

Milestone Two

Russell Reckhow

Southern New Hampshire University

CS499: Computer Science Capstone

Professor Joseph Martinez

Narrative

The artifact I used was my Weight Tracker app from CS360. It was created around June of this year. I added a line graph to show the user's weight over time. The graph pulls data from the database, shows it from oldest to newest, and displays it visually with a goal line.

I included this line graph because it demonstrates that I can improve existing software by adding a new feature to it. This also shows frontend and backend work. On the front end, I integrated MPAndroidChart and styled it how I liked with colors, date bubble markers, and a goal line. On the backend, I sorted the weight entries from oldest to newest, connected the graph to both the weight database and goal database, and added error handling to the graph. This addition to the app makes it more useful because it shows the trends visually instead of just looking at a list of numbers.

Yes, I met the outcomes that I wanted so far. With this enhancement, it shows I can design and use solutions by making the weight data easier to understand and visualize better. It also shows I can use tools and techniques like Android Studio and MPAndroidChart. I have demonstrated communication by writing about it and talking about my enhancement through the code review video. The only outcome I still need is security, which will be covered when I add my notebook database later.

I learned how to integrate an open-source library chart into my project and adjust the settings on how I wanted the graph to look. I learned how to set the chart up and display it to show the weights, a date bubble, and the goal line. There were a few challenges that I faced while implementing this. The first being it showed all points on the graph which I thought was a bad idea, so I limited it to only show six at a time and for the user to scroll through it. The second one was the goal line wouldn't show until the weight was closer to it which I thought wasn't

good. I wanted a user to be able to see the goal line the whole time, so I had to fix the view to keep it always showing. The third one I had was when the user took away the goal, it would still appear at the bottom of the graph, so I had to add a way to remove the line when the user deleted their goal weight. The final challenge I had was by showing the dates. The dates wouldn't properly show at the bottom of the graph, so I instead removed them and added a date bubble so the user can click the weight and show the date of it. Solving all these problems really helped me get better at troubleshooting and finding other ways to come up with a fix it.

Data sorting:

```
// Sets the graph from oldest to newest
out.sort(Comparator.comparing( WeightEntry a -> parseDate(a.getDate())));
```

Building and Styling the Graph:

```
// For adjusting the line and circles
LineDataSet dataSet = new LineDataSet(points, label: "");
dataSet.setLineWidth(2f);
dataSet.setCircleRadius(4.0f);
dataSet.setCircleHoleRadius(3f);
dataSet.setDrawFilled(true);
dataSet.setFillDrawable(ResourcesCompat.getDrawable(getResources(), R.drawable.color_fade, requireContext().getTheme()));
```

Adding the Goal Line:

```
// Sets the color and style of the goal line
LimitLine line = new LimitLine(goal, label: "\uD83C\uDFC6"); // Trophy unicode
line.setLineColor(ResourcesCompat.getColor(getResources(), R.color.gray, theme: null));
line.setLineWidth(1.5f);
line.enableDashedLine( lineLength: 12f, spaceLength: 12f, phase: 0f);
line.setLabelPosition(LimitLine.LimitLabelPosition.LEFT_TOP);
line.setTextSize(18f);
yAxis.addLimitLine(line);
```

Marker bubble for dates:

```
public void refreshContent(Entry e, Highlight highlight) {
    int i = Math.round(e.getX());
    if (i >= 0 && i < dateList.size()) {
        // Set the date text for this point
        dateText.setText(dateList.get(i));
    }
    super.refreshContent(e, highlight);
}
```

Graph in the app:



