DeVELOPMENT OF A Wireless BODY AREA sensing system for ALCOHOL CRAVING study

A Thesis

Presented to

The Faculty of the Graduate School

At the University of Missouri

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

By

SANDEEP RAVI

Dr. Yi Shang, Advisor

DECEMBER 2013

The undersigned, appointed by the dean of the Graduate School, have examined the thesis entitled

DeVELOPMENT OF A Wireless BODY AREA sensing system for ALCOHOL CRAVING study

Presented by Sandeep Ravi

A candidate for the degree of

Master of Science

And hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Yi Shang

Dr. Jianlin Cheng

Dr. Tim Trull

**ACKNOWLEDGEMENTS**

It gives me great pleasure to acknowledge the continuous support and guidance of my advisor Dr. Yi Shang throughout the development of my master’s project. His scholarly advice, meticulous scrutiny and patient encouragement have helped me immensely in development of this project. I would also like to thank my lab mate Ruiqi Shi for his assistance and contributions to the project. I would also like to thank the members of Wireless Sensor Network Lab for insightful advice and assistance. I would like to thank Dr. Tim Trull and Dr. Jianlin Cheng for being a part of my defense committee and for their invaluable suggestions and comments on this project.

**Table of Contents**

[*1. Introduction 6*](#_Toc371709612)

[*2. Related Work 8*](#_Toc371709613)

[*3. System Architecture 10*](#_Toc371709614)

[*3.1 External Sensor Module 10*](#_Toc371709615)

[*3.1.1 Affectiva Q Sensor 11*](#_Toc371709616)

[*3.1.2 Equivital EQ02 Sensor 12*](#_Toc371709617)

[*3.2 Mobile Computing Module 13*](#_Toc371709618)

[*3.3 Networking Module 13*](#_Toc371709619)

[*3.4 Data Collection Module 13*](#_Toc371709620)

[*3.5 Survey Module 14*](#_Toc371709621)

[*3.6 Subject Monitoring Module 15*](#_Toc371709622)

[*4. System Implementation 16*](#_Toc371709623)

[*4.1 Mobile Computing Module 16*](#_Toc371709624)

[*4.1.1 Networking Module 17*](#_Toc371709625)

[*4.1.2 Data Collection Module 20*](#_Toc371709626)

[*4.1.3 Survey Module 22*](#_Toc371709627)

[*4.2 Subject Monitoring Module 25*](#_Toc371709628)

[*5. Implementation Challenges 27*](#_Toc371709629)

[*5.1 Memory Leaks 27*](#_Toc371709630)

[*5.2 Power Management Strategy 29*](#_Toc371709631)

[*5.3 Testing Power Saving Strategy 31*](#_Toc371709632)

[*5.3.1 Test-1 31*](#_Toc371709633)

[*5.3.2 Test-2 33*](#_Toc371709634)

[*6. Results 36*](#_Toc371709635)

[*7. Future Work 38*](#_Toc371709636)

[*8. Summary 39*](#_Toc371709637)

[*9. References 40*](#_Toc371709638)

## Abstract

Alcohol Craving can be described as a powerful urge to drink or as intense thoughts about alcohol. There are lots of environmental stimuli and physiological urges that can prompt a person to crave for alcohol, which cannot be accurately measured by traditional clinical or ambulatory assessments. Most of these traditional approaches mainly reply on questionnaires and interviews with examiners where the subject is asked to rate the intensity of his or her desire to drink, sometimes in the presence of an alcohol-related cue. The main problem of using these approaches is that they don’t consider the Mood Dysregulation caused by real-time environment factors and physiological urges that can cause alcohol craving in a person.

To overcome this problem, in this project a wireless body area sensing system has been developed for detecting real-time alcohol cravings by collecting, storing, and analyzing physiological and environmental sensor data that accompany craving such as Electro Dermal Activity (EDA), heart rate, respiratory rate, body temperature, body movement, location, surrounding temperature, etc. The system contains three main components: wireless external sensors, a Smartphone, and a web server. Currently, the system supports two types of external physiological sensors wirelessly connected to an Android Smartphone. The Smartphone is a central point of the system, managing multiple wireless connections with external sensors, collecting sensor data, and transmitting the data to the web server when it is connected to the Internet. In addition, the system has a survey module implemented on the Smartphone, using multiple preset and randomly triggered surveys to determine the emotional state of the subject along with the environments factors that prompted the subject to drink alcohol. A web site has been developed to support interactive display of both archived and real-time sensor and survey data on the server. The system has been carefully designed to avoid memory leak and reduce power consumption. Experimental results show that the system satisfies the needs of the targeted alcohol craving study and can run smoothly for over 10 hours on battery.

1. Introduction

Smart Phones have powerful computational and sensing capabilities which make them a great platform to build context-aware systems that can monitor various environment factors such as Location, Body Movement, Humidity, Temperature, Light, and Pressure. Context aware systems are basically systems which use the contextual information about a user to change the way system behaves. In context aware computing, context can be described as any information that can be used to describe the state of something that is relevant to a user’s interaction with an application [1]. These sensing capabilities of a smart phone can be coupled with external physiological sensors to form a wireless body sensor network that retrieves physiological data of a person under various environment factors.

The main idea of the project is to use these context aware systems to monitor and analyze mood dysregulation, craving, and alcohol use of subjects who have history of alcohol abuse using ambulatory assessment. Ambulatory assessment is the use of field methods in clinical psychology to analyze subjects in natural or unconstrained environments [2]. Mood dysregulation refers to abnormally intense and frequent experience of negative emotions. In addition to collecting environmental information and physiological information, the context aware system collects a subject’s emotional state through self-reported and random surveys. This information about the subject’s physiological and emotional state can be used together to build predictive models to identify and predict mood dysregulation and alcohol cravings that subject might have in the future and provide possible real-time interventions in terms of sending the subject a message or notifying the authorities about the current physiological and emotional state of the subject. The Body Sensor Application which is deployed on the smart phone plays an important part in this project. Functionality of Body Sensor Application includes collecting emotional data of the subject from self-reported and random surveys, managing the connections between two external physiological sensors which connect to the application via Bluetooth and collecting physiological data of the subject retrieved from the sensors. It uses the internal sensors of the phone to retrieve information such as GPS co-ordinates, Accelerometer values, Current Temperature, Pressure, Humidity, and Light. It is also responsible for transferring the data to server and providing real time interventions when necessary based on emotional and physiological data of the subject. Due to the ubiquitous nature of Android enabled devices and due to the open source nature of the Android SDK (Software Development Kit), Android Operating System was considered as a platform to develop and deploy the Body Sensor Application in this project.



Figure (1): Android market share in the year 2012

2. Related Work

**2.1 IPainRelief**

The main idea of IPainRelief is to replace the traditional approaches of indentifying pain such as paper-based pain inventory systems that provide a momentary snapshot of the condition with a framework that can provide real time evaluation and treatment suggestions for epidermis pain. The IPainRelief [3] application connects with several multimodal wireless sensors embedded together to assess the magnitude of pain. These multimodal wireless sensors collect physiological data of the subject such as HR (Heart Rate), SC (Skin Conductance), ST (Skin Temperature) and RR (Respiratory Rate) which are used as parameters to train neural network. Neural Networks are used in the project try to determine the intensity of pain corresponding to the physiological data retrieved from sensors. Based on the intensity of pain determined from the neural networks the smart phone suggests necessary treatment the subject has to take for the ailment. Using this approach, IPainRelief mainly aims to improve the effectiveness of physical therapy interventions by helping the patients who are at risk for delayed functional recovery and who may benefit from additional non- impairment -based approaches.

There four main components or activities in the IPainRelief Application .The first two activities are responsible for connecting to the multimodal wireless sensors, retrieving, displaying the physiological data from the sensors and determining the intensity of pain. The third activity requires the subject to indentify the type of pain he or she is experiencing which is later combined with pain intensity and sent to the SQLite database as query. The final activity is responsible for displaying necessary treatment that has to be taken based on the pain values.

**2.2 Daily Mood Assessment based on Mobile Phone Sensing**

This paper mainly discusses a framework for assessing and analyzing mood in daily life called “MoodMiner” [4]. The main purpose of MoodMiner is to detect mood related mental health problems such as mood disorders, depressions, and elation in real-time. MoodMiner mainly uses mobile phone data such as mobile phone sensor data and communication data (including acceleration, light, ambient sound, location, call log, etc.) to extract human behavior pattern and assess daily mood. MoodMiner consists of client application deployed on Android Operating System which works offline with a back-end server built in java .The back-end server’s functionality includes collecting data from all users and creating a personalized model that outputs daily mood based on collected data.

The Smart Phone Application in MoodMiner consists of three components whose functionality mainly includes periodically collecting data from the smart phone sensors such as Accelerometer, GPS, Light Sensor and Sound Sensor, obtaining communication statistics from Call and SMS logs, and recording user-reported mood status on daily basis. This data collected from different sensors are feed to mood assessment model based on factor graph to detect the mood of the subject. This framework was deployed and was tested on 15 users for a period of the 30 days and the accuracy of the model was observed to be 50%.One of the main disadvantages of this framework would be its inflexible server design which cannot accommodate extra computational power and storage space if needed since mobile phones have limited extensibility .Lack of external physiological sensors in the framework is also a drawback because external physiological sensors coupled with internal sensing capabilities can increase the performance of the existing assessment model .

3. System Architecture

The Context Aware Android System consists of three important modules which are External Sensor Module, Mobile Computing Module and Subject Monitoring Module. The Mobile Computing Module can be further divided into Networking Module, Data Collection Module and Survey Module.

3.1 External Sensor Module

External Sensor Module consists of two physiological sensors: Affectiva Q sensor, Equivital EQ02 Sensor. These two sensors are responsible for recording and transmitting a subject physiological state to the mobile computing module. The functionality of these two sensors are described below

3.1.1 Affectiva Q Sensor

The Affectiva Q sensor is responsible for recording and transmitting physiological traits of subject such as Accelerometer data, Skin Conductance also called as Electro Dermal Activity (EDA), and Temperature. The sensor is worn around the wrist as shown in the figure (2) below. It connects to Mobile Computing Module via Bluetooth. The sensor while transmitting physiological data uses Bluetooth’s RFCOMM protocol which ensures simple reliable data stream to the Mobile Computing Module and since RFCOMM protocol has wide spread support and publicly available API’s ,it is compatible with most of the Android Operating System versions .

The rate at which the physiological data is transmitted can be configured on Q sensor; the transmission rate can be configured between 2Hz to 32Hz per second. The Q sensor has in built memory storage which can serve as a backup in case the Bluetooth connection fails while transmitting data. The data is transmitted as a string containing values such as Sequence Number, 3-axis Accelerometer Data, Battery, Temperature and Electro Dermal Value where each value is comma separated. The EDA value transmitted is measured in micro Siemens and Temperature in Celsius.



Figure (2): Affectiva Q Sensor

3.1.2 Equivital EQ02 Sensor

Equivital EQ02 Sensor has two main components: EQ02 Sensor Belt and Sensor Electronics Module (SEM) .The EQ02 Sensor Belt is embedded with a system of sensors and electrodes which connect with the Sensor Electronics Module (SEM) to provide mobile monitoring capability. The EQ02 Sensor can sense, record and monitor physiological traits of a subject such as Heart Rate, Breathing Rate, Body Position, Body Movement, Core Temperature, Skin Temperature, Accelerometer Data and Oxygen Saturation in Body. The EQ02 Sensor transmits data via Bluetooth and also uses RFCOMM protocol to transmit data to the Mobile Computing Module in the system. Unlike Affectiva Q sensor, EQ02 Sensor transmits encoded data which can be decoded only by using core decoder library provided in the Equivital SDK which is responsible for parsing the raw data from the SEM, and providing a set of high level functions to interface with it. Primarily, the decoder uses a publish/subscribe (push-based) model; the body sensor application in the mobile computing module will subscribe to one or more messages which they are interested in and the decoder will notify any subscribed clients when data is available. The rate at which physiological data is transmitted can be configured using Equivital Manager. The EQ02 sensor also has an inbuilt memory storage which can serve as a backup in case the Bluetooth connection fails while transmitting data. The figure (3) below shows how a EQ02 sensor is worn around the chest.



Figure (3): Equivital EQ02 Sensor

3.2 Mobile Computing Module

Mobile computing module mainly consists of a Smart Phone Application called “Body Sensor App” .Body Sensor App is context aware mobile application which is developed to run on Android Enabled Devices. Body Sensor App can be divided into three modules: Networking Module, Data Collection Module and Survey Module. The functionality of these modules is explained below and implementation of these modules will be discussed later in the report.

3.3 Networking Module

Networking module’s functionality includes initializing, maintaining, reporting changes in Bluetooth connection with the external physiological sensors .Networking module is also responsible for transmitting data collected from external physiological sensors to remote server .It determines if data connectivity exits in form Wi-Fi or GPRS before transmitting data the physiological data.

3.4 Data Collection Module

Data Collection Module maintains a backup of the data transmitted from External Sensor Module by storing them in text files. Text files are used in this application to store data because they provide a better time complexity in terms of reading and writing large amounts of data when compared to SQLite database which requires complex queries to be run against the database to retrieve or store data which can increase the time complexity and response time. Data Collection Module is also responsible for storing data from surveys answered in the survey module and data collected from internal sensors of the phone such as Accelerometer, Location Sensor, Light Sensor, Humidity Sensor, Temperature Sensor and Pressure Sensor.

3.5 Survey Module

Survey Module consists of a series of self-reported surveys and a random survey. These surveys are designed to analyze and record a subject’s emotional state during ambulatory assessment of the subject. The self-reported surveys are primarily categorized into six types: Morning Report, Initial Drinking, Mood Dysregulation, Craving Episode and Alcohol Consumption. A subject is required to answer the morning report survey when the subject starts to use the Body Sensor Application in the morning. The Initial Drinking Survey needs to be completed when the subject first drinks. Craving Episode Survey needs to be completed when the subject experiences an alcohol craving. Mood Dysregulation survey needs to be completed when the subject suffers from mood dysregulation. A subject is required to complete Alcohol Consumption survey if a subject indicates that he/she has consumed alcohol during another survey.

The Random Survey is triggered six times at random time periods during the ambulatory assessment of the subject. Random Survey tries to find out if the subject had alcohol since his last survey and his emotional state based on his answer. A scheduling scheme has been employed to make sure that the random surveys are triggered during the time periods where subjects are most active during their ambulatory assessment and to increase the randomness of the survey. The scheme requires a subject to enter his/her start time and end time during which each he/she is most active while using the Body Sensor Application. A service running on the background collects this time periods and calculates the minimum delay the surveys have to maintain between them, before another survey is triggered. This increases the randomness of the survey and ensures that random surveys prompts are noticed by the subject.

3.6 Subject Monitoring Module

The Subject Monitoring Module plays an important role in analyzing the data collected from the external physiological sensors and detecting possible signs of alcohol craving. It is also responsible for sending real-time alerts back to the phone when alcohol craving is detected .The Body Sensor Application receives these real-time alerts and handles them by prompting the subject to fill out a survey to check if he/she has alcohol in past 15 minutes and based on the response of the subject the App prompts the user to fill out surveys which are designed to record his emotional state at that moment .Currently, the Subject Monitoring Module is hosted on a remote Apache Server which consists of PHP scripts which handle the data reception and analysis .The methods to detect alcohol craving will be explained in the implementation section of this report.

4. System Implementation

4.1 Mobile Computing Module

The Body Sensor Application in this project is deployed on Android Operating System [5] which is open-sourced operating system that is used primarily on mobile devices, such as cell phones and tablets. Android Architecture primarily consists of four layers which are built-in android applications, Application or Android Framework, Native Libraries and Linux Kernel. The figure (4) shows the overall android architecture and different components in the architecture. Most android Phones come with built-in android application such as applications to browser the file system, Contacts, applications to set clock and alarm. Android Applications are written in java are translated to Dalvik executable code files before installation on Android Operating System and are executed by Dalvik Virtual Machine [6].Android Framework provides an application with built-in support to access a wide variety of services and libraries which are written mostly in java. The Body Sensor Application uses these service and libraries provided by android framework to access and handle a phone’s resources such as internal sensors, WI-FI, GPS, and Bluetooth. It also handles data sharing and context-switching between different application activities and services running on the Android Operating System. Android Native Libraries are libraries are written in c or c++ language and are specific for a particular hardware. The Linux Kernel is responsible for maintaining interaction with drivers for hardware and providing core functionalities such as, networking, file system access and inter-process-communication.

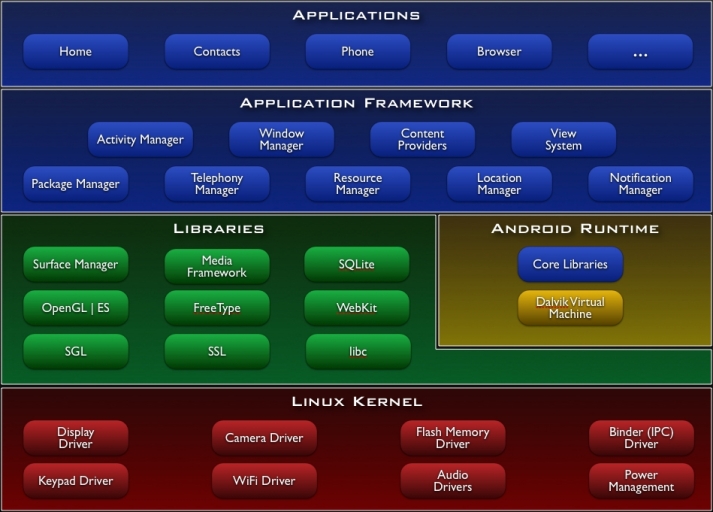
****

Figure (4): Android System Architecture

4.1.1 Networking Module

Networking Module in the Mobile Computing Module is responsible for connecting the Body Sensor Application via Bluetooth with external physiological sensors such as Affectiva Q sensor and Equivital EQ02 Sensor. Networking Module uses the Android Bluetooth API [6] to accomplish the four major tasks necessary to communicate using Bluetooth: setting up Bluetooth, finding devices that are either paired or available in the local area, connecting devices, and transferring data between devices. In order to discover and connect to devices which are already paired or available in the Phone’s Bluetooth range two Activity [7] classes have been setup in the Body Sensor Application. The SensorConnections.java class associated with connections.xml User Interface handles connection requests made by the subject to connect to external physiological sensors. Figure (5) shows the User interface associated with SensorConnections.java class. The DeviceListActivity.java Activity class is triggered via Intent [7] when “Connect to Wrist Sensor” or “Connect to Chest Sensor” is pressed. The DeviceListActivity.java class import’s android.bluetooth package and uses the classes and interface in the package to discover the remote Bluetooth devices by starting discovery service via an instance of Bluetooth adapter class. The DeviceListActivity.java class when triggered displays a list of paired device on its associated device\_activity\_list.xml User Interface by querying the local Bluetooth adapter for paired Bluetooth devices. It also displays remote Bluetooth devices which are not paired with phone when “Scan for Devices” button is pressed. Figure (6) shows the User Interface associated with DeviceListActivity.java class. When a device in the list is selected from the list, the DeviceListActivity.java class is redirected back to SensorConnections.java class with remote or paired device’s name and MAC address.

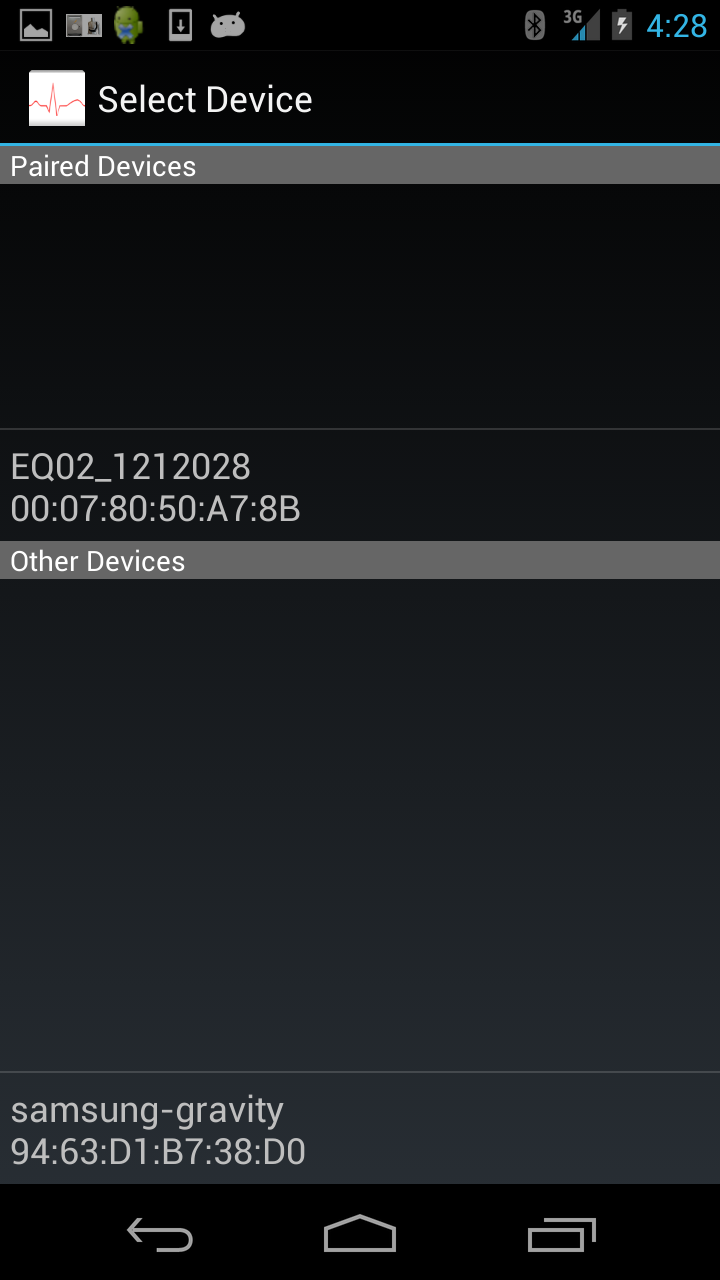
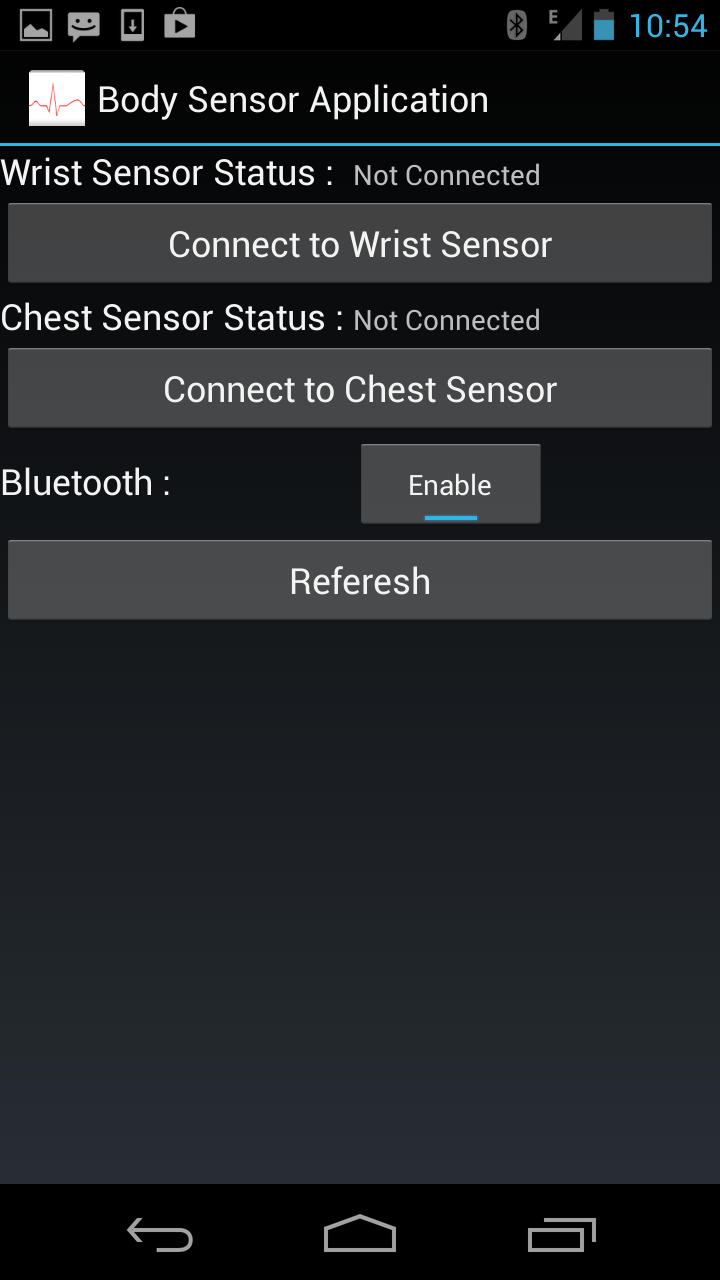
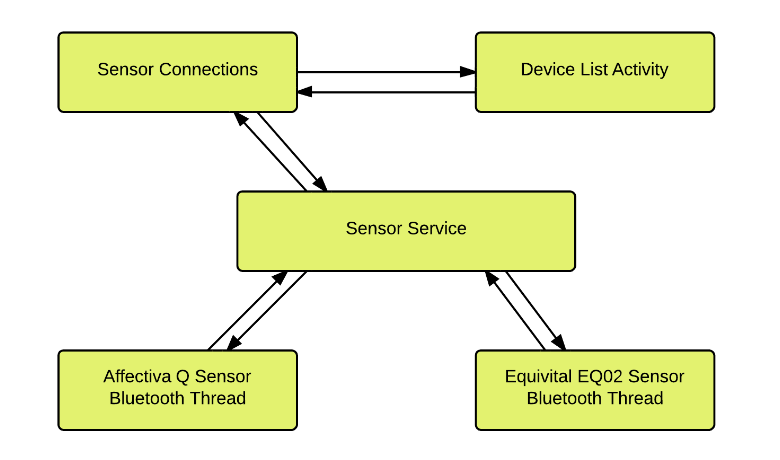


Figure (5): SensorConnections User Interface Figure (6): DeviceListActivity User Interface

The SensorConnections.java class validates if the Bluetooth device selected is Affectiva Q sensor or Equivital EQ02 Sensor and passes the MAC address of the remote device to SensorService.java which is an Android Background Service (the implementation details of Sensor Service will be explained later in the report). One of the issues with Android Bluetooth stack is it doesn’t allow connections to be created on a main thread and in order to handle multiple connections request and multi-threaded system is required. The Sensor Service handles this creating an individual thread for connecting with Affectiva Q and Equivital EQ02 Sensor. The Individual Bluetooth threads use the android.bluetooth package to initialize and maintain Bluetooth data stream via Bluetooth Socket object and they report back their Bluetooth State to the Sensor Service and SensorConnections.java class by broadcasting intent containing their Bluetooth State. Bluetooth State basically tells us about the current status of the Bluetooth connection, there are four main Bluetooth states: None, Listening, Connecting, and Connected. The Sensor Service and SensorConnections.java class have BroadcastRecievers [7] setup to listen to the System Wide Broadcasts including broadcasts made by Individual Bluetooth Threads. The approach followed by individual Bluetooth threads to connect and collect input data from Affectiva Q sensor and Equivital EQ02 Sensor are different; Affectiva Q sensor can be connected by traditional Bluetooth Connection setup where a Bluetooth socket is set up and after connection established via Bluetooth socket the input stream from the socket is parsed and stored. The Equivital EQ02 Sensor Connection Thread unlike the Affectiva Q Sensor Connection thread supports a push-based model and requires additional libraries such as equivital.sdk.connection to initialize and maintain the Bluetooth connection, whenever data is received from the sensor an event is triggered with the data as argument from the Individual thread and this event is handled in Sensor Service by parsing and storing the data.



MAC address

MAC address

Bluetooth State

MAC address

Data, Bluetooth State

MAC address

Bluetooth State

Figure (7): System Flow Diagram of Network Module

4.1.2 Data Collection Module

Data Collection Module is an integral part of Mobile Computing Module; its functionality includes storing data derived from internal sensors, external physiological sensors, Activity recognition service and results generated from surveys. Currently, the data collection module uses a flat file system to store data because flat file system has better time complexity and requires less computational power to store and read data when compared to relational databases like SQLite which requires complex queries to be run on the database to store and read data. Data Collection Module implemented in SensorService.java class, initiates data collection from internal sensors of the phone such Light, Accelerometer and Pressure via an instance of sensor manager class. Data Collection Module uses a push-based model to manage sensors. When the Sensor Service is created the data collection module first registers itself as a receiver to the Android Operating System and at the same time, it specifies a sensitivity level at which it wants to receive sensor data. Then an event is sent to the Data Collection Module when sensor value is changed according to the sensitivity level.

When the Sensor Service is initially created it initiates Activity Recognition Service which is an Intent Service that communicates with one of the built-in Google Play Services which detects the activity or movement of a user .The activity it can detect includes walking, on a bicycle, in a vehicle, still and tilting. Activity Recognition Service communicates with the data collection module to adjust the sampling rate at which location co-ordinates are stored in text files. This is part of a power-saving scheme to save the rate at which GPS is used on the phone. The functionality and implementation details of this scheme will be discussed later in the report.

The data from Affectiva Q sensor and Equivital EQ02 sensor is also sent to the data collection module. Affectiva Q sensor follows a traditional data transmission protocol where it sends a continuous stream of data after the socket is connected. Equivital EQ02 sensor follows the same approach but triggers an event whenever data is received. The Equivital Event Listeners in Sensor Service should subscribe to these events to get data from the sensor which is later parsed and stored in a text file. Data collection module uses File Writer instance to write data; the file used to store data has naming convention where the Data source and Date are used to form the name of each file. Whenever a new data point is given to the data collection module, it determines whether the file to store the data point already exists and if it does the data point is simply appended to the file. Each value in the data point is comma separated and time stamped when it is stored in the file.

4.1.3 Survey Module

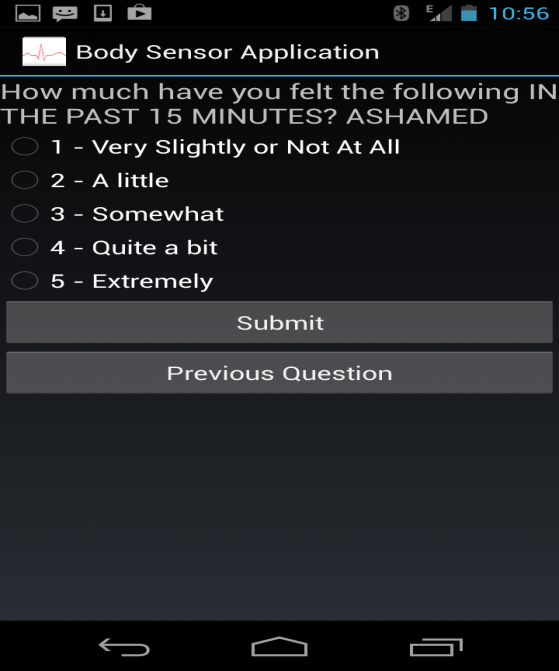
Survey Module is responsible for collecting emotional state of the subject during ambulatory assessment. Currently, the surveys are written in XML and the XML file corresponding to survey is parsed and displayed in a survey format when the survey is started by choosing an option in the survey menu. Figure (8) shows the Survey Menu in the Body Sensor Application.SAX parser which is derived from javax.xml.parsers.SAXParser library is used to parse the XML survey. Currently the application supports the following question types: radio buttons (choose one answer), check boxes (choose multiple answers), numeric questions, test questions and each question type has submit and pervious button. Figure (9) (10) and (11) shows standard survey layout in Body Sensor Application. Questions of the same category in the survey module are currently grouped together into one XML file. Each Survey has questions derived from multiple files which can be randomly ordered if necessitated by the survey. This allows easy modification and extension.

Figure (8): Survey Menu Figure (9): Surveys Question with Radio Buttons

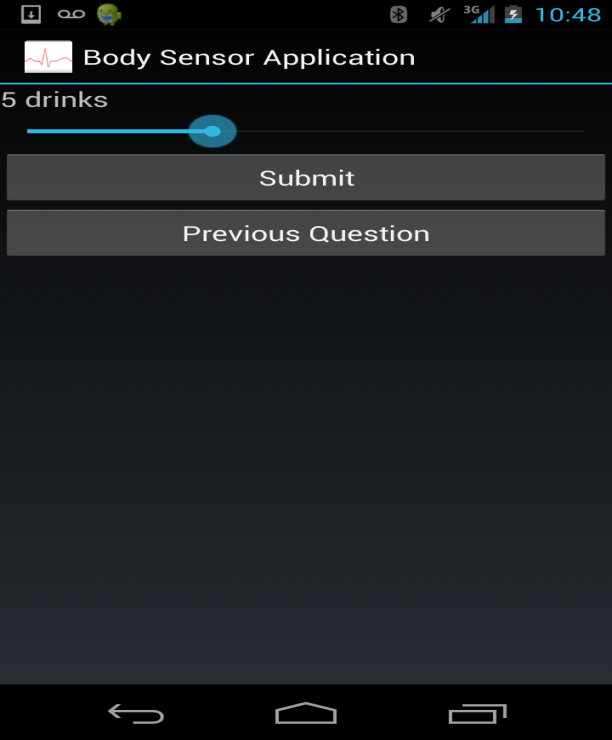
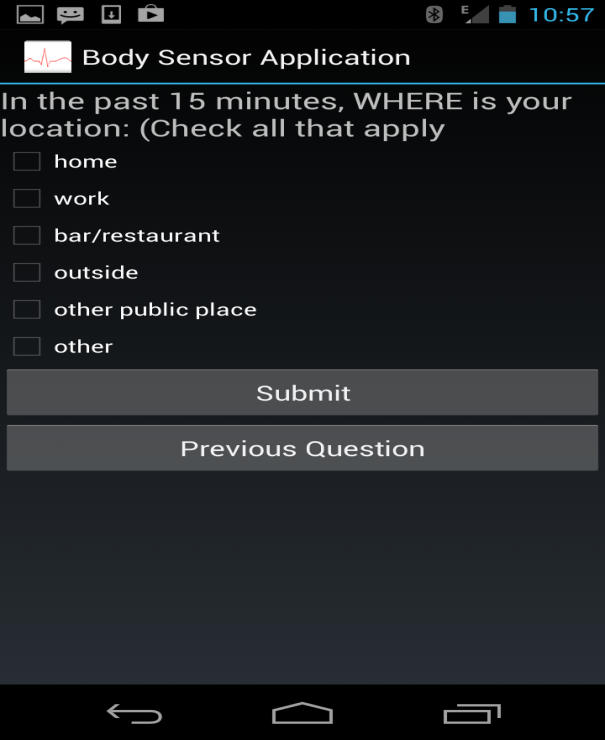


Figure (10): Survey Questions with Check Boxes Figure (11): Numeric Questions

Random Surveys in the Survey Module is triggered at six random time periods during the ambulatory assessment of the subject; it tries to determine if the subject had a drink since his last survey along with his emotional state at that moment. A scheme has been implemented to trigger random surveys to ensure that the surveys are triggered at time periods where the subject is active and to increase the randomness of surveys. The scheme requires the subject to enter a time period where a subject is active, once the ambulatory assessment has started. Figure (12) shows the user interface which shows the status of the random survey and Figure (13) shows the user interface to enter the time periods from the subject.

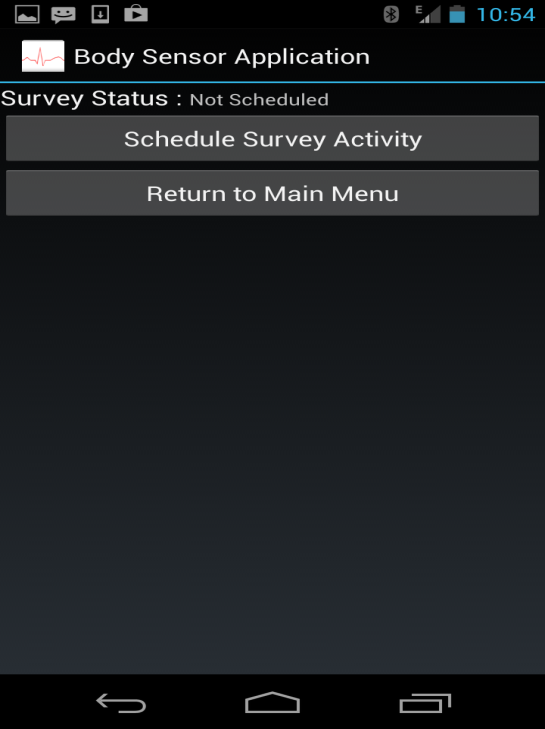
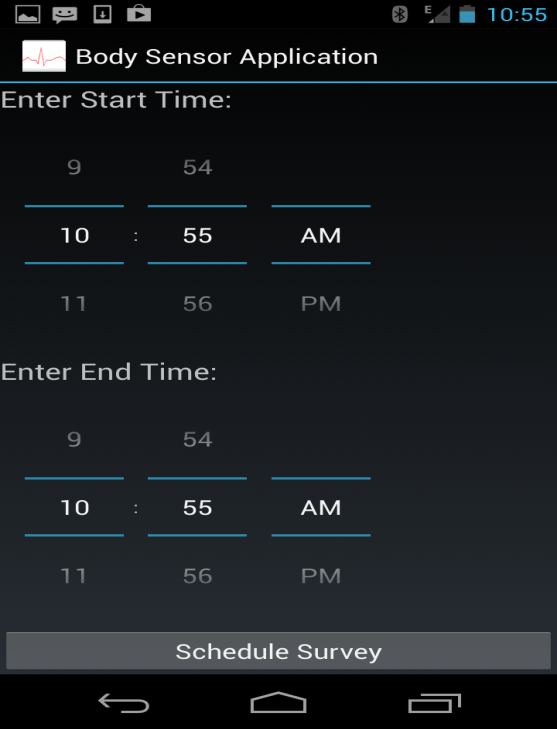


Figure (12): Survey Status User Interface Figure (13): Survey Scheduling User Interface

Random Survey Scheduling Scheme uses an alarmManager Class derived from android.app.AlarmManager package to trigger the surveys. Once the subject enters the start and end time in Survey Scheduler the sensor service running in the background collects the values and triggers the random survey scheduling scheme .Figure (14) shows an example of random survey scheduling scheme ,where the subject choose a time period starting from 8:00 to 10:00 am .Based on the values entered by the subject the scheme calculates the interval time; interval time is time period in which a random survey is triggered and its basically the difference between start and end time divided by six. A minimum delay is determined based on the interval time; this minimum delay time tries increases the randomness of the survey and ensures that random surveys prompts are noticed by the subject. Trigger Time is the period in which a survey is randomly triggered. Equations (1) shows how Minimum Delay, Trigger Interval and Trigger Time are calculated based on Interval Time. In the example below the hour difference between start and end time is 2 hours and interval time is 20 minutes. So, based on the scheme the alarmManager waits for 10 minutes (Minimum Delay) before triggering the survey randomly between 8:10 to 8:20 which is considered the trigger interval.

Delay (10) Trigger Time (Random (1-10)) Delay (10) Trigger Time (Random (1-10))

8:00 Interval (20) 8:20 8:40

Figure (14): Example of Random Survey Scheduling Scheme

Delay=Interval/2

Trigger Interval=Interval-Delay

Trigger Time=Random (1-Time Interval)

Equations (1)

4.2 Subject Monitoring Module

Subject Monitoring Module is mainly responsible for collecting and analyzing data sent from the body sensor application. A PHP script running on a remote Apache Server in the subject monitoring module handles the post request made by the Body Sensor Application. A post request to the Subject Monitoring Module must contain file name and the data associated with the file. Whenever a post request is made, the PHP script determines if the file already exists on the server and if doesn’t, the PHP script creates a new one and appends the data to the file. Subject Monitoring Module has a flat-file system to store data sent from the Body Sensor Application. Alcohol Craving Detection and Analysis part in the Subject Monitoring Module is in development phase and currently the Subject Monitoring Module can detect the fluctuations in EDA (Electro Dermal Activity) values and send real-time notifications back to the phone.

Subject Monitoring Module is also responsible for providing real-time visualization of the physiological data such as ECG (Electrocardiography), EDA (Electro Dermal Activity) and location data sent from the Body Sensor Application. Figure (15) show a graph displaying the EDA values sent from the Body Sensor Application.

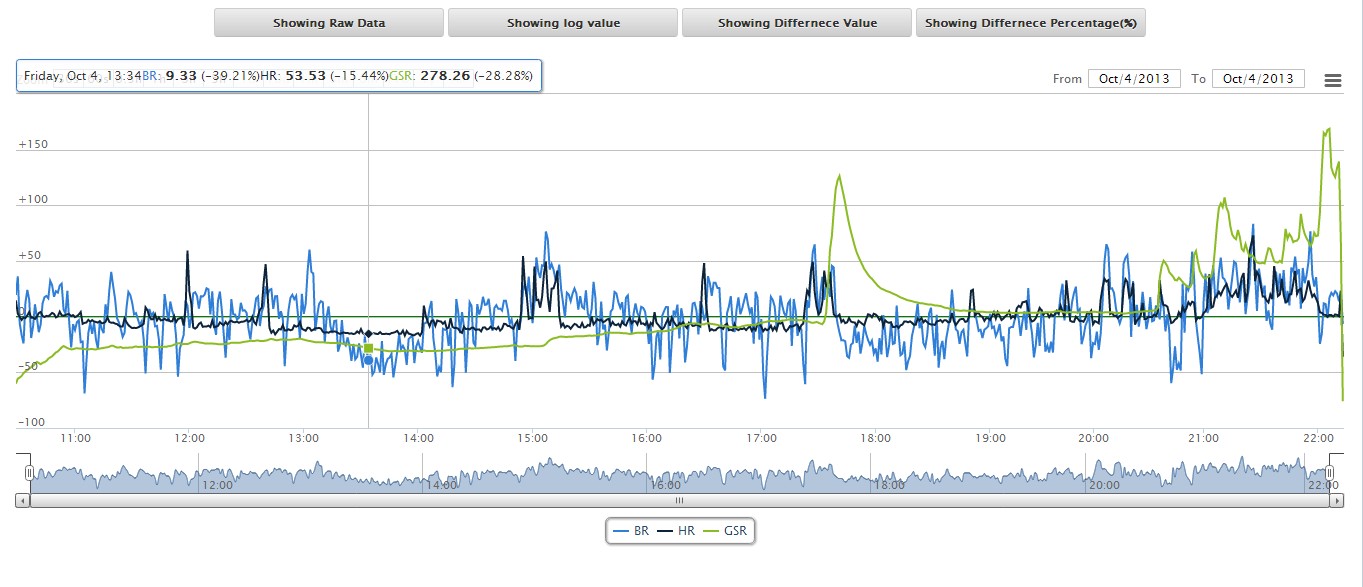


Figure (15): Graph displaying real-time data sent from the Body Sensor Application

5. Implementation Challenges

5.1 Memory Leaks

One of the difficult challenges in implementing the Body Sensor Application was to detect and optimize code to fix Memory Leaks. A Memory Leak [10] occurs when an application repeatedly fails to return memory to the Operating System it has obtained for temporary use causing gradual loss of available System memory. In portable devices such as smart phones were memory is very limited, a memory leak can exhaust the memory allocated to an application over time and force the Operating System to kill the application process. Android Applications are written in java and one of the main reasons for a Memory Leak to occur in object-oriented programming is when an object or resource which is unused is still referenced in the application causing the Garbage Collector not to remove the unused objects which eventually exhaust the memory allocated to the application.

There are several ways to detect a memory leak in an Android Application and one of the efficient ways is to take the snapshot of an Application's Heap Memory called Heap Dump. A Heap is basically the dynamically allocated memory to and Android Application during run-time to satisfy its memory needs. Memory Leaks can also be analyzed by looking at the application log messages which give us information such as amount of memory freed by the Garbage Collector, total memory allocated for the application, amount of memory used by the application and time taken for Garbage Collector to free unused memory . A log message about memory usage of an application typically looks like this “09-16 01:40:57.287: D/dalvikvm (26990): GC\_CONCURRENT freed 402K, 5% free 9261K/9728K, paused 4ms+3ms, total 36ms”.

Android Applications can also use methods such as freeMemory() and totalMemory() from the Runtime class to analyze the runtime memory usage of the application overtime to detect possible memory leaks.

In Body Sensor Application, the Sensor Service running background had Memory Leaks which forced the Android Operating System to kill the Application Process after a short duration which resulted in an Application Crash or Application Restart. This was considered a major problem as the Ambulatory Assessment of a subject using a Smart Phone was expected to run for at least 11 hours while external sensors where connected to phone. Analyzing the Heap Dump of the Body Sensor Application using Memory Analyzer (MAT) [11] was one of initial steps taken to study the Memory Usage of the Body Sensor Application. An increase in usage of Heap Memory was detected over time during the analysis which was considered as sign of Memory Leak. The Heap Dump was further analyzed using Memory Analyzer to see the Shallow Heap and Retained Heap Size of different objects in the Application. Shallow Heap is the size of memory that is held by the object itself and Retained Heap is the size of memory that is freed once the object itself is deleted along with its dependent objects that were made unreachable from Garbage Collector.

The analysis of Shallow Heap and Retained Heap helped to detect and free the unused objects which were holding a reference to major part of the Heap Memory of the Body Sensor Application. One of the memory leaks was caused by an http post thread created by the background Sensor Service. The main functionality of this thread was to store http requests from different threads in the application and execute them if there was more than one request. A list data structure was used to store the requests and every time a request was executed, the pervious requests were not cleared which resulted in Memory Leak caused unused objects piling up in the list data structure without Garbage Collector able to clear them. This Memory Leak was fixed by implementing a network buffer which clears the data structure every time a post request was executed.

5.2 Power Management Strategy

Developing an Android Background Service which significantly uses internal sensors without draining the battery life was one of the challenges in implementing the Body Sensor Application. In order to reduce power consumption from different sensors on the Smart Phone a Power Management Strategy has been implemented in the Body Sensor Application. The Power Management Strategy uses the internal sensors on the phone and several other techniques to reduce the overall power consumption caused by the Body Sensor Application and saves the battery life of the phone during the ambulatory assessment of the subject. One of the techniques that Power Management Strategy uses to reduce the power consumption is by reducing utilization of GPS by the Body Sensor Application. Activity Recognition is a built-in Google play service which has been used in this project to detect the subject's current physical activity, such as walking, driving, riding a bicycle, titling or standing still.

Sensor Service tries to get periodic updates from this Activity Recognition Service and determines whether GPS co-ordinates have to be collected based on the activity value returned from the Service. In order to get periodic updates from the Activity Recognition Service an Activity Recognition Client and an Intent Service is setup in the Application. The ActivityRecognitionScan.java class implements the ConnectionCallbacks, OnConnectionFailedListener interface which contain abstract methods to start stop and set the updates from the Activity Recognition Service. Once the periodic updates start, the Intent Service running in the background collects and handles the periodic updates by sending Activity value and confidence of the prediction to the Sensor Service. Based on these values, the Sensor Service starts and stops the Location Manager to reduce the power consumption by GPS. Currently, the Body Sensor Application collects the GPS values when the activity of subject is walking, driving or riding a bicycle and confidence of the prediction is greater than 75.

5.3 Testing Power Saving Strategy

The Power Management Strategy in the Body Sensor Application has been tested using a Third Party Application called Power Tutor [9].Power Tutor is an application targeted to run Android based mobile platforms; its functionality includes displaying the power consumed by major system components such as CPU, network interface, display, GPS receiver and different applications. The main idea of this application is to allow software developers to see the impact of their design changes on power efficiency. Application users can also use it to determine how their actions are impacting battery life. An initial set of tests have been performed to test different aspects of the Power Management Strategy. The implementation details of these tests are described below.

5.3.1 Test-1

This test was designed to test the power efficiency of Power Management Strategy after activity recognition has been added to it. The Body Sensor Application was run with and without the Power Management Strategy for one hour and Power Tutor application was used to record power metrics during the test. The Graph (1) displays the results from this test.

From Test 1 it was observed that the 3G network system component consumed lot of energy. The Activity Recognition in the Power Management is targeted to reduce the usage of network by reducing the number of times the location co-ordinates were retrieved and sent to the server. Without Activity Recognition, the Body Sensor Application is programmed to retrieve and send the location co-ordinates every 30 seconds. This can be sometimes unnecessary as the user might be idle for long period of time and location co-ordinates recorded and sent to server can be redundant and it might not give us much information about a location.

Activity Recognition Service used in the Power Management Strategy uses low power sensors such as accelerometer to detect if the user is on foot, on a bicycle or in car and record location co-ordinates as opposed to recording location co-ordinates when the user is still. According to Graph (2), this strategy is observed to reduce overall power consumption of the Body Sensor Application.

Graph (1): Displaying energy consumption of individual system components in Test 1

Graph (2): Displaying the Average Power Consumption in Test 1

5.3.2 Test-2

This test was designed to test the power consumption of the Body Sensor Application after buffers where implemented in the network module of the application and to test the responsiveness of the Application. This test was also run for one hour. Graph (3) below displays the energy usage of different system components during this test.

Graph (3): Displaying energy consumption of individual system components in Test 2

From this test, it was observed that the energy consumption of system components such as CPU and 3G significantly increases without buffers because the data points sent from different sensors on the phone and external sensors are sent simultaneously over the network which increases the energy consumption by the network and also significantly increases the CPU usage as the code to send data over the network is executed multiple time which increases the energy consumption of the CPU. Graph (4) shows the average power consumption of the Body Sensor Application over a period of one hour during this test.

Graph (4): Displaying the Average Power Consumption in Test 2

The responsiveness of the Body Sensor Application during the test case without buffers was very slow and it would often trigger Application Not Responding Message. Figure (16) shows the Application Not Responding message triggered when Body Sensor Application was running.

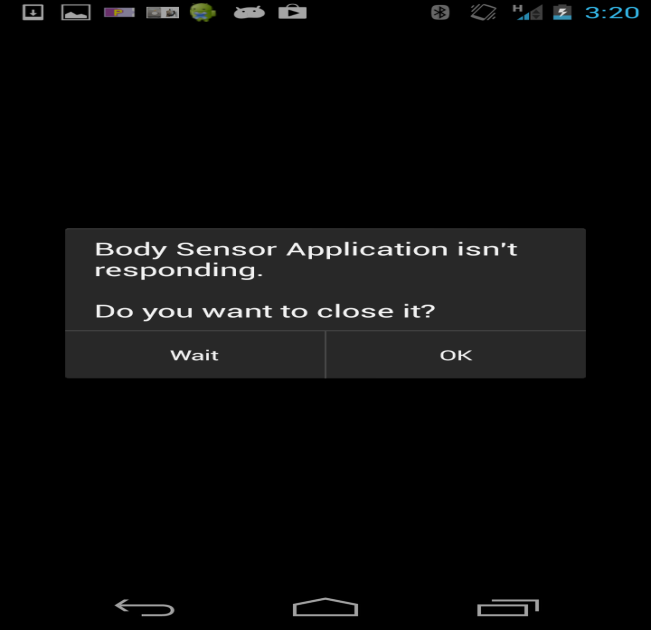


Figure (16): Application Not Responding Message

When an Application is insufficiently responsive for a period of time, the Android Operating System displays a dialog that says your app has stopped responding and offers the user an option to quit the app. One of the reasons for the Android OS to trigger ANR is when an application blocks on some I/O operation (frequently a network access) on a thread so the system can't process incoming user input events. This issue was handled by using buffers which gave sufficient time for an operation to be completed before a new user input was processed in network threads in Body Sensor Application.

6. Results

The Body Sensor Application was successfully tested and deployed on different versions of Android Operating System such as 4.0(Ice cream Sandwich), 4.1, 4.2, 4.3(Jelly bean).During the ambulatory assessment of the subject, the network module in the application has been observed to maintain a reliable connection for 11-12 hrs with the external sensors and during this period the application was able to collect physiological data such as Heart Rate, Breathing Rate, Skin Conductance, Body Position, Body Movement from external sensors and emotional data from a series of manual surveys and random surveys which were answered by the subject. The data from the internal sensors on the smart phone such as Accelerometer and Light was also recorded to observe the movement and ambient settings of the subject in the assessment. The Application was able to send the data collected from the Surveys and Sensors by bundling the data into data packets was sent to the subject monitoring module were the data was stored and visualized in form of graphs.

The Body Sensor Application showed significant improvement from its initial implementation in terms of functionality, performance and reliability. The functionality to connect to multiple external sensors wirelessly by using multi-threaded framework and send the data collected simultaneously to an online server was an important addition to the initial implementation. The performance of Body Sensor Application in terms of power consumption and responsiveness has been improved by implementing Power Saving Strategy and Buffers. The reliability of the application has been greatly improved by detecting and reducing the memory leaks in the application which cause the application to crash and also by implementing reconnection and notification strategies to keep a connection alive during the assessment.

From the data collected from initial assessments, interesting co-relations were observed between GSR (Galvanic Skin Response) and Alcohol Craving. Figure (17) shows the GSR values plotted on Y-axis and timeline of the assessment plotted against X-axis. When every the subject indicated he was drinking or under the presence of an alcohol related cue a significant spike was observed in the GSR values and after subject indicated that he had a drink, a drop in GSR value was observed after few minutes.

