell the app how many “tracks” or graphs to display at once. Due to the limited real estate available on a mobile screen, we have set a maximum of two tracks for simultaneous viewing.

7. In the upper right hand section of the screen press the “+” button to add a track.

8. You should now have a screen similar to **Figure 5**.

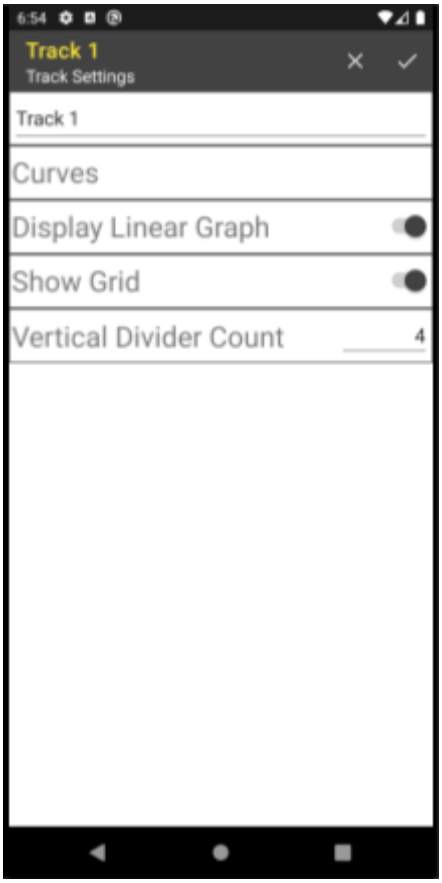



**Figure 5**

9. By pressing on the “Track 1” text, a custom name could be set for the track. If a custom name is used make sure to press the green enter button on the virtual keyboard **(*Not the return key on your physical keyboard or it won’t save*)**.

10. Next, press on “Curves” to select the well’s log “curve” data you want to display for this track. You are shown a list of every tool measurement available for this particular well. This is the data that was imported from the LAS file earlier.

11. For this track we will display two curves, navigate to the curve named “SP” and press the empty box to the left of its name to select the data for display. Now press on the pencil icon to the right of “SP” to set up it’s scale and display style/color. Set style to dotted and the color as red. For the Scale Min set it to -400 and set the max to 1. Make sure to press the green enter button on the virtual keyboard for each setting **(*Not the return key on your physical keyboard or it won’t save*)**.Click the check mark in the upper right to move back to the “pick curves” screen.

12. Now Scroll and find the curve named “GR” click the empty box to the left to select this curve and then press on the pencil button to set it’s style and scale values. Leave style as normal and change the line color to blue. Leave the scale min at 0 and change the scale max to 150. Make sure to press the green enter button on the virtual keyboard **(*Not the return key on your physical keyboard or it won’t save*)**. Click the check mark in the upper right hand corner to finish editing the curve’s settings.

13. Click the check mark button in the upper right hand portion of the screen to finish picking curve data for this track as seen in **Figure 6**.

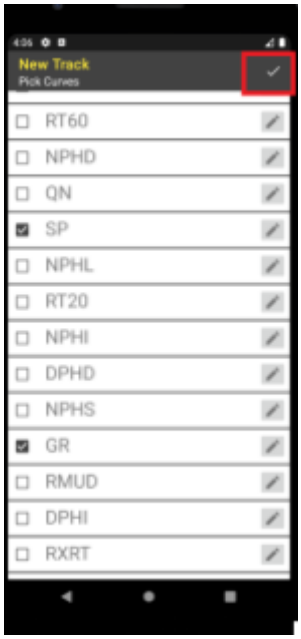



**Figure 6**

14. You are now back at the track settings screen seen before in **Figure 5**. **Make sure the “Display Linear Graph” is toggled ON for this track.** There are a few more settings such as Show Grid and Vertical Divider Count that will affect how the track/graph itself displays, but will be left to their defaults. Press the check mark button in the upper right corner to finish setting up this track.

15. You have now successfully set up two log curves to display in Track 1, the spontaneous potential (SP) and Gamma Ray (GR).

