Component Specification

1. **Software Components**

This software will involve many Python-curated packages that are essential in modeling biochemical reactions as well as visualization programs to illustrate the change in products over time.

Tellurium:
I will use this Python library to model and simulate biochemical interactions. This is critical as my topic is ruled in part by biochemical processes (this software project will not incoirporate the biomechanical aspect of crossbridge cycling) therefore I must be able to define and simulate these reactions. This package allows me to define multiple variables and have the audience that uses this program to freely alter these variables to a parameter of their choosing. This also has simple plotting and visualization capabilities which I will incorporate to produce more data but will incorporate another Python package to produce more customizable visualizations.

- The code will pertain a sections explaining the numerous variables created.

RoadRunner:
Technically not a stand-alone Pythiong package, this import will function alongside Tellurium to help simulate my biochemical process/ network. Roadrunner allows me to apply ODE's to my software and produce a time scale for these reactions. Again critical as protein function/ kinetics are dynamics. I will not closely analyze these reactions in terms of time to completion. I will more so focus on the overall production over a set period of time in arbitrary units.

Matplotlib:
I will utilize this comprehensive library to create my "final" visualization for my program. This is because although Tellurium is capable of producing plots of simulated reaction, given the magnitude of variables this biochemical reaction utilizes, it is essential to be very descriptive and intentional with my figures. This package is capable of creating 2D/ 3D static and even interactive visualizations in Python. I will definitely take advantage of the highly customizable tools in Matplotlib. Lastly, this data visualization package appealed to me because it works well alongside my last package, Numpy.

Numpy:
This is a powerful numerical computing library in Python utilized by almost every biologist in their computational work. Here I will be using Numpy to calculate the difference between my reaction equations and use the numerical results as comparisons as my system is perturbed.

2. **User Interaction with software components**

This highly customizable software will first have the user run the code to establish foundational parameters and visualization for cardiac muscle contractions. From there, the user will then be able to interact with the code and customize the parameters to reflect their scope of interests. For example, if they were interested in a DCM Myosin Mutation that is located within the myosin head and is predicted to have effects on the myosin actin formation by decreasing myosin's ability to be in an active state which leads to a disease pathology of hypocontractility. They can then examine this step in the code and alter parameters in myosin/action or actomyosin and this software will compute the difference in the rate constants associated with this step. This makes the tool very helpful when combined with clinical data to dissect which part of the chemomechanical cycle is being disrupted by the introduced mutation. Likewise a drug of interest effecting concentration or binding affinity can also be evaluated in a similar way.

(Preliminary Plan recorded in a separate document)