**Test Plan and results:**

Preliminary unit testing

1. Show that pb0 and pb1 inputs and tasks are working by reading in the push buttons and outputting the result to an LED
2. Show that the cap sensor task is working properly by reading in various cap values
3. See that the LCD screen works and that it shows preliminary visuals (basic pendulum line, center bar etc.)
4. Hardcode in a pattern and see that the LCD updates and that the pendulum looks at last semi good
5. Set theta to 0 or pi and see that LED1 turns on as this is an error
6. Move the position by using the cap sense and verify that the slider actually moves the position by outputting the value to the LCD screen

Acceptance testing

1. Observe a steady pendulum when no inputs are pressed
2. Push PB0 and see that the pendulum slows down (DONE)
3. Push PB1 and see that the pendulum speeds up (DONE)
4. Push PB0 repeatedly to try and get the pendulum to fail and fall to one side, causing LED1 to turn on

**Statement of where your project stands:**

My project is in the architecture planning and implementation stage. I have been focusing on hammering out the architecture and how different tasks are communicating with each other and the timing of everything.

As of now, we have done 15% of the total work for the project with 10% of the implementation. With the structure of the program thought out, the implementation will be a matter of putting ideas to paper and thinking through how to get this done.

**List of in-scope work items:**

1. Incomplete
   1. Implement a physics engine – 2 hrs
   2. Display functions – 2-3 hrs
2. Complete
   1. Startuptasks
   2. Structure planning
   3. Implement button input tasks – 1 hr
   4. Implement cap slider task – 1 hr
   5. Define unchanging physics – 1 hr

**Update your risk register:**

**Summary:**

It seems that all of the backend functionality is working well with basic testing complete (seeing that the pushbutton will trigger the correct event). The only thing left to do is to implement the physics engine and the display functions.