16. We will now add one more track to display resistivity curve data. Press the “+” button in the upper right to add another track as shown in **Figure 7**.

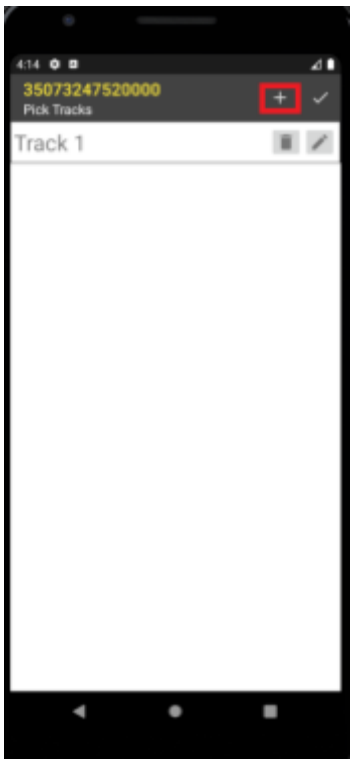



**Figure 7**

17. This is named a default of “Track 2” (remember to press the virtual keyboard enter button, not your physical keyboard if changing) and then press on “Curves” to select data for display.

18. You will select three curves for this track. First select the box next to “RT20” and then press the pencil button. Leave the color as red and set the Scale Min to 0.2 and the Scale Max to 2000. *Remember to use the virtual keyboard enter button for each.* Press the check mark in the upper right portion of the screen to move back to the “Pick Curves” screen.

19. Repeat this process for the curves named “RT30” and “RT90” using the same scale values of 0.2 to 2000 and changing the colors of one to green and the other to blue.

20. You should be back at the “Pick Curves” screen after setting the scales and colors for the three resisitivity curves. Press the check mark button in the upper right corner to finish.

21. You should now be back at the “Track Settings” screen for Track 2. **Toggle “Display Linear Graph” to the OFF position**, as this data is plotted on a logarithmic scale.

22. Finish the setup by once again pressing the checkmark button in the upper right hand corner.

23. You Should now be back at the “Pick Tracks” screen and have a similar view to **Figure 8**. Press the check mark button in the upper right of the screen.

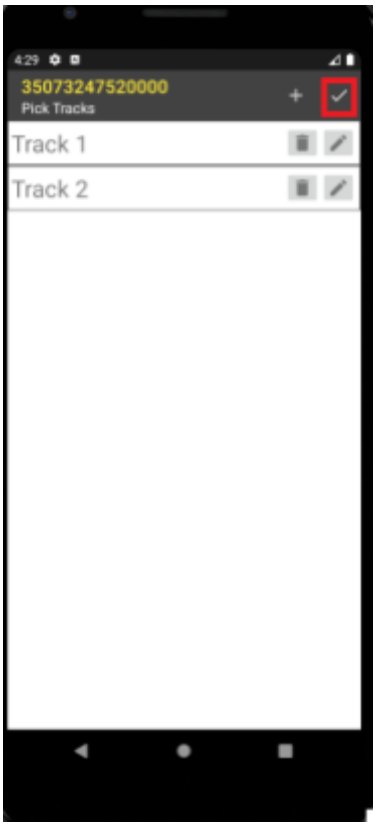



**Figure 8**

24. You are now back at the main “Select LAS File” screen. Press anywhere on the UWI number outlined in red in **Figure 9** to graph the curves.

