7. System Test Plan

The system will be tested in four sections; the first section will test the client software, the second section will test the embedded software on the Arduino device, the third section will test the hardware on the cane, and the fourth section will test the full expected operating procedure of the device. Each section will also be tested for its interaction with the other section; mainly, the communications between user interface and hardware device and the protocol utilized. Afterwards, the standard operating procedure of the device will be tested.

7.1 Software Application Test Plan

To ensure that each component of the user interface maintains its functionality, unit tests will be implemented to target specific functions in the.

- The XML parser will be tested against various types of proper and improper XML messages.
- The user input fields will be tested against various proper and improper user inputs
- Google maps API querying will be tested for basic mapping queries
- The device communications protocol that has been established to communicate with the hardware device will be tested to verify that the proper syntax and symbols are sent to the proper USB port.
- All necessary text will be verified to be readable by screen-reading and text-to-speech technology
- The accuracy of the GPS coordinates will tested against the GPS module's coordinates
- Test the application's data processing speed against computers with varying processing power and internet connection

7.2 Embedded Software Test Suite

To ensure that each basic function of the embedded software on the device maintains its functionality, a suite of unit tests are being implemented which target specific functions in our program.

- The communications port will have unit tests functions that verify each aspect of the communications protocol that has been established to communicate with the client. This includes waypoint lists, waypoint list size, and clear commands.
- The magnetometer and its accompanying functions have unit test functions tested to verify the calibration values that had been determined by re-running the calibration routine to find the maximum and minimum pin values.

- The path detection algorithm will have unit tests to verify its proper functionality and for its response to various proper and improper inputs.
- The waypoint detection algorithm will have unit tests to verify its proper functionality and for the response to various proper and improper inputs.
- The waypoint lists on the device will have unit tests to ensure that it tracks and switches waypoints properly upon reaching a new node location.

7.3 Hardware Tests

These tests target specific hardware devices utilized in our design to ensure that they are each in working order including the Arduino itself. Each pin on the Arduino device will be activated in turn to verify that they work

The motors will be individually tested for activation at varying duty-cycles. They will be tested for activation, strength, and consistency within a degree of accuracy.

1 Utilize haptic motor test sketch
2 Turn on the device and charge to full capacity with hardware modules activated and connected to a PC
3 Note the changes in vibrational feedback in relation to the serial monitor data

The ultrasonic sensor will be tested to verify it reports the proper range and voltages.

1 Utilize ultrasonic sensor test sketch
2 Turn on the device and charge to full capacity with hardware modules activated and connected to a PC
3 Place an object at various distances to the ultrasonic sensor
4 Read the values returned to the serial monitor by the Arduino on the serial monitor
5 Note any major differences in values

The magnetometer will be tested to verify the calibration values that had been determined by re-running the calibration routine to find the maximum and minimum pin values. The magnetometer will also be tested for proper heading and accuracy against other compasses.

1 Utilize compass calibration sketch
2 Turn on the device and charge to full capacity with hardware modules activated and connected to a PC
3 Read the values returned to the serial monitor by the Arduino
4 Note any major differences in values
5 With current values, read compass bearing and compare with standard compass and electronic compasses

The GPS will be tested for activation, rate of message frequency, baud rate, positional accuracy, and message accuracy (checksum).

1. Turn on the device and charge to full capacity with hardware modules activated
2. Query GPS for status
3. Query for position
4. Utilize another GPS device and Google maps to verify the accuracy

The battery will be tested for basic functionality (the ability to power the device) and for idle device battery capacity.

Test 1:

1. Turn on the device and charge to full capacity with hardware modules deactivated
2. Leave idle
3. Note Time to Power Failure

Test 2:

1. Turn on the device and charge to full capacity with GPS, Compass, and Ultrasonic Sensors activated
2. Leave Idle
3. Note Time to Power Failure

7.4 Operating Procedure Test

This test simulates the simplest full workflow for our device. The tester will follow the steps listed below.

1. Utilize the software user interface on a PC to select a simple route from their current location within walking distance (1-5km)
2. Upload the waypoints to the hardware on the NaviCane
3. Disconnect the device from the PC with a full-charge
4. Follow the haptic-feedback guided waypoints to their destination
5. Specifically note the absence or presence of feedback based on approaching objects and obstacles that fall within the activation region for the ultrasonic sensor detection
6. Note the charge on the battery during (if the battery fails) and after the run through of the procedure

7.5 Demo Test

This test demonstrates the functionality of our device that is testable within the confined space of a demonstration room.

1. Connect the NaviCane to the com port on the PC
2. Utilize the software user interface on the PC to select a simple route from their current location within walking distance (1-5km) and note the location of the waypoints on the displayed map
3. Upload the waypoints to the hardware on the NaviCane
4. Disconnect the device from the PC with a full-charge
5. Sweep the NaviCane to detect the first waypoints direction and note the strength of the feedback
6. If possible, walk in the direction of the waypoint indicator and note the presence or absence of a change in feedback direction relative to the distance of the waypoint.
7. Detect obstacles that are blocking the path to waypoint with vibration feedback.
8. Test the intensity of the vibration feedback correlated to the distances of the obstacle
9. Repeat steps 6 - 8 until reaching the final destination
10. Test the reverse route by doing steps 6 - 8 to get back to the original location.
11. Test the wireless charging of the device.