Verification and Validation Report
Formulate

Team 25, MECHTRON 4TB6
Ahmed Nazir, nazira1
Stephen Oh, ohs9
Muhanad Sada, sadam
Tioluwalayomi Babayeju, babayejt

April 5, 2023

1 Revision History

Date	Version	Notes
March 3, 2023	1.0	Final Revision

2 Symbols, Abbreviations and Acronyms

symbol	description
Formulate	Mechatronics Engineering Capstone Course
UART	Universal Asynchronous Receiver-Transmitter
SPI	Serial Peripheral Interface
I2C	Inter-Integrated Circuit
USB	Universal Serial Bus
TCP	Transmission Control Protocol
IP	Internet Protocol
SPDT	Single pole, double throw
TCP/IP	Transmission Control Protocol / Internet Protocol
SOC	Software On Chip
SPST	Single pole, single throw
TX	Transmit
RX	Recieve
USB	Universal Serial Bus
LED	Light-emitting diode
GB	GigaByte
SD	Secure Digital
PCB	Printed Circuit Board

Contents

1 Revision History **i**

2 Symbols, Abbreviations and Acronyms **ii**

3 Functional Requirements Evaluation **1**
- 3.1 Sensor Verification . . . 1
- 3.2 Device Telemetry . . . 1
- 3.3 Device Hardware . . . 2
- 3.4 Desktop Application . . . 3
- 3.5 Data Dashboard . . . 3
- 3.6 Database . . . 4

4 Nonfunctional Requirements Evaluation **5**
- 4.1 Usability . . . 5
- 4.2 Performance . . . 8
- 4.3 Security . . . 9

5 Unit Testing **10**
- 5.1 Module - ui_functions.py . . . 10
- 5.2 Module - mainArduino.ino . . . 17
- 5.3 Module - mainESP8266.ino . . . 17

6 Changes Due to Testing **18**
- 6.1 Functional Requirements . . . 18
- 6.2 Nonfunctional Requirements . . . 18
- 6.3 Unit Testing . . . 18

7 Trace to Requirements **19**

List of Tables

1 Sensor Validation . . . 1
2 Device Telemetry . . . 1
3 Device Hardware . . . 2
4 Desktop Application . . . 3
5 Data Analytics . . . 3
6 Database . . . 4
7 Usability . . . 6
8 Performance . . . 8
9 Security . . . 9
10 Unit Test - connect_wireless(self) . . . 10

11 Unit Test - disconnect_wireless(self) . . . 10
12 Unit Test - connect_wired(self) . . . 11
13 Unit Test - disconnect_wired(self) . . . 11
14 Unit Test - ping(self) . . . 12
15 Unit Test - startTest(self) . . . 12
16 Unit Test - stopTest(self) . . . 13
17 Unit Test - declineData(self) . . . 13
18 Unit Test - setup_page . . . 13
19 Unit Test - login_into_app . . . 14
20 Unit Test - continue_signup . . . 15
21 Unit Test - move_to_submit_test . . . 16
22 Unit Test - browse_and_display_pictures . . . 16
23 Unit Test - upload_test_info . . . 16
24 Unit Test - Arduino Bytestring . . . 17
25 Unit Test - ESP8266 Wireless Test . . . 17

List of Figures

1 ST-U 1-A Test User Ratings . . . 6
2 ST-U 1-B Test User Ratings . . . 7
3 Average Latency Results of Each Test . . . 8

3 Functional Requirements Evaluation

3.1 Sensor Verification

Test Number	Type	Input	Output	Result
ST-SV 1	Manual	Room at a constant 25 °C ambient	Constant 25.4 °C reading from temperature sensor	Pass
ST-SV 2	Manual	Room at constant 40% humidity °C	Constant 43% reading from humidity sensor	Pass
ST-SV 3	Manual	4 seperate phonecalls with accelerometer measuring haptic feedback	Maximum error between acceleration profiles of phone calls within 0.2 meters per second squared	Pass