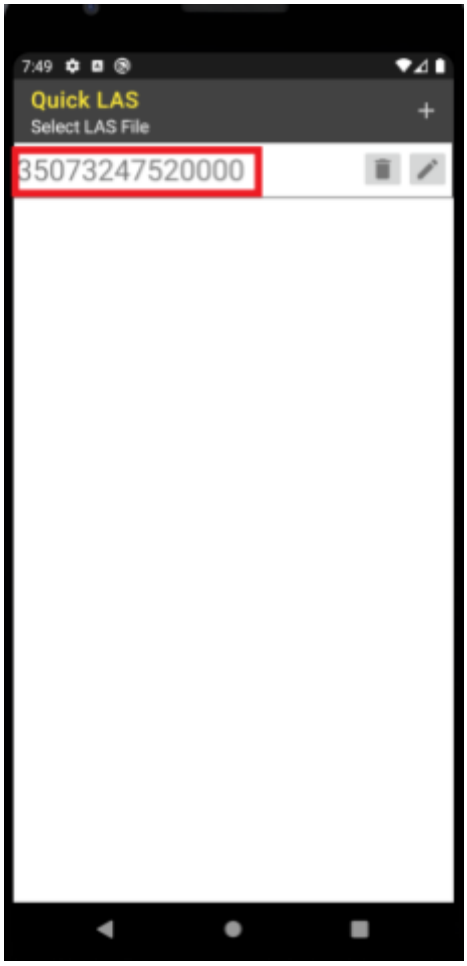



**Figure 9**

25. You should now see the two tracks displaying the curve data you selected and set up as shown in **Figure 10**. You can scroll up and down the wellbore to view additional data by swiping up and down. When finished you can navigate back to the LAS import screen by pressing the back button in the lower left corner of the screen.

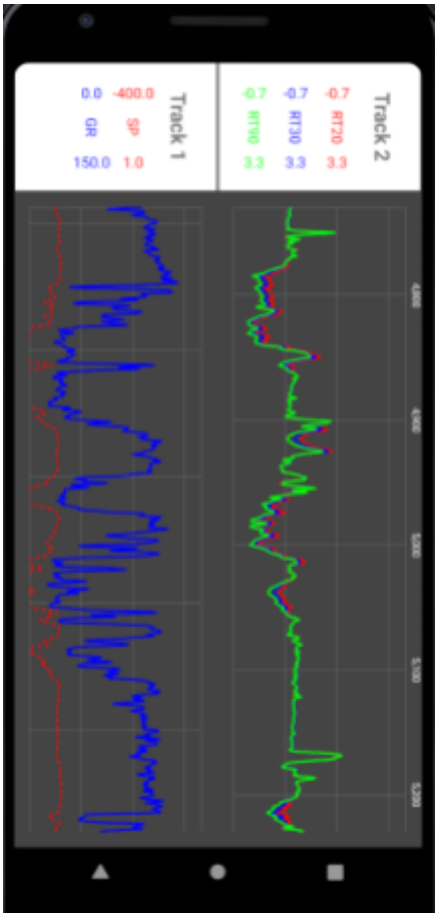



**Figure 10**

# Client Feedback

During the duration of the semester, the team touched base with the client to confirm the progress of the application as well as the expected deliverables. On April 27th, we demonstrated to the client the basics of the application: how to import and set up files for graphing. We showed off the UI, features, what was accomplished, and what we could not get done in time. Overall, the client was satisfied with the functionality and gave positive feedback. The main objective of the project was produced and worked properly. He did, however, express the desire to be able to import geologic horizons, or “tops”, into the application to display on top of the tracks. This is a common practice when viewing these geological graphs and would increase the functionality and efficiency of the application. However, this was outside of the original scope of our project and could not get done in time.

# Team Experience

**Problem**: *Graphing Library*
Needed a prebuilt library to handle the graphing functions inside Android. Building a function to graph would have been a very tedious and long process. Most apps who have a similar function use code from scratch inside the Android libraries. This would have taken months to develop with open source platforms like Github. We took a week or so to look over and test different libraries.

**Solution**: *MPAndroidChart*
After much digging and testing of different libraries for graphing. We chose to use MPAndroidChart since the documentation was accessible and the library was open source. This was the strongest library we could find for Android for graphing functions, though some features were not included that we hardcoded ourselves.

---

**Problem**: *Data Storage*
A solution to storing user data and pulling it to generate the graphs and persistence inside the app.

**Solution**: *SQLite Database*
One of the developers spent a good amount of time developing SQLite inside Android and building the functions to pull, swap, and replace data exactly how we intended. This was not an easy task since gestures like button clicks, input values, and settings needed to be pushed to the DB. The framework of a DB is complex and needs to be able to handle errors and segues internally while listening to the user.

