Milestone Three Narrative

This artifact for the milestone is the same Android Weight tracking app from CS360 that I enhanced last week by restructuring it to a MVVM architecture and integrating room for persistent data storage. In this week's milestone, I focused on algorithmic and data structure improvements to enhance how data is process and presented to the user. For this milestone, I implemented two key enhancements. I added a line chart visualization which utilizes MPAndroidChart to graph the 7 most recent weight entries, and a percent change calculation over the last 7 weight entries to give users a quick way to see a metric of their weight trend.

These enhancements required working with our ordered datasets, converting database entities into data structures that are chart friendly, and perform algorithmic calculations to compute percent changes dynamically. The percent change calculation was designed to handle cases where seven or less entries exist by adapting and updating to available data. Basically, if there are only three entries, it will still be able to show the percent change and the graph. I also finally added a way to delete weights out of the weight table. This artifact demonstrates my ability to combine database data structures, algorithmic processing and UI updates to deliver meaningful insight to users. By adding these enhancements, I was able to showcase my skills in things like implementing algorithms, designing logic that handles data ordering, and integrating external libraries for visual representation of computed data.

This milestone aligns with the course outcome: "Design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices". By adding these enhancements, the application now provides insights algorithmically

rather than just displaying raw stored data. I believe this demonstrates my ability to evaluate trade offs like balancing efficient database queries with client side calculations to achieve practical solutions.

One challenge I faced during this week, was ensuring that weight entries are displayed properly in the line graph. More specifically, in chronological order in both the RecyclerView, and the line graph. When I designed the DAO originally, it returned entries in descending order for efficiency, the data required post processing before plotting. To fix this, I had to reverse the list and ensure that the X-axis values on the chart mapped correctly to the corresponding weight values. On top of this, when adding the percent change calculation, I needed to make sure I avoided errors that would be the result of certain cases. For example, if the initial weight is zero for some reason, I handled this case to avoid the error. Lastly, I had to integrate the calculation and chart updates with LiveData. Because LiveData observes database changes, I needed to ensure that both the chart and the percent change text updated every time new weights were added or removed. This in turn reinforced the importance of MVVM architecture in decoupling logic and UI updates.