Table 1: Sensor Validation

3.2 Device Telemetry

Test Number	Type	Input	Output	Result
ST-DT 1	Manual	The device is connected via Wi-Fi and the start button is pressed on the GUI	The device beings to send and display the sensor data in the GUI	Pass
ST-DT 2	Manual	The device is connected via Wi-Fi and the start button is pressed multiple times in the GUI	The start button greys out not allowing the user to press it multiple times and the sensors still send data to the GUI	Pass
ST-DT 3	Manual	The device is connected via Wi-Fi and the stop button is pressed on the GUI	The device stops sending data to the GUI	Pass
ST-DT 4	Manual	The device is connected via Wi-Fi and the stop button is pressed multiple times in the GUI	The stop button greys out not allowing the user to press it multiple times and the device does not send data	Pass

Table 2: Device Telemetry

3.3 Device Hardware

Test Number	Type	Input	Output	Result
ST-DH 1	Manual	Temperature sensor male end JST connector unconnected to the device's female JST connector	Device measured the room's ambient temperature at 23.4 °C	Pass
ST-DH 2	Manual	Device collecting sensor data at low battery charge	Device unoperational upon battery depletion and halted sensor data collection	Fail
ST-DH 3	Manual	5 kg dumbbell placed on each corner of the device chassis	No plastic failure in device chassis	Pass
ST-DH 4	Manual	A member on the formula electric team was given a cross screwdriver, M4 screw, the device, and DIN rail	The formula electric team member mounted the device in 40 seconds	Pass

Table 3: Device Hardware

ST-DH3 failed the functional test case. The team estimated the time required to design and integrate a rechargeable battery subsystem within the project's timeline was 1.5 weeks and decided the effort was not worthwhile relative to other project objectives. As a result, the failure of this test case lead the expected operation at low battery to be adjusted. Users are now expected to replace the battery on a regular schedule to prevent a failure in test integrity due to charge depletion.

3.4 Desktop Application

ST-DA3 and ST-DA4 are included in unit testing, refer to section 5.3 for further details.

Test Number	Input	Expected Output	Actual Output	Result
ST-DA5	User clicks on 'Start Test' button in 'View Test' page	Live data start to populate a table in the UI	Live measurement data is displayed in a table in the UI	pass

Table 4: Desktop Application

3.5 Data Dashboard

Test Number	Type	Input	Output	Result
ST-DD 1	Manual	The user was given a username and password to login into Power BI	The user was able to view all the different data that was being recorded during the test and previous tests as well	Pass
ST-DD 2	Manual	The user was given a fake username and password that is not authorized to view the data	The user was not able to view the Power BI dataset that was created to view all the different test data	Pass

Table 5: Data Analytics

We created our Data Dashboard through a visualization tool called Power Bi and tested it to see if we were able to connect to the database which contained all our data values. We able to get all the values of our data from the database since Power Bi has a method that can connect to database that authorized users are allowed to connect to. Since this was the case we were able to pass this this requirement for our the data analytics.

3.6 Database

Test Number	Input	Expected Output	Actual Output	Result
ST-D1	User clicks on 'Submit' button in 'Submit Test' page consecutively in a short period of time	UI displays error message 'Too many submissions' in red	'Too many submissions' is displayed in red	pass

Table 6: Database

4 Nonfunctional Requirements Evaluation

4.1 Usability

Test Number	Type	Input	Output	Result
ST-U 1-A	Manual	User connects a thermistor to device, begins collecting sensor test data and gathers data for 1 minute, completes collecting sensor test data, adds remarks to the test, and submits test data to database	User completed the overall process in 3 minutes 43 seconds and rated the sensor mount and measuring procedure a 5. All other categories were given a 4	Pass
ST-U 1-B	Manual	User adjusts Arduino code for a fluid flow rate sensor, connects the sensor to the device, begins collecting sensor test data and gathers data for 1 minute, completes collecting sensor test data, adds remarks to the test, and submits test data to database	User completed the process in 49 minutes 15 seconds and rated the overall experience a 2. The sensor mount and measuring procedure categories were given a 5. All other cattegories were given a 4	Fail
ST-U 2-A	Manual	Device collecting sensor data with wired connection to a laptop running the python application	Python application GUI displayed at minimal latency. Less than 1 second latency for changes in sensor measurements to display on GUI. Users selected Formulate for all 4 categories	Pass
ST-U 2-B	Manual	Device collecting sensor data with wireless connection to a laptop running the python application	Python application GUI displayed at minimal latency. Less than 1 second latency for changes in sensor measurements to display on GUI. Users selected Formulate for all 4 categories	Pass

ST-U 3-A	Manual	Device connection to application is broken. User submits previous test data held on the micro-SD module to the database	Test data submits and test contents can be viewed on dashboard. An average score of 4 was provided	Pass
ST-U 3-B	Manual	Application connection to database is broken. User can connect a temperature sensor, start a test, and stop a test	Device recognizes sensor, starts test, and stops test. An average score of 4 was provided	Pass

Table 7: Usability

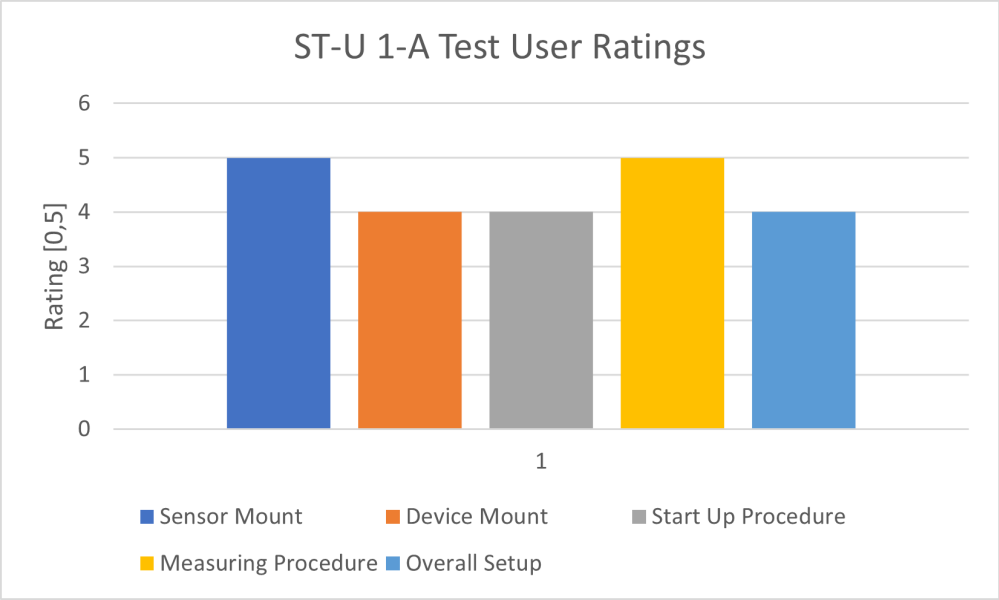

Figure 1: ST-U 1-A Test User Ratings

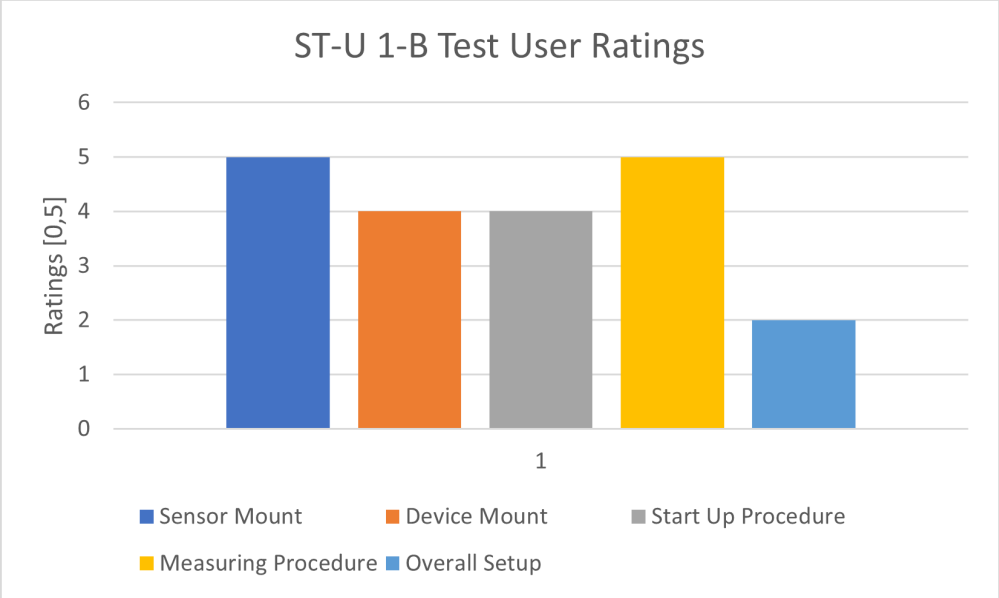

Figure 2: ST-U 1-B Test User Ratings

ST-U 1-B failed the usability test case. The difference between ST-U 1-B and ST-U 1-A was the user requirement to adjust and write the Arduino code for a sensor not previously used with the device. Notably, the user experienced difficulty and was intimidated with adjusting existing code to integrate a new sensor. This effect was shown in both the increased time to complete the overall process and the decreased rating in the overall test experience category. As a result, the project will have an approach whereby the user should only interact with the graphical user interface to reduce the user's feelings of complexity and intimidation when integrating a new sensor. The goal is to abstract the adjustments in the backend when implementing a new sensor by having the user interact only with the GUI and following guided steps in plain English to fill in the required information to integrate a new sensor.

4.2 Performance

Test Number	Type	Input	Output	Result
ST-P 1	Dynamic, Manual	The device will be mounted and be tested in various conditions	The device was operational and stayed physcially intact after being tested in various conditions	Pass
