Module	Unit Test Name	Description	Pass/ Fail	Comments
Remote Smartphone Sensor	Basic App Functionality	Verify that pictures are being taken, the correct meta-data is appended to the filename, files are sent to the server (also an Integration Test), and the Garbage Collection Utility is functional.	Pass	Test was successful; everything executed as planned. Module is ready for integration.
	IOIO Interface: External Hardware Interface Test	Demonstrates 12C communication between IOIO and external sensors, as well as USB communication between IOIO and Android phone. Proves future sensor interfacing capability	Pass	All aspects passed. Next test is to replace Arduino with actual sensors, port code into base app.
Remote Power Supply	Power Supply	Verify that the power supply circuit for the phone is operated within specified limits (5V and 0.5-1A) with all possible input voltages (11-14V)	Pass	Successful, ready for integration with overall Remote Sensor.
On-Grid PV Sensor	Power Sensing Circuitry	Verify that the power sensing circuitry for the on-grid sensor operates correctly: current transformer and voltage divider step down signals	Pass	Successful, ready for integration with MSP430.
	Arduino Power Measurements	Test voltage, current, and power measurement accuracy. Verify capability to connect to server over WiFi	N/A	Waiting for parts.
	MSP430/ CC3000 Communication	Test capability of MSP430 to communicate with the CC3000 WiFi chip and	Pass	Communication successful. Need to set up loop for accurately timed

		connect to the UCB Wireless network as well as communication with central server.		data transfer with low power/sleep settings and demonstrate SPI communication with AS8002 power sensing chip
On-Grid Power Supply	Power Supply	Verify that the power supply circuit for the on-grid sensor operates within specified limits (3.3V, 0.3A) with input voltages (120VAC)	Pass	Successful, ready for integration with overall On-Grid Sensor.
Server	Continue waiting	The server continues to wait until data is received.	Pass	Non-blocking socket implementation is always ready for data.
	Data Successfully Transferred	The server receives data sent from multiple sensors	Pass	A new socket is opened for each data set being received.
	Data Pushed onto Queue	Once the data is received it must be pushed onto the queue as a packet.	Pass	Each transmission is put into the queue by matching it with it's associated packet via a group ID.
	Back to Ready State (1)	If data received is not sufficient or there is no present data to be pushed onto the queue the system must return to the ready (waiting) state after filling the queue	Pass	Non-blocking socket implementation is always ready for data.
	Image processor pops image data off queue and runs algorithm	Once all received images pushed onto queue the image processor pops them off and creates forecast	Pass	MATLAB performs this asynchronously with respect to server.

		images and determines motion vectors		
	Writes Power Data	Once all received power data pushed onto queue the database pops off and stores with associated sensor image	Pass	Queue has been implemented as a database table with group IDs that match with the associated image.
	Writes Image/Vector Data	Forecast images and motion vectors outputted from image processor is stored on the database	Pass	Image and vector path is stored in database, not image itself.
	Back to Ready State (2)	Once database stores all relevant data together, must return to the constant state of waiting for data from sensors	Pass	Non-blocking socket implementation is always ready for data.
	User requests map	User selects desired forecast map time range	Pass	Google Earth kml allows user to focus on desired time ranges
	Make KML algorithm runs	Forecasting module runs, which creates a raw data forecast of future solar insulation.	Pass	Loops through template for kml creating one large forecast kml updating the coordinates according to motion vectors.
	NetworkLink KML connects to server forecast kml	Local kml connects to forecast kml on server.	Pass	NetworkLink created with refreshMode = onRequest.
	User Refresh	Network link refreshes when user requests.	Pass	Right click on kml and click refresh.