---

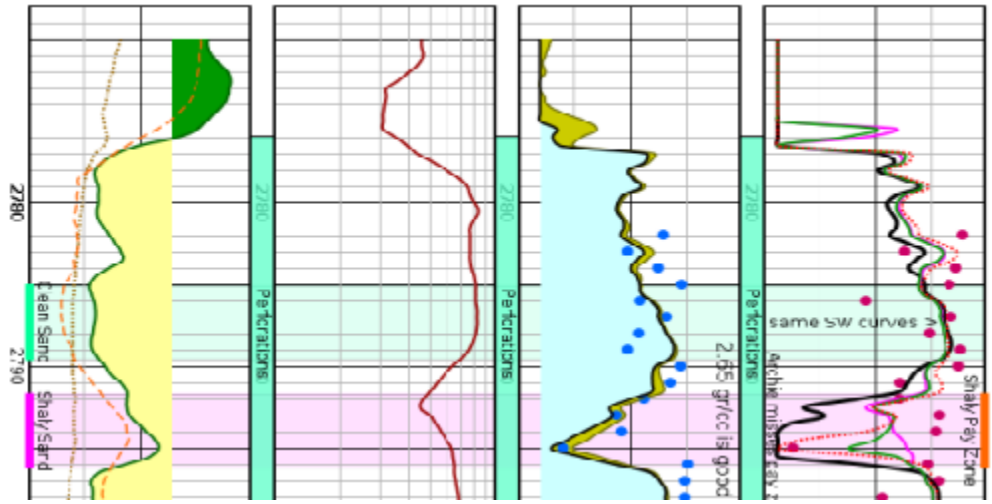
**Problem**: *LAS Parsing*
The LAS file format is stored in a text file but is arranged very differently than other common file types. There is nothing in the Java/Kotlin main libraries that could readily parse it for us. We needed to extract its data by searching through the file and parse out each individual curve's data.

**Solution**: *Custom Java LAS Parser*
Another one of the developers built a class which parsed the LAS data and used SQLite DB to store it within the app. This was a fundamental first step in the project and helped generate some traction on further development.

# Improvements

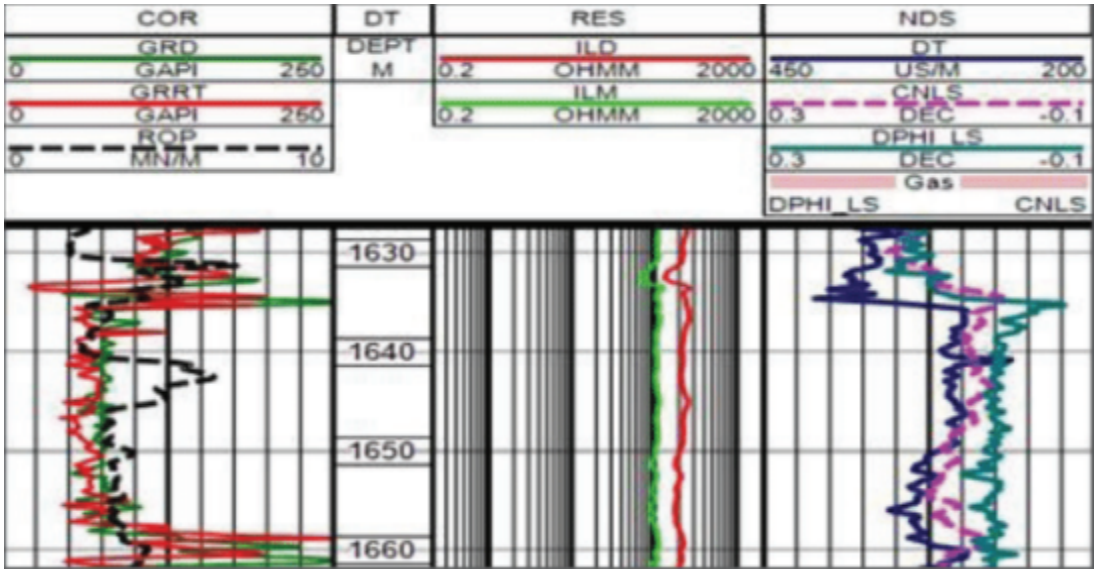
**Drawable Fills:**

The MPAndroidChart library allows gradient and solid fill from a value up to a line. This is done with the drawable's included in Android Studio. Implementing this with settings (fill in from > to) and color allows the user to see paths easier and faster.