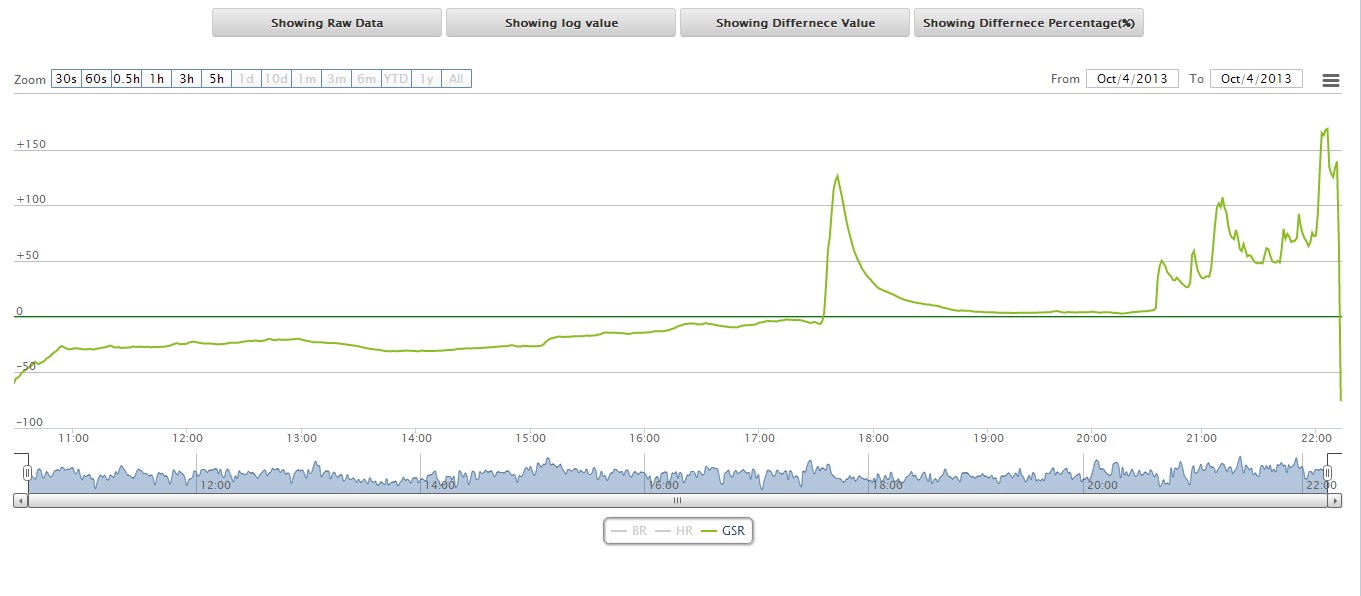


Figure (17): GSR values during the assessment

7. Future Work

Currently, Lab experiments are being performed using the device to establish the ground truth. After the ground truth has been established, predictive models to detect Mood Dysregulation will be included the Subject Monitoring Module in the Body Sensor Project. The functionality to provide real-time interventions to the subject when Mood Dysregulation is detected will also be added to the Body Sensor Application in future implementations.

Context-Aware features such as Geofencing will be added to Body Sensor Application. Geofencing is a built in feature of Google Location Services which combines awareness of the user's current location with awareness of nearby features, defined as the user's proximity to locations. Geofencing is started by initially defining latitude, longitude and proximity for the certain places of interests called Geofences. Whenever a subject enters or exits a Geofence the Location Service can detect the change and perform user defined operation corresponding to the event. An application can have multiple active Geofences at one time. In Body Sensor Application, Geofencing can be used to detect if the subject is entering into a Geofence which has places such as Bars, Dance Clubs where the subject is mostly likely to experience Alcohol Craving, A real-time intervention can provided to the subject whenever he is in proximity to a Geofence.

Security has been an important concern in the current implementation since the subject data transferred between smart phone and server is currently not encrypted. This can pose a privacy concern has confidential survey data answered by the subject is not protected. Currently, steps are being taken to encrypt the data before sending it to the server.

8. Summary

The thesis describes the development of a Context-Aware Body Area Sensing System on Android Platform for detecting Alcohol Craving. The Body Area Sensing System can significantly improve alcohol craving assessments by providing a Context-Aware ambulatory assessment which considers the environmental factors as well as the physiological factors that affect a subject’s emotional state as opposed to traditional approaches where the subject’s emotional state is just assessed by a survey questionnaire where the subject is required to rate his intensity of alcohol craving. In the future, Context-Aware Body Area Sensing System can be extended to detect a variety of mood disorders such as depression, PTSD’s etc. The Body Area Sensing System has been successfully tested and deployed on variety of Android Platforms.

9. References

1. G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, “Towards a better understanding of context and context-awareness,” Proc. 1st Int’l Symposium on Handheld and Ubiquitous Computing, HUC '99, pages 30-307, 1999.
2. Society for Ambulatory Assessment, 2012, [www.ambulatoryassessment.org](http://www.ambulatoryassessment.org).
3. Rajesh, M. ; Dept. of Computer Sci. & Eng., SRM Univ., Vadapalani, India ; Muthu, J.S. ; Suseela, G., “ IPainRelief -A pain assessment and management app for a smart phone implementing sensors and soft computing tools”, Information Communication and Embedded Systems (ICICES), 2013 International Conference 2013.
4. Yuanchao Ma Dept. of Comput. Sci. & Technol., Tsinghua Univ., Beijing, China Bin Xu ; Yin Bai ; Guodong Sun ; Run Zhu , “ Daily Mood Assessment Based on Mobile Phone Sensing”, 2012 Ninth International Conference on Wearable and Implantable Body Sensor Networks.
5. Android Fundamentals, <http://developer.android.com/about/index.html>
6. Android Bluetooth Connectivity,

<http://developer.android.com/guide/topics/connectivity/bluetooth.html>

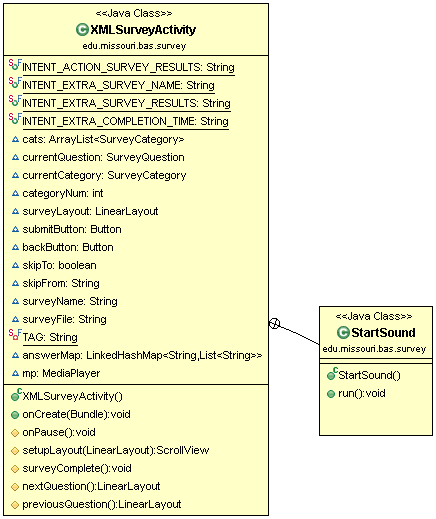
1. Application Fundamentals,

<http://developer.android.com/guide/components/fundamentals.html>

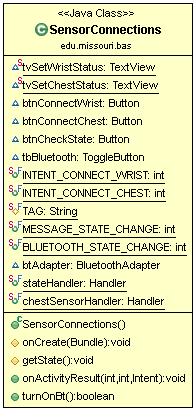
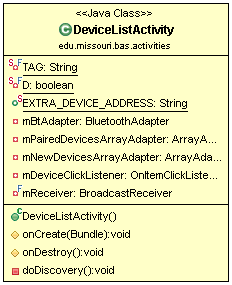
1. Lux, <http://en.wikipedia.org/wiki/Lux>
2. Power Tutor, <http://ziyang.eecs.umich.edu/projects/powertutor/index.html>
3. Avoiding Memory leaks, <http://android-developers.blogspot.com/2009/01/avoiding-memory-leaks.html>
4. Memory Analyzer (MAT), <http://www.eclipse.org/mat/>
5. Affectiva, Q Sensor 2.0 Datasheet, 2012, http://www.affectiva.com/download/Affectiva\_Q\_User\_Manual.pdf.
6. Equivital, EQ02 LifeMonitor Sensor Electronic Module Datasheet, 2012, http://www.equivital.co.uk/assets/common/SEM\_Data\_Sheet\_General\_HIDA3330-DSG-02.2\_.2\_.pdf.
7. Android Sensor Overview, 2012, <http://developer.android.com/guide/topics/sensors/sensors_overview.html>.
8. Android Location Strategies, 2012,<http://developer.android.com/guide/topics/location/strategies.html>.
9. Society for Ambulatory Assessment, 2012, [www.ambulatory-assessment.org](http://www.ambulatory-assessment.org).
10. Android AlarmManager, http://developer.android.com/reference/android/app/AlarmManager.html
11. SAXParser<http://developer.android.com/reference/javax/xml/parsers/SAXParser.html>
12. P. Baskett, Y. Shang, M. Patterson, T. Trull. Towards a System for Body-Area Sensing and Detection of Alcohol Craving and Mood Disregulation. In publication, IEEE Consumer Communications and Networking Conference (CCNC) 2013 Demos.
13. K. Plarre et al.,”Continuous inference of psychological stress from sensory measurements collected in the natural environment,” Proc. 10th Int’l Conference on Information Processing in Sensor Networks (IPSN), pp.97-108, 2011.
14. E.W. Boyer, R. Fletcher, R.J. Fay, D. Smelson, D. Ziedonis, and R.W. Picard, “Preliminary efforts directed toward the detection of craving of illicit substances: the iHeal project,” J Med Toxicol. 8(1):5-9, March 2012.

**Appendix: Class Diagrams**

**Xml Survey Activity**

****

**Sensor Connections Activity Device List Activity**

****