Test Plan and Results

Overall Test Plan

To verify the functionality of the Metaverse Maintenance diagnostic system, this testing procedure will first test the functionality of all individual software and hardware components. Simulated data will be utilized to understand the connectivity of various components and the latency between the vehicle and final data analysis points. Next, the system, as a whole, will be tested using multiple automotive vehicles to analyze its functionality, accuracy, and reliability. Moreover, both normal and abnormal testing conditions will be utilized to ensure the system is performing in extraordinary operating conditions.

Test Case Descriptions

NC1.1 **Notecard Connectivity**

NC1.2 This test will ensure the IOT connection between the Blues.io notecard and the IOT cloud platform.

NC1.3 The raspberry pi with connected Blues.io cellular notecard will be powered on. A data simulation program will be run to require the cellular notecard to pass values to Losant. The device log will be assessed to ensure communication is established between the device and Losant.

NC1.4 Inputs: Simulated data from the raspberry pi.

NC1.5 Outputs: The device log inside Losant will update with the simulated data from the raspberry pi.

NC1.6 Normal

NC1.7 Blackbox

NC1.8 Functional

NC1.9 Unit Test

OB1.1 **On Board Display Connectivity**

OB1.2 This test will ensure the interface inside the automobile is connected to the raspberry pi and displaying the appropriate dashboard.

OB1.3 The raspberry pi with connected Blues.io cellular notecard will be powered on. A data simulation program will be run, displaying the in-car dashboard.

OB1.4 Inputs: Simulated data from the raspberry pi.

OB1.5 Outputs: The dashboard will display the updating simulation data

OB1.6 Normal

OB1.7 Blackbox

OB1.8 Functional

OB1.9 Unit Test

NI1.1 **Noda Interface 1**

NI1.2 This test will ensure the interface connection between Noda and Losant is working correctly.

NI1.3 Inside of Noda, a vehicle import will be initialized. Losant will process this request and return the vehicle node point map of the selected vehicle. The corresponding map of a vehicle will be displayed for a user. Multiple requests should be attempted for multiple vehicle types to ensure the interface is working.

NI1.4 Inputs: Import model request inside of Noda

NI1.5 Outputs: Vehicle map is displayed for the user.

NI1.6 Normal

NI1.7 Blackbox

NI1.8 Functional

NI1.9 Unit Test

NI2.1 **Noda Interface 2**

NI2.2 This test will ensure the communication between Noda and Losant is working properly.

NI2.3 Data will be simulated for a device inside of Losant. With the vehicle map loaded inside of Noda, corresponding sensor values should change, matching the values from the simulator.

NI2.4 Inputs: Import model request inside of Noda, Losant data simulation

NI2.5 Outputs: Vehicle map is displayed with sensor values and colors changing.

NI2.6 Normal

NI2.7 Blackbox

NI2.8 Functional

NI2.9 Unit Test

NI3.1 **Noda Interface 3**

NI3.2 This test will ensure the communication between Noda and Losant is working properly.

NI3.3 Data will be simulated for a device inside of Losant. With the vehicle map loaded inside of Noda, corresponding sensor values should change, matching the values from the simulator. The simulator will be turned off. Values within Noda should stop changing.

NI3.4 Inputs: Import model request inside of Noda, Losant data simulation

NI3.5 Outputs: Vehicle map is displayed with sensor values and colors changing, then not changing.

NI3.6 Abnormal

NI3.7 Blackbox

NI3.8 Functional

NI3.9 Unit Test

NI4.1 **Noda Interface 4**
NI4.2 This test will ensure the communication between Noda and Losant is working properly.
NI4.3 Inside of Noda, a vehicle import will be initialized. Losant will process this request and return the vehicle node point map of the selected vehicle. The corresponding map of a vehicle will be displayed for a user. A clear model request should be executed. The node map should disappear.
NI4.4 Inputs: Import model request inside of Noda, clear model request inside of Noda.
NI4.5 Outputs: Vehicle map is displayed and then removed.
NI4.6 Normal
NI4.7 Blackbox
NI4.8 Functional
NI4.9 Unit Test

WA1.1 **Web Application 1**
WA1.2 This test will ensure the communication between the web interface and Losant is working properly.
WA1.3 Data will be simulated inside of Losant for all vehicle sensor attributes. Within the web application interface, the real time and historical overview displays should update properly.
WA1.4 Inputs: Losant data simulation.
WA1.5 Outputs: Updating dashboards
WA1.6 Normal
WA1.7 Blackbox
WA1.8 Functional
WA1.9 Unit Test