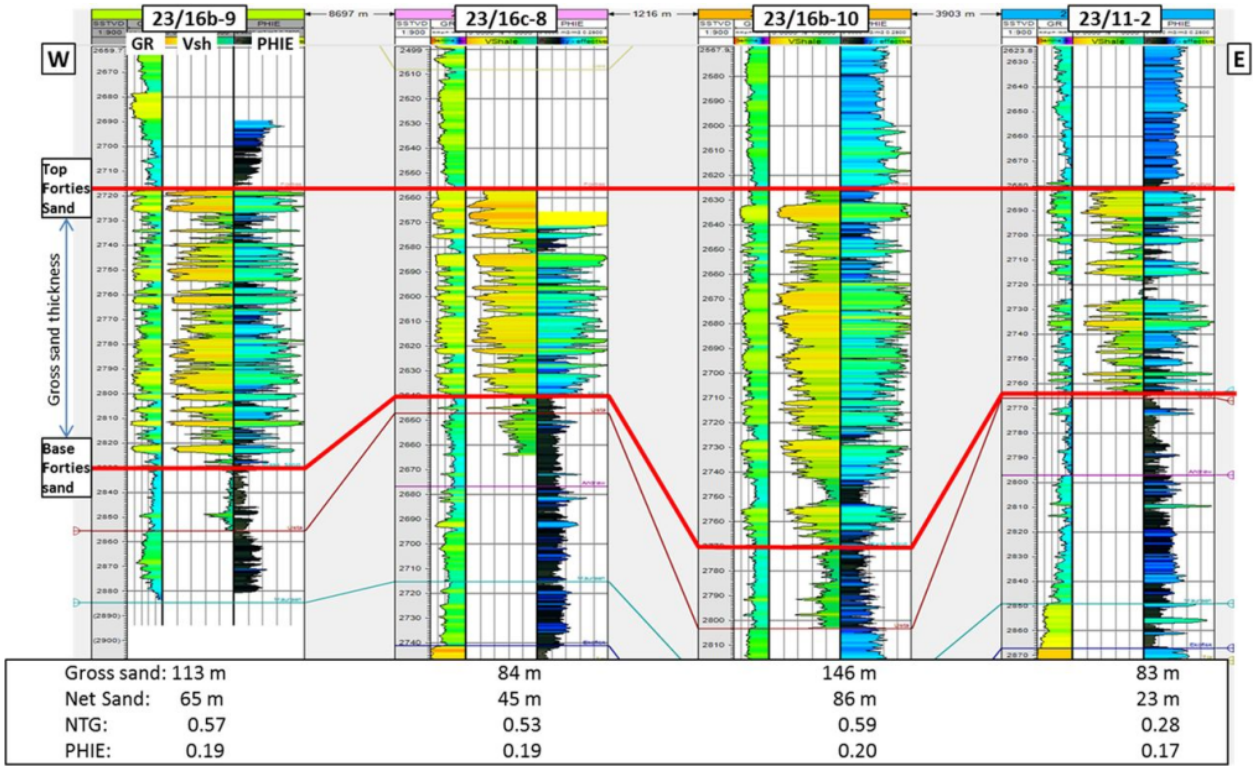

**Graph Headings:**

For a Graph to be complete it needs easily accessible data to view. Implementing this inside the XML layout file and using the resources from the SQLite. To accomplish this, we could have generated a list of Curves and assigned values to the Label's from the parsed data and by settings we already have available. For now the graph layout is usable but not with different curve settings overlapping each other.



**Geologic Horizons or “Tops”:**

Tops are used by the geologist to mark where the formations occur in the subsurface. By inputting this data into the chart, it helps to signify where you are looking downhole and to get oriented, acting similar to “sign posts”. This is extremely helpful as some well log depths can range from a few thousand feet to more than ten thousand feet. Importing this data and using a simple calculation to create a marker would help them find desired sections quicker. The below image demonstrates geologic horizons (bold red lines) or tops marking off sections of the well log.