ST-P 2	Dynamic, Manual	The device will be operating and collecting data from a test and streaming the results to our desktop application	The latency between the collecting of data and the streaming of it remained below 10 seconds	Pass
ST-P 3	Dynamic, Manual	The device will be operating with one or more of the connections to either the device, application and or database	The user is still able to obtain the test results of the current test no matter which connections are missing on the overall connection	Pass

Table 8: Performance

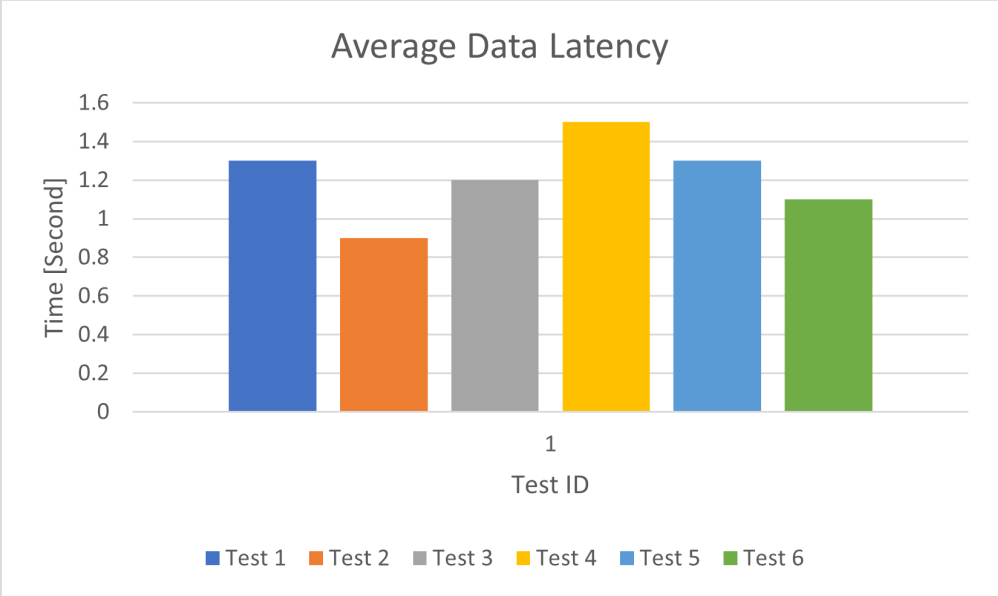

Figure 3: Average Latency Results of Each Test

4.3 Security

Test Number	Type	Input	Output	Result
ST-S 1	Dynamic, Manual	User will receive numerous usernames and passwords and attempt to access the database	The database was only accessible using the correct username, password and only with specific ip addresses that have been accepted	Pass
ST-S 2	Dynamic, Manual	Multiple fake data points were attempted to be added to the database	The database remained the same even after the attempted modification ensuring that only authorized were allowed to modify the database	Pass

Table 9: Security

5 Unit Testing

5.1 Module - ui_functions.py

connect_wireless(self) - Tested by clicking connect button on wireless page

Test No.	Input	Expected Output	Actual Output	Result
U1	Connected to Formulate Wi-Fi	The application should connect to the device and display that it is connected	The application shows that the device is connected	Pass
U2	Not Connected to Formulate Wi-Fi	The application should display a pop up error indicating to the user that they need to connect to the Formulate Wi-Fi	A popup is displayed to the user to indicate they need to connect to Wi-Fi	Pass
U3	-Connected to Formulate Wi-Fi -Device is connected via Serial (Wired)	The application should disconnect from serial and follow U1	The application disconnects from serial and follows U1	Pass

