**Spike Plan**

**Name: Serial port -Socket**

**Context:**

**For this spike, it only consists of a sketch file, a client file, and a server file. Serial port was used to allow the connection to the device, Arduino board. The features that are tested by using with an Arduino board which consist of LED function and motion function but for this spike, only motion function was tested. One of the features that are tested in this spike are server read data from the motion sensor of the board and print the motion data with the timestamp of the server on the console. Moreover, the server collects motion data and send the motion data to the client for display purposes. Lastly, motion data must be received by the client from the server and displays the motion data such as motion status, the timestamp of the server and the time taken for the server to send data to the client which labeled as response time.**

**Gap:**

**For this spike, bugs can be handled easily because the program is simple and short. The code for motions to function properly in the device was separated to a different file so this reduced the complexity of the code structure.**

**Goals:**

* **Make sure the team understands serial port**
* **Verify the library for serial port**
* **Set up serial port and verify the requirements to have serial port to function properly**
* **Check whether the device was connected to the serial port**
* **Check if data was sent correctly.**
* **Check the time taken for the data sent from server to client.**

**Planned start date: 18/4/2017**

**Deadline: 23/4/2017**

**Planning notes:**

**Firstly, we studied the serial port library, syntax, and the requirements to have the serial port to function. We verify the spike requirements to give us a better understanding of the features that are needed to be implemented in the program. We shared the same workspace and done our code through Github. Once done, we compare the code structure with another spike to make sure they work the same.**