5 Testing

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, power system, or software.

The testing plan should connect the requirements and the design to the adopted test strategy and instruments. In this overarching introduction, given an overview of the testing strategy and your team's overall testing philosophy. Emphasize any unique challenges to testing for your system/design.

In the sections below, describe specific methods for testing. You may include additional types of testing, if applicable to your design. If a particular type of testing is not applicable to your project, you must justify why you are not including it.

When writing your testing planning consider a few guidelines:

- Is our testing plan unique to our project? (It should be)
- Are you testing related to all requirements? For requirements you're not testing (e.g., cost related requirements) can you justify their exclusion?
- Is your testing plan comprehensive?
- When should you be testing? (In most cases, it's early and often, not at the end of the project)

5.1 Unit Testing

All of the components of our project are going to be tested thoroughly including both hardware and software wings of our project. Hardware components such as, Arduino, RF Module, and sensor will be tested on a number of constraints such as accuracy, water-resistance, power consumption, power draw, longevity, and durability. Software will also be tested and making sure that the interface between the hardware and software goes smoothly with minimal data loss if any and will make sure that users can create and access their accounts in an error-free way and retrieve their information easily.

Hardware

Temperature Sensor Test Plan

Assembly

- Connect Temperature Sensor to Arduino MCU
    - Thermistor
        1. Wire thermistor and a similar value resistor in series
        2. Connect the thermistor end to the Arduino power supply port and the resistor end to the GND pin on Arduino.
        3. The node between the thermistor and resistor will be connected to an analog input pin on the Arduino.

Expected Results

- All obtained values are expected to be within 2 degrees of the actual temperature

Test Procedure

- Once the temperature sensor is configured with the Arduino MCU, the temperature sensor will be subjected to various controlled temperatures ranging from 0 to 110 degrees Fahrenheit.

RF Module Test Plan

Assembly

- Connect RF Transmitter to User's Arduino MCU
    - Three connections:
        - Module VCC pin to Arduino power supply pin
        - Module Data-in pin connected to a digital pin on the Arduino
        - Module GND pin to ground on Arduino
- Connect RF Receiver to Supervisor's Arduino MCU
    - Three connections:
        - Module VCC pin to Arduino power supply pin
        - Module Data-in pin connected to a digital pin on the Arduino
        - Module GND pin to ground on Arduino
- Upload software for data transmission and retrieval

Expected Results

- Values obtained by the User's Arduino MCU should match the values that appear on the Supervisor's Arduino MCU.
- All values must be properly formatted.

Test Procedure

- Devices will be placed in an isolated area to minimize electromagnetic interference with the device.
- Device will be subjected to interference and the team will measure its performance under those conditions. Interference will include both Narrow Band and Broadband.

Power Supply Test Plan

Assembly

- Connect power supply to Arduino MCU
    - Connect the power supply positive terminal to Vin pin on Arduino and the negative terminal to GND.

Expected Results

- Power supply should be able to power all system components for at least 8 hours under various environmental conditions

Test Procedure

- Initialize the system powered by the power supply.
- Expose system to various environmental conditions such as cold temperatures and hot temperatures.
- Leave the system running continuously while monitoring the voltage levels across the terminals of the battery to ensure regulated voltage.
- Monitor system and power supply performance over the entire time span to ensure proper system functionality.

Enclosure Test Plan

Assembly

- Install all components inside the enclosure
    - Arduino MCU
    - RF Module and Antenna
    - Power Supply
    - Auditory Alert Buzzer

Expected Results

- All the hardware components should be able to fit securely inside the enclosure.
- Enclosure should be water-resistant
- Enclosure should not severely affect the communication capabilities of the BLE and RF modules

Test Procedure

- Expose the enclosure with hardware inside to various conditions that users will face
- Direct exposure to snow
- Simulated impact on the enclosure from falling on the ski hill
- Test connectivity through the exposure by testing the RF modules outside the enclosure versus inside the enclosure

5.2 Interface Testing

What are the interfaces in your design? Discuss how the composition of two or more units (interfaces) are being tested. Tools?

Interfaces

Arduino MCU and RF Module

Arduino MCU BLE Module to Phone Application

Power Source and Arduino MCU

Arduino MCU, Temperature Sensor, LED Indicators, Alert Alarm, and Power Supply

Arduino MCU and Temperature Sensor

Testing

Arduino MCU and RF Module

- This interface will be tested by using the RF transmitter configured to the Arduino of the user to transmit the collected data from the temperature sensing device to the RF receiver configured to the Arduino of the Supervisor. The data should be transmittable for distances up to 500m as constrained by the client. Also, the RF transmitter must be able to transmit at least one data reading per minute to the supervisor's RF receiver as specified by the client.

