React-native charts

Features of Required chart
  1. Horizontal Scrollable view
  2. Slider and Only last data point must be movable along with it
  3. Zoom in and out feature
  4. Scaling the graph according to the dimensions of the device

The image below represents the required chart

The following charting library were used to implement the above requirements

CaptialOne charting library

This library is a cross-platform (iOS/Android) library of charts/graphs using react-native-svg and paths-js based on the excellent work done by Roman Samec in the react-pathjs-chart library.

Features Implemented
1.Horizontal Scroll view
2.Slider
3.Scalling of the graph according to the device dimensions

Issues faced and how it was resolved
1.The library provided basic charts, hence had to make few changes in the library to get the required features.
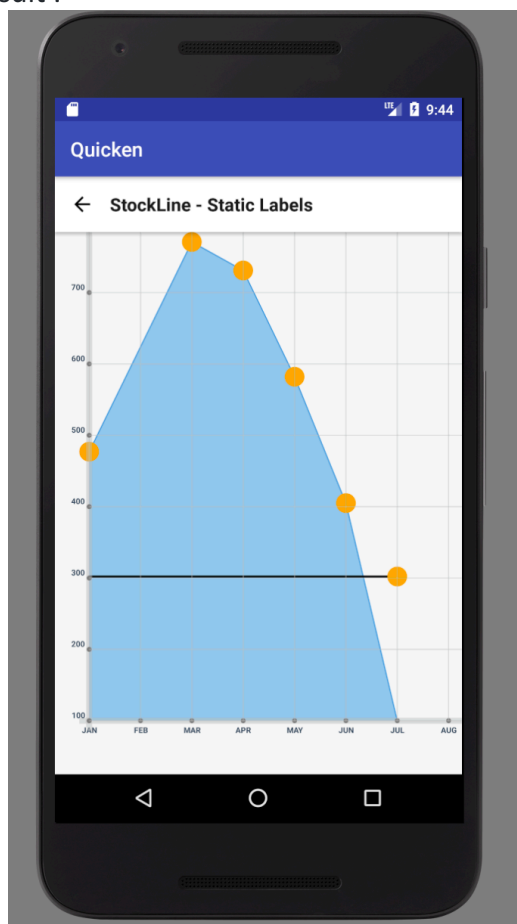a. Added circles for the data points.
b. Scale the graph

c. Added conditions to make only the last data point on the graph movable along with the horizontal line(slider) using pan responder and made sure the slider could move only along y-axis.

2. Maximum offset for horizontal scroll view had to be device dependent, which was challenging to implement.

3. Response for the touch event lagged due to memory issues. This issue was caused due to re-rendering of all the components. It was resolved by implementing code that re-rendered only required components rather than all the components. (File – LastCircle.js)

NOTE: Zoom in and out couldn't be implemented as it required re-rendering of all the components every time the graph was zoomed in and out. Since the graph already had few memory issues, adding this feature would make the app laggy.

Result :

Victory Native charts
Victory Native extends Victory for use on iOS and Android. Victory Native uses an identical API, so Victory code can easily be reused across platforms

Features Implemented
1. Zoom in and out feature
2. Horizontal Scroll View
3. Scaling of graph according to device dimensions

Victory Native charts provides most of the features required such as Zoom in and out, Scroll view, Scaling of the graph dynamically etc. It is easier to implement over CapitalOne

Issues Faced
1.Since victory native charts already uses pan responder for the zoom feature, it cant be used to implement the slider.
2.Response lags on Android devices due to memeory issues.

NOTE: Response for the touch and zoom feature lags only on Android devices whereas it works as expected on iOS devices smoothly

Result: