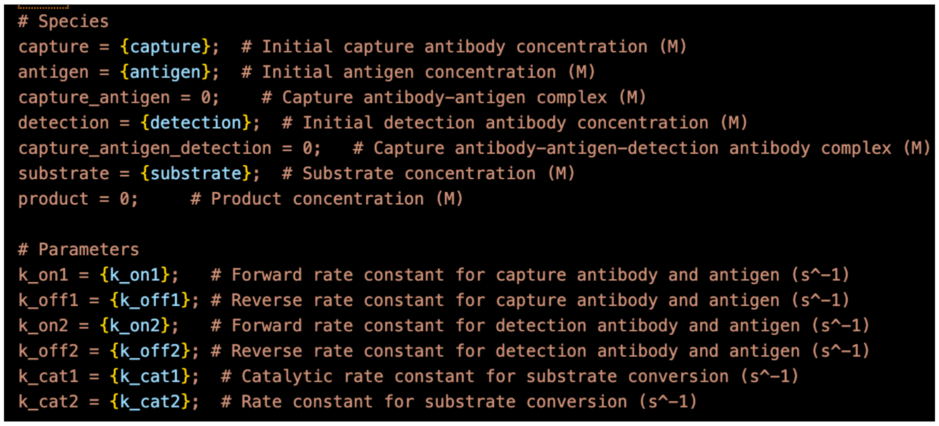
Component Specification Draft

1. **Software components**:
    a. Model ELISA in tellurium.
    Tellurium is a python-based modeling environment for systems and synthetic biology. This package will be used to simulate the kinetics that occur during the ELISA antigen and antibody binding steps.
        i. This component requires user inputs for the species and kinetic constants



```
# Species
capture = {capture};  # Initial capture antibody concentration (M)
antigen = {antigen};  # Initial antigen concentration (M)
capture_antigen = 0;    # Capture antibody-antigen complex (M)
detection = {detection};  # Initial detection antibody concentration (M)
capture_antigen_detection = 0;   # Capture antibody-antigen-detection antibody complex (M)
substrate = {substrate};  # Substrate concentration (M)
product = 0;     # Product concentration (M)

# Parameters
k_on1 = {k_on1};   # Forward rate constant for capture antibody and antigen (s^-1)
k_off1 = {k_off1}; # Reverse rate constant for capture antibody and antigen (s^-1)
k_on2 = {k_on2};   # Forward rate constant for detection antibody and antigen (s^-1)
k_off2 = {k_off2}; # Reverse rate constant for detection antibody and antigen (s^-1)
k_cat1 = {k_cat1};  # Catalytic rate constant for substrate conversion (s^-1)
k_cat2 = {k_cat2};  # Rate constant for substrate conversion (s^-1)
```

        ii.
    b. Run_simulation with tkinter and matplotlib:
    This function interacts with the tkinter and matplotlib packages to create a GUI which intakes user parameters and outputs a plot.
        i. This package will require the following inputs:
            1. “capture_entry": Initial capture antibody concentration (M)
            2. "antigen_entry": Initial antigen concentration (M)
            3. "detection_entry": Initial detection antibody concentration (M)
            4. "substrate_entry": Substrate concentration (M)
            5. "k_on1_entry": Forward rate constant for capture antibody and antigen (s^-1)
            6. "k_off1_entry": Reverse rate constant for capture antibody and antigen (s^-1)
            7. “k_on2_entry": Forward rate constant for detection antibody and antigen (s^-1)
            8. "k_off2_entry": Reverse rate constant for detection antibody and antigen (s^-1)
            9. "k_cat1_entry": Catalytic rate constant for substrate conversion (s^-1)
            10. “k_cat2_entry": Rate constant for substrate conversion (s^-1)
            11. “overall_time_entry": Overall reaction time (s)
        ii. It will output a plot which shows the concentration of the species overtime and the minimum time at which the reaction reaches steady state time.
    c. Clear_plot_and_inputs:
    This clears the plot and input fields if the user wants to start over and test new values

        i. Inputs: entries: dict (referring to the input values), plot_frame:float (referring to the plot)
        ii. Output: the GUI input boxes and the plot are cleared.

2. **Interactions to accomplish use cases**. Describe how the above software components interact to accomplish at least one of your use cases.
    a. **For use case 2**:
        i. When the user runs the program the GUI will initiate with the help of the tkinter function. They will input parameters and press run simulation which will execute the run_simulation function. This will take in the user inputs and insert them into the antimony model, then the model will be simulated, and the results will be outputted in a plot crafted by matplotlib. If the user wants to run another simulation, they press the clear button which will run the clear_plot_and_inputs function to clear all the parameters and graph.
3. **Preliminary plan**. A list of tasks in priority order.
    a. Find kinetic model to describe sandwich ELISA
    b. Recreate kinetic model in python using tellurium
    c. Output a plot that shows that concentration of the species overtime.
        i. On the plot denote the steady state time
    d. Allow user inputs to be incorporated into the tellurium model
    e. Incorporate the model into a GUI
    f. Allow users to specify default parameters
    g. Allow users to clear data and re-run the simulation