Arduino MCU BLE Module to Phone Application

- This interface will be tested by sending data from the BLE Module to the Phone Application and by sending data from the Phone Application to the device. Since these interfaces will be within close proximity at all times, the BLE Module does not need to have a large transmission range. However, the BLE Module must be able to reliably send correctly-formatted data to the Phone Application and must be able to do so at least once per minute as specified by the client.
- Additionally, if a loss of connection ever occurs between the Arduino MCU BLE Module and the Phone Application, the Arduino MCU must be able to store the offline data and upload it once

connection is restored. In order to test this functionality, our group will connect the Arduino MCU and Phone Application, begin transmitting data readings from the Arduino MCU to the Phone Application, break the connection between the Arduino MCU and Phone Application, and analyze that the Arduino MCU's internal storage is correctly storing data readings. From there, we will restore the connection between the Arduino MCU BLE Module and Phone Application and ensure that the data readings stored in the Arduino MCU's internal storage are correctly uploaded.

Power Source and Arduino MCU

- This interface will be tested by using the power source to power the Arduino MCU. The client's desired battery life is at least 8 hours of use. Our group will test this by leaving the device running in a similar environment and documenting the battery drain and total battery life. Depending on the results, a larger or smaller power source may be utilized in the final design. Another aspect that we will need to test is whether the power source is outputting enough voltage for the Arduino for the entire time. We will test this by measuring the voltage output of our power source at many different battery levels. It will be most important to test the voltage output when the battery level is low.

Arduino MCU and Temperature Sensor

- This interface will be tested rigorously in order to ensure data readings are accurate within 2 degrees Fahrenheit, as specified by the user. Our plans for testing are to integrate the Arduino MCU and Temperature Sensor, expose the temperature sensor to controlled temperature environments, and analyze the accuracy of the sensor. Failure to meet the required accuracy will result in the utilization of a different temperature-sensing element.

Arduino MCU, Temperature Sensor, LED Indicators, Alert Alarm, and Power Supply

- This interface captures a vast majority of the hardware wing for our project design. Once the Arduino MCU and Temperature Sensor interface has been tested to ensure accurate data readings, our team will integrate the LED indicators and Alert Alarm to the Arduino MCU. From there, our device must be able to alert the user through the LED indicators and Alert Alarm when temperature readings indicate the user is approaching or has approached dangerous temperature levels. The temperature ranges will be customizable for the user based on preferences; therefore, our team will be testing upon numerous temperature ranges to ensure that the system responds accordingly. Additionally, the LED indicators will also be used to visually indicate low-battery of the device. Therefore, our team will be continually measuring the terminals across the power supply and ensuring that when the device's power level falls below 20% that the LED indicators respond accordingly as well.

5.3 Integration Testing

What are the critical integration paths in your design? Justification for criticality may come from your requirements. How will they be tested? Tools?

Definition: different units, modules or components of a software application are tested as a combined entity

Hardware:

- From a hardware standpoint, the first and most important integration revolves around the Arduino MCU and Temperature Sensor. These interfaces will be integrated and tested upon as described

previously to ensure accurate data readings. From there, our team will integrate the LED indicators and Alert Alarm and test to ensure the alert systems meet all requirements of the system. Continuing on, our team will integrate the RF Module and BLE Module; however, the BLE Module is already built-in to the Arduino MCU. Our team will test to ensure data transmission and retrieval meet all design requirements. Finally, our team will integrate the power supply to ensure that it has the capability to power all system components for at least 8 hours subjected to various extreme ambient temperature conditions.

Software:

- The account setup page, when completed with proper information (no improperly formatted information), creates a user in the database and allows that user profile to interact as a logged in profile.
    - Critical, as if the user cannot make an account, they cannot save preferences (temp range, sharing info with other profiles, etc.) and the app becomes redundant.
    - Tested by JUnit tests. Create many account creation requests, some with purposely bad information. Account creations with good info should be accounts that are created and logged in the database, bad account creations trigger error response.
- The sensor page will be tested to make sure that it displays and updates the temperature from the sensor on specific period and if the current temperature exceeds the max temp, the app will start the alarming process.

Combination:

- The hardware components bluetooth/radio-frequency device sends properly formatted data to the user's/supervisor's android device.
    - Critical, as if the hardware does not send data, or the data is incomplete/incorrect, the software cannot accurately alert the user of potential issues.
    - Tested via both in-person testing and JUnit testing. In-person testing will have the hardware fed a temperature range, and see if the software shows correct info. JUnit testing by feeding the hardware random variables, and seeing how it displays on software.

5.4 System Testing

Describe system level testing strategy. What set of unit tests, interface tests, and integration tests suffice for system level testing? This should be closely tied to the requirements. Tools?

Controlled system testing

System testing will utilize tests from combined integrations testing and most of the interface testing mentioned in the previous sections. This can be done by exposing the temperature sensor to various conditions and verifying that the correct parameters are met. Given a specific temperature, the system will need to react with the following outputs.

- Auditory alert from the hardware module
- Appropriate LED indications from hardware module
- Transfer of data via Bluetooth and RF
- Application receives and accurately processes the data

- Temperature displayed in the application is correct

Real World System Testing

This system will be exposed to the conditions mentioned in the Unit testing section. This will be done by using the device for its intended purpose. Using the device in a sport such as snowboarding or skiing will allow us to test the different variables mentioned in our unit testing section such as

- Weather conditions
- Data transfer ranges
- Power supply duration

5.5 Regression Testing

How are you ensuring that any new additions do not break the old functionality? What implemented critical features do you need to ensure do not break? Is it driven by requirements? Tools?

In regards to hardware, one way that we will ensure that the new additions do not break old functionality will be by understanding the requirements and specs of the new hardware addition. We will utilize circuit analysis to ensure the proper integration of the new hardware. Finally, we can test all the components together. Testing the components all together will ensure that the critical features such as an 8-hour battery life, appropriate data transfer ranges, durability will not break.

For software additions we are utilizing Git. This allows the team to alter the code and make new additions and test them without affecting the master code. If a new software addition is successful, the altered branch can be merged to the master branch.

5.6 Acceptance Testing

How will you demonstrate that the design requirements, both functional and non-functional are being met? How would you involve your client in the acceptance testing?

Through our planning process, we have kept a close relationship with Adaptive Adventures and veterans who will use our product. We will do acceptance testing with the client in a few ways.

Since we are working with people of different levels of experience with technology, we will do user testing of the application with the client. The application user testing can be done fairly frequently because it can be done remotely.

For example, when it comes to the application, our team has a few different visual designs options for the main page of the application. For this, we will conduct a survey within adaptive adventures and the veterans that we work with, to allow them to decide which page is easier for the user to understand. We will allow the clients to comment and from this feedback, we will determine which visual is better for the veterans and Adaptive Adventures.

When all the screens are implemented on our application, we will use a virtual environment with the client to make sure that they can find all the information necessary based on the requirements. Examples of

information to find on these screens is location of battery level indication, location of body temperature indication, location to set unsafe body temperature, location to change password, location to create profile, ability to log in to application, location to log out, and location of saved data. For a supervisor, we will test that they can do the previous items as well as see multiple profiles and see what the user has set the body temperature indicator to. In addition to making sure that these items can be achieved, we will ask about the user experience like what was difficult and what can be improved. We will do these tests a few times throughout the process to make sure that the application is as easy as possible to use.

To test the hardware requirements, we will do functional and non-functional testing with both the client and just our team. Since we are not near our client, it is difficult to do in person testing, so we must test it it the best of our abilities before giving it to the client for testing. As a team, we will check that the application works on multiple phones and connects accurately to the hardware. We will test that the hardware can function below –17 degree wind chill, accurately. We will measure temperature manually and make sure that the application reacts accordingly with all of the error messages and reactions. We will also check that the application is storing user data correctly, the hardware is waterproof, and the battery life of the hardware is accurately recorded on the application, and it lasts at least a day. We also can test that the hardware can speak to the software at distances of 300 meters.

The client will participate in a lot of the non-functional testing of the hardware. The client will test that the equipment is comfortable and easy to use. They will test that the equipment works for all their purposes while participating in the sports that they intend to use the equipment for. The clients will test that they understand when the equipment is alerting them of unsafe body temperatures and battery levels. We will then give the client a similar survey to ask about how easy and painless using the equipment is for the users.

5.7 Security Testing (if applicable)

5.8 Results

What are the results of your testing? How do they ensure compliance with the requirements? Include figures and tables to explain your testing process better. A summary narrative concluding that your design is as intended is useful.

We have not yet done the testing, but the table below shows a compiled list of requirements which must be met in testing.

Requirement	Requirement Met?	
	Yes	No
Power Supply – 8+ Hour Life at 0 degrees F		
Power Supply – 8+ Hour Life at 110 degrees F		
Thermistor - Reading within 2 degrees Fahrenheit from 0 to 110 degrees F		

RF Module – Transmitted Data Matches Received		
RF Module – Connectivity at 500m		
RF Module – One reading transmitted per minute		
Enclosure – Components function after simulated fall while skiing		
LED Indicator – Lights below 30% battery capacity		
Audio Alarm – Activates at specified temperature levels		
Application – Displays real-time temperature reading		
Application – Allows supervisory users to view user’s temperature		
Application – Historization of temperature data for one year		