An Enhanced MOBILE AMBULATORY ASSESSMENT SYSTEM FOR ALCOHOL CRAVING STUDIES

A Thesis

Presented to

The Faculty of the Graduate School

University of Missouri-Columbia

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

By

Ruiqi Shi

Dr. Yi Shang, Thesis Supervisor

APRIL 2015

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

AN ENHANCED MOBILE AMBULATORY ASSESSMENT SYSTEM FOR ALCOHOL CRAVING STUDIES

Presented by Ruiqi Shi

A candidate for the degree of Master Science

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

Dr. Yi Shang

Dr. Timothy Trull

Dr. Dong Xu

**Acknowledgements**

First of all, I would like to express my sincere gratitude to my advisor Dr. Yi Shang for his continuous support and valuable advices not only in this thesis but also throughout my senior year and graduate study. His broad scientific knowledge and research passion are my greatest motivation. It is my great honor to learn from him.

I would also like to thank Dr. Timothy Trull and Dr.Dong Xu for reviewing my thesis and providing so many insightful and valuable comments and suggestions on this thesis. I appreciate them taking precious time out of their busy schedule to serve on my thesis committee. Additionally, I would like to thank Dr. Timothy Trull for his continuous support for me and suggestions to my several projects.

I would also like to thank my colleagues in my lab, they have always been very helpful and it is my greatest pleasure to work with them. Our friendship and memory together are the most valuable treasure in my life.

Finally, I would like to thank my family and friends for their constant support and encouragement during all these years. Without their understanding, it would be impossible for me to go through all the difficult time.

**TABLE OF CONTENTS**

[**Acknowledgements** ii](#_Toc417399277)

[1 . Introduction 1](#_Toc417399278)

[2 . Related Work 4](#_Toc417399279)

[3 . System Design 6](#_Toc417399280)

[3.1 External Sensor Module 7](#_Toc417399281)

[3.2 Mobile Computing Module 8](#_Toc417399282)

[3.2.1 Networking Module 9](#_Toc417399283)

[3.2.2 Survey Module 9](#_Toc417399284)

[3.2.3 Account Management Module 11](#_Toc417399285)

[3.2.4 Data Collection Module 11](#_Toc417399286)

[3.2.5 Phone Monitoring Module 12](#_Toc417399287)

[3.2.6 Scheduling Module 13](#_Toc417399288)

[3.2.7 Background Service Module 13](#_Toc417399289)

[3.2.8 Utility Module 14](#_Toc417399290)

[3.2.9 Data Recovering Module 14](#_Toc417399291)

[3.3 Server Subject Monitoring Module 14](#_Toc417399292)

[3.3.1 Data Processing Module 15](#_Toc417399293)

[3.3.2 Administrator Management Module 15](#_Toc417399294)

[3.3.3 Data Demonstration Module 15](#_Toc417399295)

[3.4 Data Analysis Module 16](#_Toc417399296)

[4 . System Implementation 18](#_Toc417399297)

[4.1 Mobile Computing Module 18](#_Toc417399298)

[4.1.1 Android Architecture 18](#_Toc417399299)

[4.1.2 Android Main Components 20](#_Toc417399300)

[4.1.3 Networking Module 22](#_Toc417399301)

[4.1.4 Survey Module 26](#_Toc417399302)

[4.1.5 Account Management Module 30](#_Toc417399303)

[4.1.6 Data Collection Module 32](#_Toc417399304)

[4.1.7 Phone Monitoring Module 34](#_Toc417399305)

[4.1.8 Scheduling Module 36](#_Toc417399306)

[4.1.9 Background Service Module 39](#_Toc417399307)

[4.1.10 Utility Module 39](#_Toc417399308)

[4.1.11 Data Recovery Module 40](#_Toc417399309)

[4.2 Server Subject Monitoring Module 41](#_Toc417399310)

[4.2.1 Data Processing Module 41](#_Toc417399311)

[4.2.2 Administrator Management Module 42](#_Toc417399312)

[4.2.3 Data Demonstration Module 43](#_Toc417399313)

[4.3 Data Analysis Module 46](#_Toc417399314)

[4.3.1 Pre-processing Module 47](#_Toc417399315)

[4.3.2 Graphic Module 50](#_Toc417399316)

[4.3.3 Sampling Module 51](#_Toc417399317)

[4.3.4 Supervised Learning Module 51](#_Toc417399318)

[4.3.5 Preliminary Result 58](#_Toc417399319)

[5 . Future Work 62](#_Toc417399320)

[6 . Summary 63](#_Toc417399321)

[References 64](#_Toc417399322)

**LIST OF FIGURES**

[*Figure 1.1: Android market share from 2011 to 2014* 3](#_Toc417399359)

[*Figure 3.1 Overall System Architecture* 6](#_Toc417399360)

[*Figure 3.2 Equivital EQ02 Life Monitor* 8](#_Toc417399361)

[*Figure 4.1 Android System Architecture* 19](#_Toc417399362)

[*Figure 4.2 Sensor Connection Activity* 22](#_Toc417399363)

*Figure 4.3 Device List Activity..........................................................................................*.22

[*Figure 4.4 Flow Diagram of the Bluetooth Management* 23](#_Toc417399364)

[*Figure 4.5 Flow Diagram of HTTPS implementation in Android* 23](#_Toc417399365)

[*Figure 4.6 Comparison with HTTP, partially HTTPS and HTTPS* 25](#_Toc417399366)

[*Figure 4.7 Question Sample* 26](#_Toc417399367)

[*Figure 4.8 Question XML File* 26](#_Toc417399368)

[*Figure 4.9 Question Layout* 27](#_Toc417399369)

[*Figure 4.10 Survey Menu Activity* 28](#_Toc417399370)

[*Figure 4.11 Alert dialog for the Survey* 29](#_Toc417399371)

[*Figure 4.12 Admin Management Layout* 31](#_Toc417399372)

[*Figure 4.13 Subject Identity Verification Layout* 31](#_Toc417399373)

[*Figure 4.14 Hybrid Cryptosystem Implemented in the System* 34](#_Toc417399374)

[*Figure 4.15 Statistics file for the application and phone performance* 36](#_Toc417399375)

[*Figure 4.16 Suspension/Break Suspension Flow Dialog* 38](#_Toc417399376)

[*Figure 4.17 one example of data recovery* 40](#_Toc417399377)

[*Figure 4.18 Login Page* 42](#_Toc417399378)

[*Figure 4.19 Admin Management Page* 42](#_Toc417399379)

[*Figure 4.20 File Check Webpage* 44](#_Toc417399380)

[Figure 4.21 *An example of a sequence of time-stamped subject locations* 44](#_Toc417399381)

