Nick Forleo

**Feature 1:** The system shall support screen customization

**Formalization method:** Truth Table

I would choose to make a thrush table for this feature because there are a lot of conditionals that surround this feature. Some requirements for this feature would include:

* If the system is in edit mode, then menu of available data metrics is visible
* If the screen is locked, then the screen is not in edit mode

(by “locked” I mean that the screen display cannot be changed)

* If data cannot be saved to storage, the screen is in edit mode

These requirements can easily be converted to boolean expression, therefore making the truth table formalization method good for validation.

**Feature 2:** The system shall store data

**Formalization Method:** B

I would use B as a formalization method for the data storage system because it will specify exactly how the data shall be processed and stored. Additionally, since it is an “executable specification”, it can be translated to C++ which can help to verify functionality when it comes time for implementation.

**Feature 3:** The system shall be portable

**Formalization Method:** Flow Chart/Diagram

I would use a flow chart to depict this feature because the requirements for different pieces of functionality are dependent on whether or not the system is plugged in. For example, the data transfer to the main station will have to switch from a wired network protocol to a wireless network protocol or switching from battery operated to a direct wired power connection. Other features may change as well, such as current brightness depending on power source. Something like this would be dependent on the current state of the system and a diagram could represent this more accurately and with less ambiguity than words could.