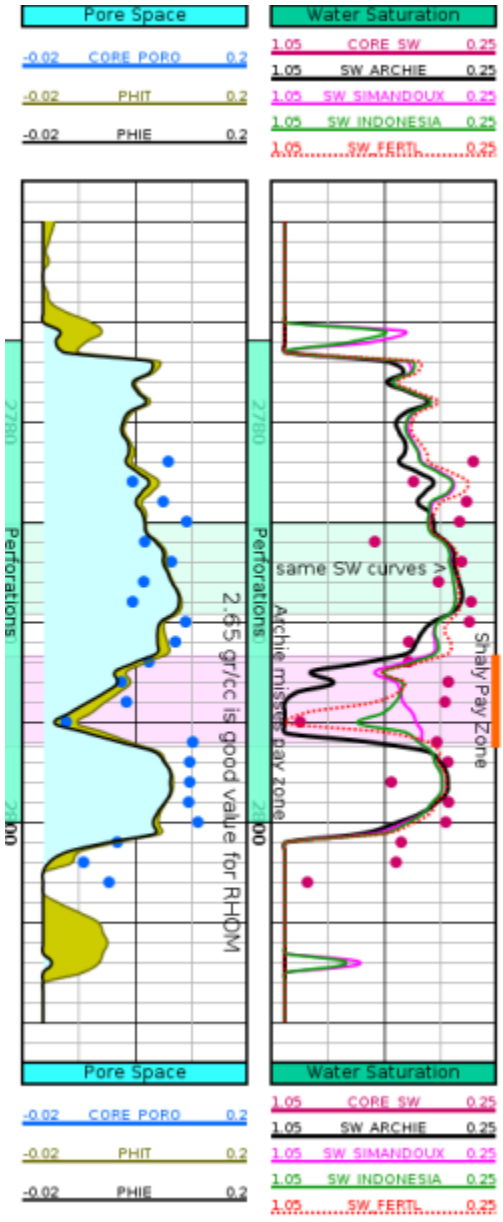

| | 23/16b-9 | 23/16c-8 | 23/16b-10 | 23/11-2 |
|---|---|---|---|---|
| Gross sand: | 113 m | 84 m | 146 m | 83 m |
| Net Sand: | 65 m | 45 m | 86 m | 23 m |
| NTG: | 0.57 | 0.53 | 0.59 | 0.28 |
| PHIE: | 0.19 | 0.19 | 0.20 | 0.17 |

**Water Saturation:**

$$SW_{\text{Archie}} = \left( \frac{a}{\phi^m} \cdot \frac{Rw}{Rt} \right)^{(1/n)}$$

$$SW_{\text{Simandoux}} = \frac{a \,.\, Rw}{2 \,.\, \phi^m} \left[ \sqrt{\left( \frac{Vsh}{Rsh} \right)^2 + \frac{4\,\phi^m}{a \,.\, Rw \,.\, Rt}} - \frac{Vsh}{Rsh} \right]$$

Archie's equation, or less commonly the Simandoux equation, is used to calculate water saturation using the measured values imported from the LAS file - namely the porosity and resistivity values. We could create a new track for this graph containing this data. In the example, the water saturation track shows different computed curves and puts them in the same track. It helps determine oil or gas bearing zones with the highest potential for success. This is great for prospecting in different wells as water saturation is used directly to determine oil saturation.



**Landscape Scroll View:**

One of the easiest ways to incorporate more tracks into the graph using the app would be to incorporate a landscape scroll view. We did not have time to fix all the minor bugs and optimize the code. Once optimized, landscape would allow for more tracks and data to be outputted from graphs in a legible manner. The graphs may look simple but the hardware is using a lot of gestures to generate the data..


**Better Graphing:**

Many of the desired improvements would be difficult to implement with the different libraries out there, as the functionality is very unique and specific. If we had more time, we wanted to design our own graphing system using a library such as Core Graphics to draw things exactly as we desired. This would allow for incredible flexibility and creativity in displaying the data, while eliminating the difficulties encountered by working with other people's graphing libraries.