[*Figure 4.22 An example of sensor and survey data graphs displayed on the server* 45](#_Toc417399382)

[Figure 4.23 *An example of CPU status of App Used and Overall Used* 45](#_Toc417399383)

[*Figure 4.24 Pipeline of the Data Analysis Module* 47](#_Toc417399384)

[*Figure 4.25 A drinking sample collected by real subject, from around 19:50 p.m. to 21:40 p.m. The four measurements are breath rate, hear rate, ambulation (1- stationary, 2-slow moving, 3 fast-moving), and skin temperature. Each blue vertical bar represents a time that the patient reported a drink in drinking follow-up surveys.* 50](#_Toc417399385)

[*Figure 4.26 Flow Chart of Supervised Learning Module* 52](#_Toc417399386)

[*Figure 4.27 An example of drinking period generation step in “Expansion” method* 53](#_Toc417399387)

[*Figure 4.28 An example of conversion step in “Expansion” method* 54](#_Toc417399388)

[*Figure 4.29 An example of shrink step in “Shrink” method* 56](#_Toc417399389)

[*Figure 4.30 An example of conversion step in “Shrink” method* 56](#_Toc417399390)

[*Figure 4.31 Deviation result of “Expansion” method* 60](#_Toc417399391)

[*Figure 4.32* *Deviation result of “Shrink” method* 60](#_Toc417399392)

**LIST OF TABLES**

*Table 4‑1 Confusion matrix of the prediction result based on the raw dataset..*.............59

*Table 4‑2 Confusion matrix and deviation of “Expansion” and “Shrink” methods*…….....60

**ABSTRACT**

Currently, most methods in clinical psychology research primarily rely on questionnaires and interviews with examiners instead of providing real-life subject behavioral and psychology data monitoring and collecting services. This thesis presents an android-OS-based mobile Ambulatory Assessment System, for psychology research -- especially alcohol craving studies – to improve current methods and provide real-time data monitoring, collecting and processing. As current generation smartphones provide more powerful communication platform, embedded with robust built-in sensor suits and vigorous processing and storage capabilities, smartphones play an increasingly significant role in various sensing tasks such as activities monitoring personal health surveillance and environment. This system consists of four parts: a wearable sensor (Equivital EQ2 sensor) that measures physiological data, an Android smartphone, a web server and a data analysis module. The smartphone is responsible for collecting and recording physiological data from the wireless wearable sensor, interacting with the users to conduct various surveys, and uploading data to the web server. The server is responsible for data processing, computation, and visualization. Utilizing machine learning methods, data from the sensor and survey build models that predict how various psychological disorders cause alcohol or other substance cravings and emotion dysregulation. The system has been deployed in a field study of alcohol craving, and initial data collected from real subjects proved promising.

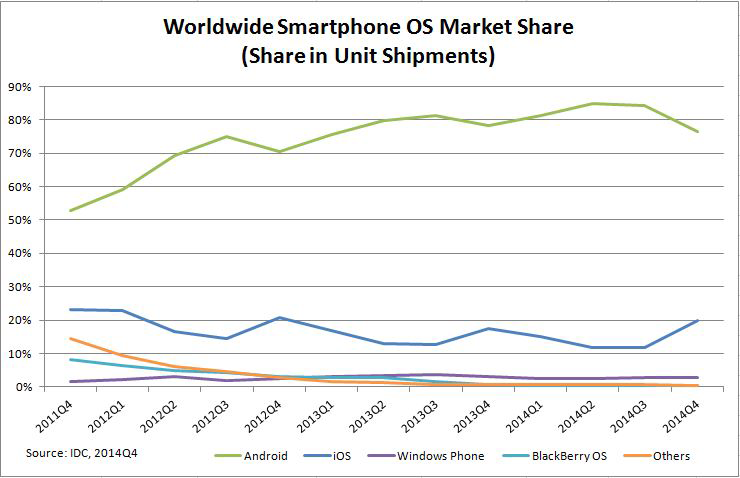
# . Introduction

Ambulatory assessment is the use of field methods to analyze subjects in a natural or unconstrained environment [1]. Mood dysregulation refers to abnormally intense and frequent experience of negative emotions. Currently, most methods in clinical psychology research primarily rely on questionnaires and interviews with examiners, which could not provide real-life subjects with behavioral and psychological data monitoring and collecting services. By combining environment information, physiological information, and information about participants’ mental states collected through system-generated, self-reported surveys, machine learning models can be built to identify changes in mood or substance cravings. The same information is also useful for context aware applications. Context aware computing proposes the use of contextual information about a user to change the way applications behave. 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 [2,3]. By combining methodology from psychophysiological field research with body area wireless sensor networks and mobile devices, context aware computing can be improved by taking into account behavior and emotion information. Applying concepts of context awareness to psychophysiological data will also improve models for predicting psychological state by taking into account the subject’s context and the effects of that context on sensor readings.

Current generation smartphones provide robust sensor suites including accelerometers, GPS, microphones, light sensors, and cameras, etc. These sensors allow smartphones to be used for various sensing tasks such as activity monitoring, personal health, and environment monitoring. Smartphones also provide a powerful communication platform as they are generally equipped with Wi-Fi, Bluetooth, and a 3/4G data connection. Smartphones are extremely portable and have powerful processing and storage capabilities. In addition, many physiological sensors come equipped with Bluetooth. They can be connected to a smartphone wirelessly to extend its sensing capabilities.

This paper presents an enhanced mobile ambulatory assessment system designed for clinical studies of mood dysregulation, craving, and alcohol use using ambulatory assessment. The system is developed around a smartphone, which is responsible for collecting body sensor data wirelessly, collecting its own on device sensor data, interacting with the user and conducting surveys, communicating with a server for uploading encrypted data and receiving instructions and a data analysis system for model construction. The system is built on a previous existing prototype which could provide data collection and simple data communication between the phone and server, yet, with low performance. For this enhanced system, more modules and function are provided with satisfying performance. Data collected by the smartphone is uploaded to the server either in real-time or when an Internet connection is established. The server will decrypt the data and prepare the data for plotting. Once data is collected from the server, the data analysis module will pre-process the data, plot the smoothed graph, generates training samples and uses supervised and unsupervised learning method for model construction.

Due to the high percentage of global operating system market share and open source nature of Android SDK (Software Development Kit), Android Operating System was chosen as the platform to develop and deploy the mobile ambulatory assessment system for the mobile device in this thesis.



