



System Test Plan

Project Team: Gurjot Singh Atwal
Sukhreet Kaur
Rajdeep Kaur
Masih Amiri
Kevin Chang

Contact Person: Sukhreet Kaur
Ska142@sfu.ca

Submitted to: Dr. Andrew Rawicz – ENSC 440w
Steve Whitmore – ENSC 305w
School of Engineering Science
Simon Fraser University

Issued On: March 28, 2016
Revision: 1.0

Test Plan

The following test plan will outline a sequence of tests to demonstrate the specifications outlined in the Functional and Design Specification documents.

1. Hardware Components (Unit & integration testing)

Component	Areas/Type of Test	Test description
Raspberry Pi microprocessor	Power through USB port (requires 5V input)	Connect the USB port to different sources of power such as PC, Laptop, and wall sockets and observe that the power indicator LED light turns on
	Pin voltage when turned on	Using a Volt meter to measure the voltage on the pins with assigned voltages from product manual and compare with the values given in the manual
Three-axis Accelerometer	Three axis values	Moving the Accelerometer should change x,y,z values displayed on Raspberry Pi terminal displayed on PC or laptop.
	Device calibration	Connect the Accelerometer to Raspberry Pi that is connected to PC or laptop, and execute the calibration code on Matlab
NFC Reader/Tag	NFC reader activation	Two group members shake hands while wearing the smartBand. The test is a success when the LED light on Raspberry Pi blinks, indicating the NFC module has been activated by the accelerometer
	Input/output test	The led on microprocessor blinks when the NFC reader receives a tag id after a handshake. And to ensure the correct tag ID has been received, the number will be checked on terminal by connecting the Raspberry Pi to a laptop or PC
	Distance covered	measure the maximum distance from which the NFC reader can receive the tag ID
SD Card	Capacity	Use H2testw software to test all memory locations to confirm SD card capacity

	Read/Write speed	Use a card reader to plug the card into computer and verify the speed of placing data on the card and opening/running that data.
Bluetooth module	Connection to Raspberry Pi	Run hcconfig code on Raspberry Pi terminal to see if it detects the Bluetooth module
	Pairing with mobile phone	Check to see if the Bluetooth module can be paired with the phone
	Distance covered	Measure the maximum distance the Bluetooth module can be detected by the phone
Battery	The amount of time the batteries will keep the smartBand functioning	Power up the microprocessor using the batteries and keep it on until the device turns off

2. Android Application

Component	Area/Type of Test	Test Description
UI	User acceptance test	Have 10-15 individuals run the application on their Android phones and provide feedback on the user interface's convenience and functionality
	Performance test	Use dumsys android tool to verify that the user interactions with the phone are running at consistent 60 frames per second without any dropped or delayed frames
Application Performance	Manual testing	Developers constantly test any changes they make on local repositories and fix all the bugs found in the code
	Asynchronous testing	The application requires SQL database connection and data retrieval. The connection and data download times are tested to make sure the user can access the data with no delays

	Integration testing	Check the compatibility of all the application modules with each other and ensure that each module works on its own and is bug-free
--	---------------------	---

3. System (Integrated unit testing)

Component(s)	Area/Type of Test	Test Description
Android application & Bluetooth module	Pairing Test	Check to see if the application can detect the Bluetooth ID of the module after pairing the module with the phone
NFC tag, Bluetooth module paired with the phone, and Android application	Tag ID registration	Use a NFC tag ID to see if Bluetooth module name is overwritten by that ID. Check to see if each ID can be automatically registered for a new account on the application
NFC Reader, Bluetooth module paired with the phone, and Android application	Connection tag ID input/output test from the Bluetooth module	After a handshake, check to see if the tag ID of another user has been received from the NFC Reader and sent by the Bluetooth module to the application
	Tag ID profile retrieval	Send different tag IDs from the Bluetooth module to the application. Check to see if the application can retrieve the correct profile information corresponding to each ID from the database
	smartBand to application data transfer timing	Measure the time it takes the Raspberry Pi to receive and send a tag ID to Bluetooth module and also the time it takes for the application to receive the id from Bluetooth module. This determines how long the user has to wait for each handshake. Minimizing this delay is fundamental for this project