Table 10: Unit Test - connect_wireless(self)

disconnect_wireless(self) - Tested by clicking disconnect button on wireless page

Test No.	Input	Expected Output	Actual Output	Result
U4	-Connected to Formulate Wi-Fi -The disconnect button is clicked	The application should disconnect from the board and display that it is disconnected	The application disconnects from board and shows its disconnected	Pass
U5	Not Connected to Formulate Wi-Fi	The application should not display the disconnect button on the connectivity widget	The application does not display the disconnect button	Pass

Table 11: Unit Test - disconnect_wireless(self)

connect_wired(self) - Tested by clicking connect button on wired page

Test No.	Input	Expected Output	Actual Output	Result
U6	Connected to PC via USB and the correct COM port is selected in the Wire connectivity widget	The application should connect to the device and display that it is connected	The application shows that the device is connected	Pass
U7	Not Connected to PC via wire	The application should not display any COM port in the widget	The wired drop down is empty and shows no COM port	Pass
U8	-Connected to PC via USB and the correct COM port is selected in the Wire connectivity widget -Device is connected via WiFi	The application should disconnect from WiFi and follow U6	The application disconnects from WiFi and follows U6	Pass

Table 12: Unit Test - connect_wired(self)

disconnect_wired(self) - Tested by clicking disconnect button on wired page

Test No.	Input	Expected Output	Actual Output	Result
U9	-Connected via wired -The disconnect button is clicked	The application should disconnect from the board and display that it is disconnected	The application disconnects from board and shows its disconnected	Pass

Table 13: Unit Test - disconnect_wired(self)

ping(self) - Tested by clicking connect button

Test No.	Input	Expected Output	Actual Output	Result
U10	Connected via WiFi	The application should read which sensors are flashed on the board and display them	The application gets which sensors are flashed and displays them	Pass
U11	Connected via wired	The application should read which sensors are flashed on the board and display them	The application gets which sensors are flashed and displays them	Pass

Table 14: Unit Test - ping(self)

startTest(self) - Tested by clicking startTest button

Test No.	Input	Expected Output	Actual Output	Result
U12	Connected via WiFi	The application should read data from the bytestring sent from the ESP8266 and display it in the table as the data is coming	The application reads the data and displays it correctly	Pass
U13	Connected via wired	The application should read data from the bytestring sent from the Arduino UNO and display it in the table as the data is coming	The application reads the data and displays it correctly	Pass

Table 15: Unit Test - startTest(self)

stopTest(self) - Tested by clicking stopTest button

Test No.	Input	Expected Output	Actual Output	Result
U14	Connected via WiFi	The application should stop reading data sent from the ESP8266	The application stops reading the data	Pass
U15	Connected via wired	The application should stop reading data sent from the Arduino UNO	The application stops reading the data	Pass

Table 16: Unit Test - stopTest(self)

declineData(self) - Tested by clicking stopTest button

Test No.	Input	Expected Output	Actual Output	Result
U16	There is data populated in the table	The application should clear all the data in the table and display an empty table	The application clears the data and displays an empty table	Pass
U17	There is no data populated in the table	The application should clear all the data in the table and display an empty table	The application clears the data and displays an empty table	Pass

Table 17: Unit Test - declineData(self)

setup_page(self, page_name) - Output observed in the UI

Test No.	Input	Expected Output	Actual Output	Result
U18	'sign up page'	Sign up page has items added to the 'team role' dropdown and password/confirm password fields have hidden entry	Sign up page has items in 'team role' dropdown and password fields have the input hidden	pass
U19	'login page'	Login page's password field entry is hidden	Login page's password field entry is hidden	pass

Table 18: Unit Test - setup_page

login_into_app(self) - Tested by clicking 'Sign In' button in Login page of the UI

Test No.	Input	Expected Output	Actual Output	Result
U20	Signing in with correct user information and 'admin' user	UI transitions to homepage and username is displayed in the top right corner	UI transitioned to homepage and 'admin' was displayed in the top right corner	pass
U21	Signing in with no password	UI displays error message 'Please enter a password' in red color	'Please enter a password' is displayed in red	pass
U22	Signing in with no username	UI displays error message 'Please enter a username' in red color	'Please enter a password' is displayed in red	pass
U23	Signing in with non-existing username	UI displays error message 'User does not exist, try again' in red color	'User does not exist, try again' is displayed in red	pass
U24	Signing in with incorrect password	UI displays error message 'Incorrect password, try again' in red color	'Incorrect password, try again', is displayed in red	pass

Table 19: Unit Test - login_into_app

continue_signup(self) - Tested by clicking 'Continue' button in Sign Up page of the UI

Test No.	Input	Expected Output	Actual Output	Result
U25	Valid account registration information	UI transitions to homepage and new rows are added to Users and Login tables in database	UI transitioned to homepage and new rows with user information and hashed passwords were added to Users and Login tables in database	Pass
U26	At least one field is missing input	UI displays error message 'Please fill in missing fields' in red color	'Please fill in missing fields' is displayed in red	Pass
U27	Username is less than 4 characters	UI displays error message 'Username requires at least 4 characters' in red color	'Username requires at least 4 characters' is displayed in red	Pass
U28	Password is less than 8 characters	UI displays error message 'Password requires at least 8 characters' in red color	'Password requires at least 8 characters' is displayed in red	Pass
U29	Password does not have a number	UI displays error message 'Password requires at least one number' in red color	'Password requires at least one number' is displayed in red	Pass
U30	Password does not contain a letter	UI displays error message 'Password requires at least one alphabet' in red color	'Password requires at least one alphabet' is displayed in red	Pass
U31	Password does not contain a non-alphanumeric character	UI displays error message 'Password requires at least one non-alphanumeric character' in red color	Correct error message is displayed but is cut off	Fail
U32	Password and Confirm Password fields do not match	UI displays error message 'Passwords do not match' in red color	'Passwords do not match' is displayed in red	Pass

Table 20: Unit Test - continue_signup

move_to_submit_test(self): Tested by clicking on 'Submit' button in View Test page

Test No.	Input	Expected Output	Actual Output	Result
U33	Clicking on button while logged in	UI transitions to Submit Test page	UI transitioned to Submit Test page	Pass
U34	Clicking on button while not signed in	UI displays error message 'Please login to submit tests'	'Please login to submit tests' is displayed	Pass

Table 21: Unit Test - move_to_submit_test

browse_and_display_pictures(self) - Tested by clicking on Upload Image button in Submit Test page

Test No.	Input	Expected Output	Actual Output	Result
U35	Described above	File explorer window pops up, user can select an image, and selected image is displayed in UI	The same as expected output	Pass

Table 22: Unit Test - browse_and_display_pictures

upload_test_info(self) - Tested by clicking on 'Submit' button in Submit Test page

Test No.	Input	Expected Output	Actual Output	Result
U36	Valid data and test information provided	UI transitions to homepage and new test information should be visible in PowerBi dashboard	New test information was visibile in PowerBi dashboard	Pass
U37	No test data	UI displays popup with error message 'No data to submit'	'No data to submit' error popup was displayed	Pass
U38	Valid test data but missing test name and/or purpose	UI displays popup with the error message 'Fill in missing fields'	'Fill in missing fields' error popup was displayed	Pass

Table 23: Unit Test - upload_test_info

5.2 Module - mainArduino.ino

Test No.	Input	Expected Output	Actual Output	Result
U39	Device is connected via Wired and temperature sensor is connected	A bytestring of data in the form of (A1XX) should display	A bytestring of the correct form is displayed on the Arduino serial monitor	Pass
U40	Device is connected via WiFi and temperature sensor is connected	A bytestring of data in the form of (A1XX) should display	A bytestring of the correct form is displayed on the Arduino serial monitor	Pass

Table 24: Unit Test - Arduino Bytestring

5.3 Module - mainESP8266.ino

Test No.	Input	Expected Output	Actual Output	Result
U41	Device is connected to the Formulate WiFi	A Green LED should turn on	Green LED is turned on	Pass
U42	Device is connected to Formulate WiFi and data is being transferred	An orange LED should turn on	A Orange LED is turned on	Pass

Table 25: Unit Test - ESP8266 Wireless Test

6 Changes Due to Testing

6.1 Functional Requirements

FR 11 was changed such that the battery under expected operational use is non-rechargeable. Batteries will continue to be used but will change from rechargeable to non-rechargeable batteries with a scheduled replacement timeline close to charge depletion. This change followed the test ST-DH 2 results.