*Figure 1.1: Android market share from 2011 to 2014*

# . Related Work

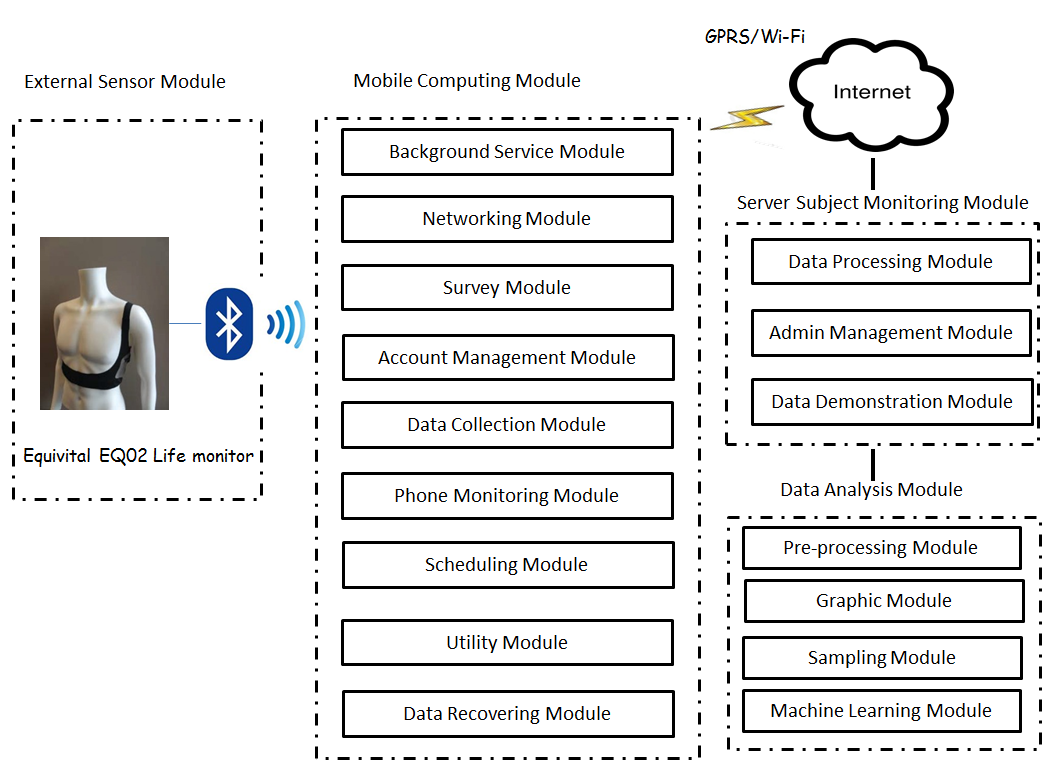
Wireless body-area sensor networks have been considered a hot topic and used for a variety of applications in mobile health, physiological monitoring, and context aware computing. The proposed system in this paper is closely related to two projects presented in [4] and [5] as they all use continuous monitoring of biosensors along with periodic self-report to predict psychological states. In [4], a system was developed for continuous identification of stress, using self-reporting and external sensors to identify stress. Sensors used include ECG, respiration, 3-axis accelerometer, and skin and ambient temperature sensors. Up to 90% accuracy was obtained using data collected in lab experiments where stress was induced. In [7], the iHeal project uses a biosensor that measures electro dermal activity, motion, temperature, and heart rate to attempt to identify substance cravings. When the system detects a change in sympathetic nervous system activity, it collects information from the biosensor and self-reported information from the user about stress, cravings, and activities. Both projects demonstrated that biosensors combined with self-reporting can be used to train models to predict mental state.

Self-assessment of emotion (e.g. surveys) provides important, yet oftentimes inaccurate information [6]. In lab experiments in [4], users correctly self-assessed their own stress only 84% of the time. In addition to incorrect self-assessment, emotions aren’t necessarily experienced in a binary way. For example, a person can experience different degrees of happiness at different times. Another problem with self-assessed psychological information in a natural environment is that unlike in a laboratory experiment, there is little control over the participants’ physical and social environment, which makes identifying the participants’ contexts a critical task [7]. Certain environmental and physical conditions can have an impact on measurements important to the classification of mood and emotion [6, 7]. These factors may include physical activity, posture, and social setting [8].

# . System Design

The proposed system in this paper consists of four important modules, shown in Figure 3.1, including: External Sensor Module, Mobile Computing Module, Server Subject Monitoring Module and Data Analysis Module.

The Mobile Computing Module can be further divided into Networking Module, Survey Module, Account Management Module, Data Collection Module, Phone Monitoring Module, Scheduling Module, Background Service Module, Utility Module and Data Recovering Module.



*Figure 3.1 Overall System Architecture*

The Server Subject Monitoring Module can be further divided into Data Processing Module, Administrator Management Module and Data Demonstration Module.

The Data Analysis Module can be further divided into Pre-processing Module, Graphic Module, Sampling Module and Machine Learning Module.

## External Sensor Module

External Sensor Module mainly utilizes one certain physiological sensor: Equivital EQ02 Life Monitor, as shown in Figure 3.2. The sensor is responsible for recording and transmitting the subject’s physiological state to the Mobile Computing Module.

The Equivital EQ02 Life Monitor 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 through Bluetooth. The EQ02 Life Monitor can provide Heart Rate, Breathing Rate, Body Position, Body Movement, Core Temperature, Skin Temperature, Accelerometer Data and Oxygen Saturation in Body.[9] The EQ02 Life Monitor transmits data via Bluetooth and uses RFCOMM protocol to transmit data to the Mobile Computing Module. Additionally, EQ02 Sensor transmits encoded data which can be decoded only by using the 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 the SEM. Primarily, the decoder uses a publish/subscribe model; the application in the mobile computing module subscribes to one or more messages in which the researcher is interested, and the decoder will notify any subscribed client when data become 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.



*Figure 3.2 Equivital EQ02 Life Monitor*

## Mobile Computing Module

Mobile Computing Module mainly consists of a smart phone application called “Craving Study App”. Craving Study App is a context aware mobile which is developed to operate on a portable device with the Android Operating System. The App is composed of nine modules: Networking Module, Survey Module, Account Management Module, Data Collection Module, Phone Monitoring Module, Scheduling Module, Background Service Module, Utility Module and Data Recovering Module. The functionality of these modules is explained as follows and implementation of these modules is discussed in later sections of this thesis.

### Networking Module

Networking Module is responsible for both the Bluetooth connection and cellular/Wi-Fi networking. The main functionality of the Bluetooth task is initializing, maintaining and reporting Bluetooth connection state. In terms of the cellular/Wi-Fi networking task, Networking Module transmits securely the data collected from the Data Collection Module to the remote server through the Hypertext Transfer Protocol Secure (HTTPS) protocol.