WA2.1 **Web Application 2**
WA2.2 This test will ensure that the request device ID functionality is working for the web application and that administrators are notified via email.
WA2.3 A request device ID transaction will be triggered through the web application. An email notification should be received by an administrator.
WA2.4 Inputs: Request device ID trigger
WA2.5 Outputs: Email notification
WA2.6 Normal
WA2.7 Blackbox
WA2.8 Functional
WA2.9 Unit Test

WA3.1 **Web Application 3**
WA3.2 This test will ensure the communication between the web interface and Losant is working properly for adding a vehicle.
WA3.3 An add vehicle transaction will be initiated in the web application. All appropriate information will be filled out and submitted. A new entry should appear in Losant corresponding to this vehicle.
WA3.4 Inputs: Add vehicle transaction
WA3.5 Outputs: Updated Losant data table
WA3.6 Normal
WA3.7 Blackbox
WA3.8 Functional
WA3.9 Unit Test

WA4.1 **Web Application 4**
WA4.2 This test will ensure the communication between the web interface and Losant is working properly for removing a vehicle.
WA4.3 A remove vehicle transaction will be initiated in the web application. The corresponding entry should be removed from the Losant data table.
WA4.4 Inputs: Remove vehicle transaction
WA4.5 Outputs: Vehicle removed from data table
WA4.6 Normal
WA4.7 Blackbox
WA4.8 Functional
WA4.9 Unit Test

WA5.1 **Web Application 5**
WA5.2 This test will ensure the communication between the web interface and Losant is working properly for adding a user.
WA5.3 An add user transaction will be initiated in the web application. All appropriate information will be filled out and submitted. A new entry should appear in Losant corresponding to this user.
WA5.4 Inputs: Add user transaction
WA5.5 Outputs: Updated Losant data table
WA5.6 Normal
WA5.7 Blackbox
WA5.8 Functional
WA5.9 Unit Test

WA6.1 **Web Application 6**
WA6.2 This test will ensure the communication between the web interface and Losant is working properly for logging a user into the application.
WA6.3 A user will enter their log-in information into the web application and click “Login”. They will be accepted into the system.
WA6.4 Inputs: User login information
WA6.5 Outputs: Access to the system
WA6.6 Normal
WA6.7 Blackbox
WA6.8 Functional
WA6.9 Unit Test

WA7.1 **Web Application 7**
WA7.2 This test will ensure the communication between the web interface and Losant is working properly for logging a user out of the application.
WA7.3 A user will enter their log-in information into the web application and click “Login”. They will be accepted into the system. They will then log out of the system and not be able to access the system.
WA7.4 Inputs: User login information, log out transaction
WA7.5 Outputs: Access to the system is no longer granted
WA7.6 Normal
WA7.7 Blackbox
WA7.8 Functional
WA7.9 Unit Test

WA8.1 **Web Application 8**
WA8.2 This test will ensure the communication between the web interface and Losant is working properly for keeping a user from logging into the application.
WA8.3 A user will enter their log-in information incorrectly into the web application and click “Login”. They will not be accepted into the system.Various variations of a username and password should be tried to ensure security.
WA8.4 Inputs: Incorrect user login information
WA8.5 Outputs: Access to the system is not granted
WA8.6 Abnormal
WA8.7 Blackbox
WA8.8 Functional
WA8.9 Unit Test

VT1.1 **Vehicle Test 1**
VT1.2 This test will ensure the proper power requirements are met by a vehicle to power

the system.

VT1.3 The device will be plugged into the vehicle’s USB charging port. The in-car display should power on.

VT1.4 Inputs: Power cord plugged into the USB port.

VT1.5 Outputs: Powered display

VT1.6 Normal

VT1.7 Blackbox

VT1.8 Functional

VT1.9 Unit Test

VT2.1 **Vehicle Test 2**

VT2.2 This test will ensure the vehicle is communicating with the device.

VT2.3 The bluetooth OBD II connector should be connected to the car. With the device powered on, vehicle data should be displayed on the dashboard.

VT2.4 Inputs: OBD II connection

VT2.5 Outputs: Vehicle data displayed on dashboard

VT2.6 Normal

VT2.7 Blackbox

VT2.8 Functional

VT2.9 Unit Test

VT3.1 **Vehicle Test 3**

VT3.2 This test will ensure the vehicle is communicating with the device.

VT3.3 The bluetooth OBD II connector should be disconnected from the car. With the device powered on, vehicle data should not be displayed on the dashboard.

VT3.4 Inputs: Powered device with no OBD II connection

VT3.5 Outputs: Vehicle data is not displayed on dashboard

VT3.6 Abnormal

VT3.7 Blackbox

VT3.8 Functional

VT3.9 Unit Test

VT4.1 **Vehicle Test 4**

VT4.2 This test will ensure the vehicle is communicating with the device accurately.

VT4.3 The bluetooth OBD II connector should be connected to the car. With the device powered on, vehicle data should be displayed on the dashboard. Speed, RPM, and runtime should be monitored by the vehicle's odometer, and clock. These values should be compared for accuracy.

VT4.4 Inputs: Vehicle data

VT4.5 Outputs: Vehicle data displayed on dashboard

VT4.6 Normal
VT4.7 Blackbox
VT4.8 Functional
VT4.9 Unit Test

VT5.1 **Vehicle Test 5**
VT5.2 This system functions properly for a vehicle.
VT5.3 The bluetooth OBD II connector should be connected to the car. With the device powered on, vehicle data should be displayed on the dashboard. The car should be operated normally by a user. Vehicle data should be compared with the Losant dashboards, web application, and Noda interface, to ensure proper function.
VT5.4 Inputs: Vehicle operation with device connected
VT5.5 Outputs: Data measurements across multiple displays.
VT5.6 Normal
VT5.7 Whitebox
VT5.8 Performance
VT5.9 Integration Test

OA1.1 **OBD II Adapter 1**
OA1.2 This test ensures all LEDs on the OBD II adapter are operational.
OA1.3 The OBD II connector should be connected to the car. With the device powered on and vehicle data properly displayed on the dashboard, the LEDs on the OBD II connector should have a red LED always on, a green LED to indicate data sent from the device to the vehicle, a yellow LED to indicate data sent from the vehicle to the device, another green LED to indicate data sent from the device to the raspberry pi, and another yellow LED to indicate data sent from the raspberry pi to the device.
OA1.4 Inputs: OBD II connection
OA1.5 Outputs: LEDs
OA1.6 Normal
OA1.7 Blackbox
OA1.8 Functional
OA1.9 Unit Test

OA2.1 **OBD II Adapter 2**
OA2.2 This test ensures correct communication between the vehicle and OBD II adapter.
OA2.3 The OBD II connector should be connected to the car. With the device powered on, the OBD II connector should also be connected to a computer instead of the raspberry pi. Manually send commands through the OBD II port to the vehicle and observe its output. Ensure vehicle output is intended based on command.

Ensure red power LED stays on, green LED for communication sent from device to vehicle flashes when commands are sent to the vehicle, and the yellow LED for communication sent from the vehicle to device flashes when commands are sent to the device.

OA2.4 Inputs: OBD II connection, laptop
OA2.5 Outputs: LEDs, laptop
OA2.6 Abnormal
OA2.7 Whitebox
OA2.8 Performance
OA2.9 Integration Test

OA3.1 **OBD II Adapter 3**
OA3.2 This test ensures correct communication between the OBD II adapter and the raspberry pi.
OA3.3 The OBD II connector should be connected to the car so it can be powered on. With the device powered on, the OBD II connector should also be connected to the raspberry pi and a laptop. Manually send commands from the raspberry pi to the adapter, and manually send commands from the adapter to the raspberry pi. Ensure the commands are correct and receive correct responses. Ensure red power LED stays on, green LED for communication sent from device to raspberry pi flashes when commands are sent to the raspberry pi, and the yellow LED for communication sent from the raspberry pi to device flashes when commands are sent to the device.
OA3.4 Inputs: Raspberry pi, laptop
OA3.5 Outputs: LEDs, laptop
OA3.6 Abnormal
OA3.7 Whitebox
OA3.8 Performance
OA3.9 Integration Test

Test Case Matrix

Test Case ID	Normal/ Abnormal	Blackbox/ Whitebox	Functional/ Performance	Unit/ Integration
NC1	Normal	Blackbox	Functional	Unit
OB1	Normal	Blackbox	Functional	Unit
NI1	Normal	Blackbox	Functional	Unit

NI2	Normal	Blackbox	Functional	Unit
NI3	Abnormal	Blackbox	Functional	Unit
NI4	Normal	Blackbox	Functional	Unit
WA1	Normal	Blackbox	Functional	Unit
WA2	Normal	Blackbox	Functional	Unit
WA3	Normal	Blackbox	Functional	Unit
WA4	Normal	Blackbox	Functional	Unit
WA5	Normal	Blackbox	Functional	Unit
WA6	Normal	Blackbox	Functional	Unit
WA7	Normal	Blackbox	Functional	Unit
WA8	Abnormal	Blackbox	Functional	Unit
VT1	Normal	Blackbox	Functional	Unit
VT2	Normal	Blackbox	Functional	Unit
VT3	Abnormal	Blackbox	Functional	Unit
VT4	Normal	Blackbox	Functional	Unit
VT5	Normal	Whitebox	Performance	Integration
OA1	Normal	Blackbox	Functional	Unit
OA2	Abnormal	Whitebox	Performance	Integration
OA3	Abnormal	Whitebox	Performance	Integration