6.2 Nonfunctional Requirements

Additional considerations to the GUI must be made in response to NFR1, NFR2, and the test result ST-U 1-B. In particular, the team plans to improve the GUI to improve the user experience by minimizing and ultimately eliminating interaction with Arduino code when integrating new sensors and making the user work with the GUI to integrate new sensors.

6.3 Unit Testing

There was only one failed unit test case which was U31. The test case failed because the correct error message was being displayed but, it was cut off meaning only a portion of the error message was visible. In order to fix this, the size of the UI text window for the error message was increased and the result was that the UI now correctly displays the full message.

7 Trace to Requirements

Requirement	Test
FR 1	ST-SV 1, ST-SV 2, ST-SV 3
FR 2	ST-DT 1, ST-DT 2, ST-DT 3, ST-DT 4
FR 3	ST-DT 1, ST-DT 2
FR 4	ST-DT 3, ST-DT 4
FR 5	ST-SV 1, ST-SV 2, ST-SV 3
FR 6	ST-DA 1
FR 7	ST-DA 3, ST-DA 4
FR 8	ST-DD 1
FR 9	ST-DH 3
FR 10	ST-DH 4
FR 11	ST-DH 2
FR 12	ST-DT 1, ST-DT 2, ST-DT 3, ST-DT 4
FR 13	ST-DH 4
FR 14	ST-DA 5
FR 15	ST-DH 1, ST-DA 5
FR 16	ST-DD 2
FR 17	ST-DD 1
FR 18	ST-DA 3
NFR 1	ST-U 1-A, ST-U 1-B
NFR 2	ST-U 2-A, ST-U 2-B, ST-U 3-A, ST-U 3-B
NFR 3	ST-P 2
NFR 4	ST-P 1, ST-P 3
NFR 5	ST-U 2-A, ST-U 2-B
NFR 6	ST-P 1
NFR 7	ST-P 3
NFR 8	ST-U 2-A, ST-U 2-B
NFR 9	ST-S 1, ST-S 2

Appendix — Reflection

In our Verification and Validation Plan we had planned to create a website containing all the information and test data received throughout various tests by Mac Formula Electric. The website will organize data and help the user analyze it as well. We ultimately decided to use Power Bi, which is an interactive data visualization software that all McMaster students can access. We used Power Bi to meet our data visualization requirements as it helped improved user ease of use and compatibility with the database where our test data was being stored. We were able to verify that using Power Bi worked for our data analytics portion because the members of the Mac Formula Electric were able to use the visualized test data to aid them in future designs.

The Verification and Validation Plan for usability differed from the actual activities conducted in the VnV. For example, the usability test ST-U 1 did not originally account for sensor code integration by the user during the test. But during VnV, the team realized that we needed to know both the total process time taken when the user already had the required code and when the user had to integrate the code to adequately determine the project's ability to meet the FR's and NFR's defined in the SRS. In the future, prototyping earlier in the design process can help identify important test cases when generating the VnV Plan.

For the most part, our VnV plan the physical device and embedded code was closely aligned with the actual VnV activities we conducted. During the VnV planning session, our team worked diligently to clearly outline the specific objectives and requirements for our final product. We conducted extensive research, and actively engaged with Formula Electric team members to gain a deeper understanding of the type of device they were looking for. This helped us develop a comprehensive VnV plan that was tailored to the specific needs and expectations of the project. While there were some changes to the plan as we progressed, the core elements were largely unchanged. This was due to a thorough planning process and our ability to anticipate potential challenges or modifications that might arise over the course of the project. Ultimately, this allowed us to execute our VnV activities with precision and confidence, and to demonstrate the high level of quality that was required for the final product.

After unit, desktop application, and database testing, the GUI verification activities were different from the plan in a variety of ways. Firstly, there was no unit testing in the plan because it was too early in the project to determine modules and units therefore, most test cases for unit testing were new. Secondly, some desktop application and database system tests mentioned in the plan were not applicable because some features were changed or not added. During UI implementation, it was possible to anticipate those changes in future projects but it would depend on the nature of the project. In this case, the project's UI was modified greatly from initial design to implementation. However, since some backend features of the project changed, it is completely normal for the verification to be also modified to accommodate the new or edited features of the UI.