### Survey Module

Considering that mobile ambulatory assessment system may require different sets of survey items, the resource file of Survey Module is designed to be created and modified in a convenient manner. Thus, the surveys are written in XML, which is parsed by the application when a survey is started. For the current Android Application “craving study app” dealing with alcohol craving and mood dysregulation, six surveys are designed, divided into two categories: self-reported survey and system-triggered survey.

Survey Module is composed of a series of self-reported surveys and system-triggered survey. These surveys are designed to record and analyze the subject’s drinking behavior and emotion state during the ambulatory assessment. The self-reported surveys, all manually triggered, are divided into three categories: the Initial Drinking survey, the Mood Dysregulation survey and the Craving Episode survey. The subject is required to take the Initial Drinking Survey after taking their first drink. The Mood Dysregulation survey is required to be filled out when the subjects suffers from sudden mood dysregulation. Craving Episode survey is required to be completed when the subject experiences an alcohol craving.

The system triggered survey also includes three questionnaires: the Morning Report Survey, the Random Survey, and the Drinking Follow-up Survey. The trigger mechanism is maintained by the Scheduling Module, which is discussed later in the thesis.

The Morning Report Survey is scheduled to be prompted based on the most recent wake-up time set by the subject. If the Morning Report Survey is not completed by the subject during his preset wake-up time, it could also be manually triggered before 12:00 P.M. of the same day.

Between the subject’s set-wake-up time and midnight, the Random Survey is triggered six times at random. The Random Survey is designed to retrieve the drinking scenario and emotion state of the subject’s daily life. Additionally, some restrictions are applied when scheduling the interval between random surveys to make sure the Random Survey monitors the subject’s entire day.

The Drinking Follow-up Survey is triggered three times in every 30 minutes after the subject reports drinking behavior in the Initial Drinking Survey, Random Survey or even a previous Drinking Follow-up Survey. Whenever a new drinking behavior is reported, the schedule for the Drinking Follow-up Survey trigger time will be updated and reset.

Also, the system provides a survey completion reminder, which is a combination of vibration and phone ringtone that go off three times if the survey is not completed within a certain timeframe. If the subject still fails to complete the survey after three reminders, the survey will be cancelled by the system. Also, for the system triggered survey, there will be a vibration and ring tone alarm to remind the subject to start the survey. Each survey’s completion state and completion time is recorded by the system to help the researcher obtain more information on the subject’s survey completion behaviors.

### Account Management Module

The Mobile application is mainly operated by two groups of users -- the administrator group and the subject group. Thus, Account Management Module is composed of two parts: one is for the administrator to set the account ID, which serves as the identifier for different subjects. This module will cooperate with the Networking Module to communicate with the server to add or verify the subject’s account on the server. The module’s other function is allowing the subject to set or verify his PIN number, which is designed to verify the subject’s identity, before the survey is filled, in case the survey is accidently taken by someone other than the subject himself. This increases the credential of self-report survey.

### Data Collection Module

Data Collection Module is responsible for retrieving data and maintaining the backup data. To retrieve data, Data Collection Module starts different threads to collect different kinds of data including external sensor data, internal sensor data, survey data and phone state data. The external sensor data includes all the data retrieved from the EQ02 life monitor such as Heart Rate, Breathing Rate, Heart Rate Confidence, Skin Temperature, Breathing Rate Confidence, Body Position and Body Movement. The internal sensor data includes all the data retrieved from the sensor embedded in the phone. In this system the internal sensor data include Accelerometer, Location. Phone state data, which is mainly phone performance data and subject operation data, is retrieved from the Phone Monitoring Module, which this thesis discusses later. To maintain the backup data, the system first applies a cryptosystem, which combines the convenience of an asymmetric public-key cryptosystem with the efficiency of a symmetric-key cryptosystem, to conjure the security data ready to be transmitted to the server, and keep the backup file in encrypted code. Then, the Data Collection Module stores the backup data in flat file system rather than in local database system, as flat file system are less complex and more robust than the local SQLite database in terms of reading and writing large amount of data and querying, inserting and updating operations.

### Phone Monitoring Module

Phone Monitoring Module helps researchers monitor significant phone behaviors such as shut-down and boot state and device performance such as device battery, networking state, and memory and CPU usage. By default, the Phone Monitoring Module records the state in which the phone is shut-down or booted and documents the phone’s battery level and connection state every five minutes. Also, the Phone Monitoring Module provides an advanced mode to detect the networking traffic, memory usage and CPU usage of the Application.

### Scheduling Module

The Scheduling Module is responsible for scheduling or canceling different surveys and alarms and restarting the Application from the system call after phone is booted. The Scheduling Module mainly uses Alarm Manager and the broadcast receiver component in Android Operating System to schedule, cancel, suspend or re-schedule all the timer-related tasks. The scheduled surveys, which consist of Morning Report, Random Survey and Drinking Follow-up Survey, have already been explained in this paper’s discussion of Survey Module. The alarms include the charging reminder, Bluetooth disconnection reminder and survey incompletion reminder. The Scheduling Module also provides a suspension function that allows the subject to disable all alarm reminders in circumstances when the subject is involved in meetings or other events that call for minimal disruption.

### Background Service Module

Background Service Module is the daemon component of the whole Mobile Computing Module. It is mainly responsible for three tasks: maintaining communication of different modules, preparing the dynamic broadcast receiver and starting separate threads for performing different tasks. Background Service Module can perform long-running operations in the background even when the subject switches to another application.

### Utility Module

Utility Module is the module that defines a set of methods which perform common, often re-used functions, a series of static variables which never change, resource files, and the subject’s preference information. The resource files include all the display resource such as display text, sound, photo, and the static broadcast receiver registration. The subject’s preference information include the all application settings defined or updated by the subject, such as wake-up time, subject ID and PIN number.

### Data Recovering Module

Data Recovering Module is responsible for recovering all the collected data from the phone’s local encrypted text files created and maintained by the Data Collection Module. This Module is called when the data stored on the remote server is replaced or deleted due to some unexpected operations. In this way, Data Recovering Module improves the data reproducibility for this ambulatory assessment system in some abnormal circumstance.

## Server Subject Monitoring Module

The Server Subject Monitoring Module, built on a Linux remote server with a MYSQL Database, is composed of three modules: Data Processing Module, Administrator Management Module and Data Demonstration Module. This Module supports data decryption, processing, storage and data visualization on the web.

### Data Processing Module

The Data Processing Module is performing three tasks sequentially: Data Decryption, Data Processing and Data Storing. The designed flow list is as follows: After the server received the encrypted posted data from the phone, this Module calls a decryption program to decrypt the file. Next, Data Processing Module on the server processes the raw data to produce human-readable data and other types of data. At last, this module stores the data into flat file system with the file named through the subject ID, data generation time and the survey category of the data.

### Administrator Management Module

The Administrator Management Module provides for the researchers with two functions: administrator login function and subjects manipulating function. The login function applies a name-password pair to prevent unauthorized user from accessing the research data. The subjects manipulating function verifies the subject’s identity through collaboration with the Mobile Computing Module. Once the subject identity is verified, the administrator is allowed to perform various tasks including assign, maintain and remove subject information. All of the data is stored in the MYSQL Database installed on the remote server.

### Data Demonstration Module

Data Demonstration Mode is responsible for providing online file reading and downloading as well as real-time graph illustration for the researchers. There are eleven categories of files: application status files, which store the scheduling information of the application; accelerometer files; physiological files retrieved from the external sensors; six types of survey files; emotion score files, which is computed from different survey files to show emotion state of the subject; phone state files, which show the phone battery level. In terms of graph illustration, three kinds of graphs are available: subject location map, subject physiological graph and phone state graph. All graphs are in real-time mode and designed with an interactive interface to help the researcher zoom in or out the graph, with the choice of hiding or displaying different items on the data graph.

## Data Analysis Module

The main purpose of the Data Analysis Module is to estimate from sensor data the subject’s intensity of alcohol craving. Machine learning methods are applied to construct models from field data. The data can be annotated either by the participants’ self-assessments or later by analyzing the data, and thus both supervised and semi-supervised learning methods can be applied to generate predictive models. In addition, machine learning methods can be applied on the identification of various features of the user’s context, using data collected through available sensor data and emotion self-reporting. Participants’ activities, emotional states, emotion dysregulation, alcohol use and craving, environment, and other physical state information can all be investigated and analyzed.

The Current Data Analysis Module consists of four modules which serve as the sequence pipeline: Pre-processing Module, Graphic Module, Sampling Module and Machine Learning Module. Preprocessing Module is responsible for cleaning and assembling the raw data and fitting missing data. Graphic Module mainly performs curve fitting/smoothing to extract global trend. Sampling Module serves mainly to generate training samples by sampling and feature calculation. The Machine Learning Module is responsible for running supervised and unsupervised machine learning/statistical method on the data for future model construction.

# . System Implementation

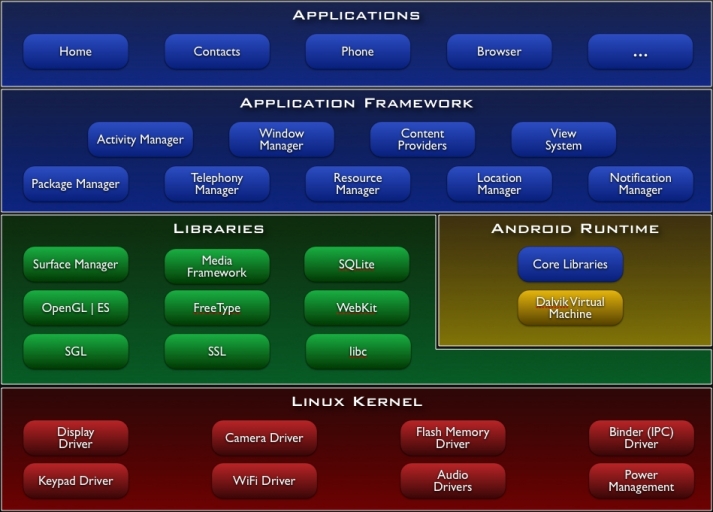
This chapter details the implementation of this mobile ambulatory assessment system for alcohol craving study. The Mobile Computing Module is implemented in JAVA language with the Android Framework and XML. The Server Subject Monitoring Module is implemented in PHP, JavaScript, HTML, CSS, and SQL languages. The Data Analysis Module is implemented in JAVA and MATLAB languages.

## Mobile Computing Module

In this section, Android System Architect and Android Main Components are first introduced. Then the implementation details of each sub-module of Mobile Computing Module are delineated.

### Android Architecture

The Mobile Computing Module, referred to as “craving study app” in the previous section, is implemented and deployed on Android Operating System, which is an open source, multi-user Linux system having been used on Android Wear, phones, tablets, Android TV and Android AUTO [10]. Figure 4.1 shows the overall Android Architecture and different components in the architecture. The Android System Architecture is composed of four layers: Linux Kernel, Android’s native libraries, Application Framework and Application. The Linux Kernel layer is the base layer responsible for maintaining interaction with hardware drivers. This layer provides basic system functionality like process and memory management, as well as management of devices such as



*Figure 4.1 Android System Architecture*

camera, keypad, and display. It is this Linux that interacts with the hardware and contains all the essential hardware drivers. Drivers are programs that control and communicate with the hardware [11]. The next layer is the Android’s native libraries. It is a layer that enables the device to handle different types of data. These libraries are written in C or C++ language and are specific for a particular hardware [11]. For example, SQLite is the database engine used in android for data storage purposes. Android Runtime consists of Dalvik Virtual machine and Core Java libraries.Dalvik Virtual Machine is a type of JVM used in android devices to run apps and is optimized for low processing power and environment with low memory. Unlike the JVM, the Dalvik Virtual Machine doesn’t run .class files, instead it runs “.dex” files. “.dex” files are built from .class file at the time of compilation and provide higher efficiency in low resource environments. Core libraries enables Android application developers to write Android applications using standard Java programming language [11]. The third layer is Application Framework Layer, which provides built-in support to access a wide variety of services and libraries written mostly in java. The Mobile Computation Module uses these services and libraries provided by Google Android Framework to manage the phone’s basic functions such as resource management, internal sensor data collection, Bluetooth management and Android Activity/Service/Broadcast Receiver management. The Top Layer is the Application Layer where the implementation of Mobile Computing Module, “craving study app”, is installed and executed.

### Android Main Components

In the field of Android Application Programming, there exist several important components such as Activity, Service, Broadcast Intent Receiver, Content Provider and etc. In this Mobile Computing Module, the most frequently-used and significant components are Intent, Activity, Service, Broadcast Receiver, and sharedPreferences storage, which are key-value sets.

Intent is a message object which is used to request an action from another application component. Most of the communication in this system between the different modules applies the Intent object and Pending Intent object, which is a wrapper around an Intent object, to inform the Android system’s Alarm Manger to execute the Intent.

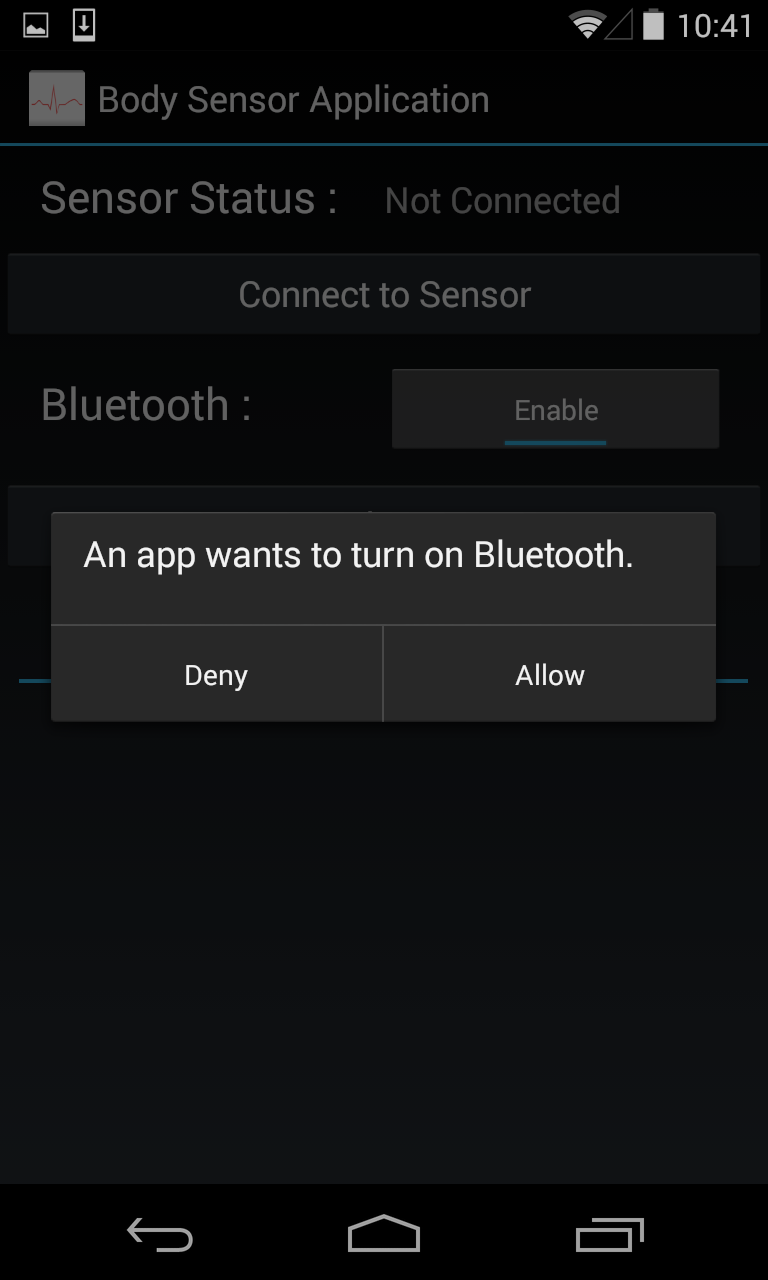
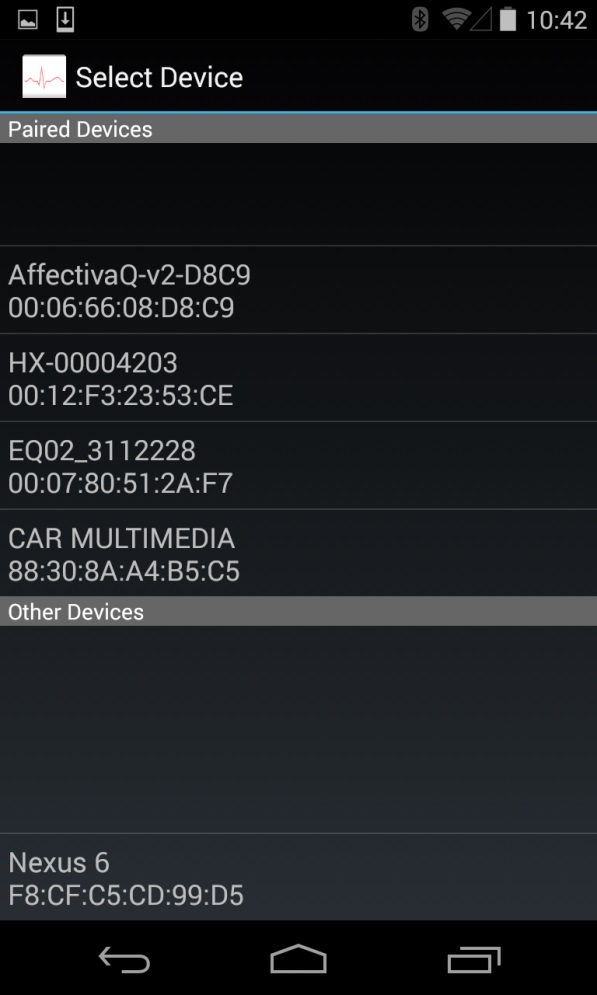
Activity is an application component that provides a screen with which users can interact in order to operate something on the phone. All the user interfaces in this system apply the Activity to implement interactions between the phone and users.

Service is an application component that can perform long-running operations in the background and does not provide a user interface. After another application component starts a service, the service keeps running in the background even if the user switches to another application [12]. Most of the operations such as triggering music alarm, sending broadcast intent, preparing dynamic broadcast receiver, or starting a new data collection thread are handled in the service in this mobile computing module.

Broadcast Receiver is the component that responds to the broadcast message. It includes dynamic type, with a lifecycle that could be controlled in the service, and static type, with a life cycle the same as the application itself. Both types are used in this mobile ambulatory assessment system.

SharePreference is the Android Framework built-in component used to store some collection of key-values in the system binding with the application data. In this system, all the system preference attribute and subject information is saved in the SharePrefernce storage model.

In this module, the application is composed of eleven broadcast receiver, ten activities, two services, several other android components and other normal Java Class files.

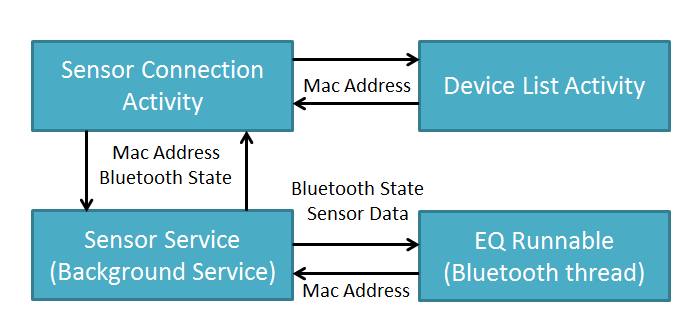
 

*Figure 4.2 Sensor Connection Activity Figure 4.3 Device List Activity*

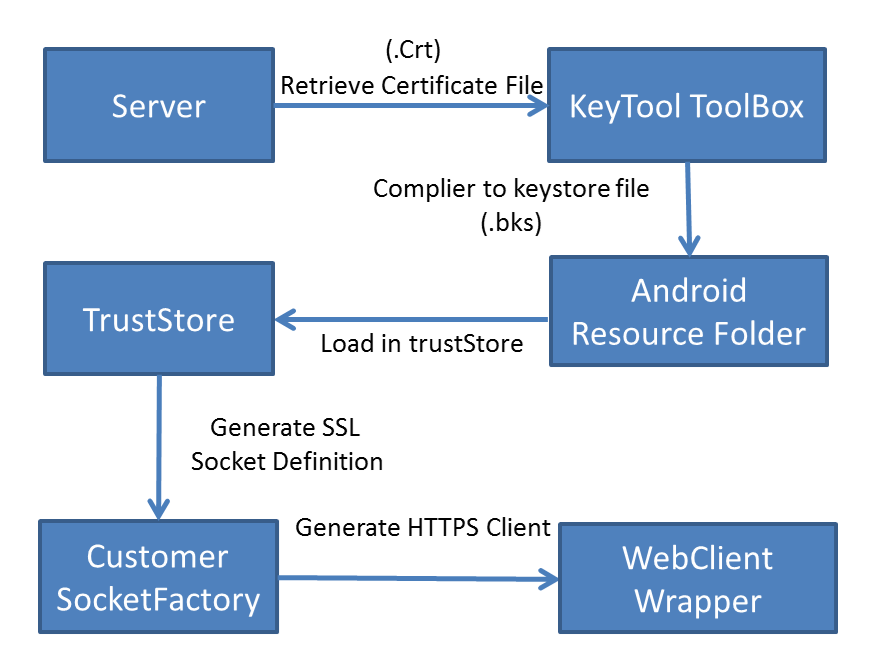
### Networking Module

Networking Module in the Mobile Computing Module is responsible for two tasks: Bluetooth Connection and Communication with Server.

For the first functionality, Networking Module uses the Android Bluetooth API to accomplish four major tasks to communicate using the Bluetooth: setting up Bluetooth, querying devices that are either paired or available in the local area, connecting to other devices through Bluetooth discovery and transferring data between devices [9]. The overall flow is shown in the Figure 4.4. Figure 4.2 and Figure 4.3 shows the User Interface for the Bluetooth Connection in Android Application.



*Figure 4.4 Flow Diagram of the Bluetooth Management*

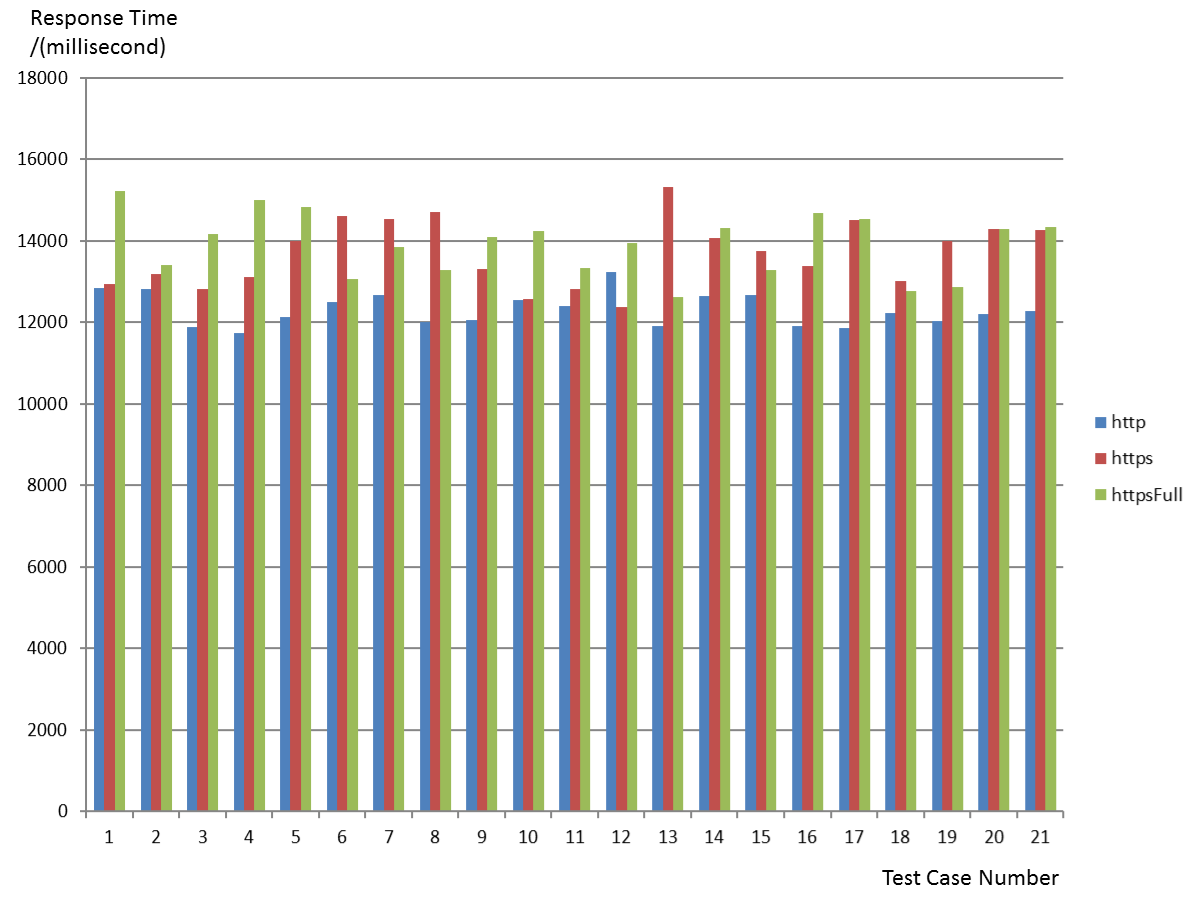


*Figure 4.5 Flow Diagram of HTTPS implementation in Android*

The second functionality -- Networking Module -- is responsible for posting the collected data to the server and retrieving the response message from the server. All communication over the internet takes place through the Hypertext Transfer Protocol Secure (HTTPS) protocol to prevent wiretapping and man-in-the-middle attacks so as to add another layer of security.

To implement the HTTPS POST in Android, two types of object need to be introduced: trustStore and keyStore. A trustStore contains certificates from other parties that the developer expects to communicate with, or from Certificate Authorities that the developer trusts to identify other parties. And, a keyStore contains private keys and the certificates with their corresponding public keys. By default, the android system does not trust the Certificate Authorities provided on remote server. Thus, to be able to verify the certificate from remote server, keyStore file is needed to be generated, then loaded into trustStore and finally applied into Android Framework.

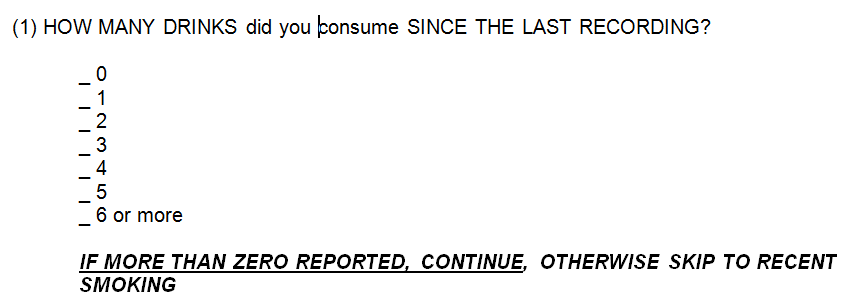
The implementation flow of HTTPS flow is shown in Figure 4.5. First, the certificate (.crt) on the server needs to be complied into keyStore files (.bks). Then the keyStore file is assembled in the resource folder (assets folder) of the android application. Then the networking module customizes the SSL Socket Factory based on the customized trustStore object with the assembled keyStore file and then creates a new class extends the DefaultHTTPClient class to add HTTPS support, including the port settings. In addition, all the communication tasks are executed in a separate thread by using the android built-in AsyncTask Class to improve system robustness and stability.



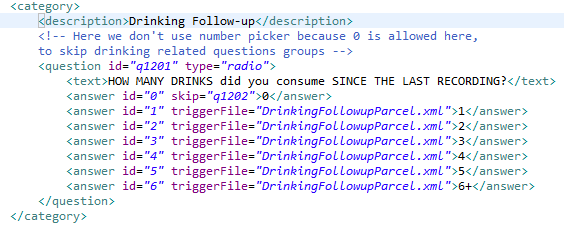
*Figure 4.6 Comparison with HTTP, partially HTTPS and HTTPS*

The response time of HTTPS connection, implemented in this system, is increased in a negligible way compared to traditional HTTP connection. Figure 4.6 shows one test case comparing the HTTP connection, Partially HTTPS connection (skip the verification of certificate) and HTTPS connection. In this test case, the package size is 4.2MB, the two communication parties are MOTO Droid Phone using the LTE connection provided by Version and remote server applied in this whole system. The difference of the average response time among three different connection methods is less than 2 seconds. In our real case, the package size of the system is much smaller than the test case, in that the addition of the response time applying the HTTPS other than HTTP can be ignored.

### Survey Module



*Figure 4.7 Question Sample*

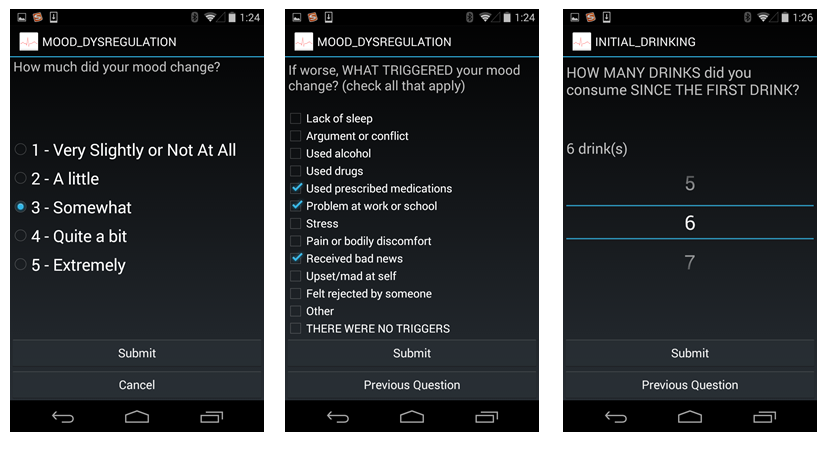


*Figure 4.8 Question XML File*

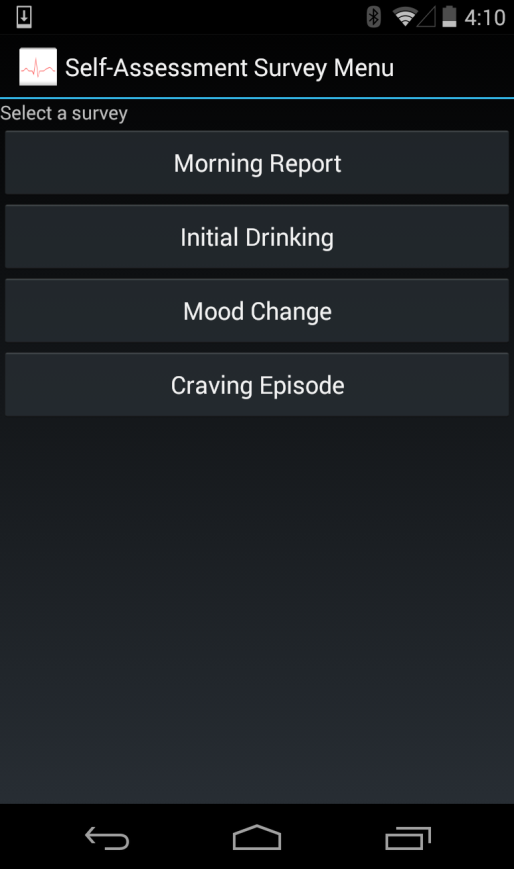
The survey resource file is written in XML file, which is extensible markup language. Figure 4.7 and Figure 4.8 shows an example of the way to integrate raw survey questions into corresponding XML file. XML consists of tags, which are text surrounded with “<” and closed with either “>” or “/>”. Tags also have attributes, which go inside tags, and allow texts between tags. To parse the XML files, the SAX parser [13], derived from javax.xml.parsers.SAXParser library, is used.

All the surveys are generated in the XML Survey Activity. The corresponding XML file is passed to the parser when XML Survey Activity is started and read into a data structure, which is essentially a linked list. Each question object contains a list of answer objects. Questions are then loaded into categories. Questions, Answers, and Categories are all represented with interfaces and abstract classes which allows easy refactoring, re-use and extension [14].

The Survey Module supports a variety of question types, and due to the object oriented structure of the survey module it is convenient to create new question types. Currently the application supports three following question types shown in the Figure 4.9: radio buttons that only permits one answer to be selected, check boxes that allows multiple answers to be checked and Numeric questions that allows the subjects to pick a number as the answer.



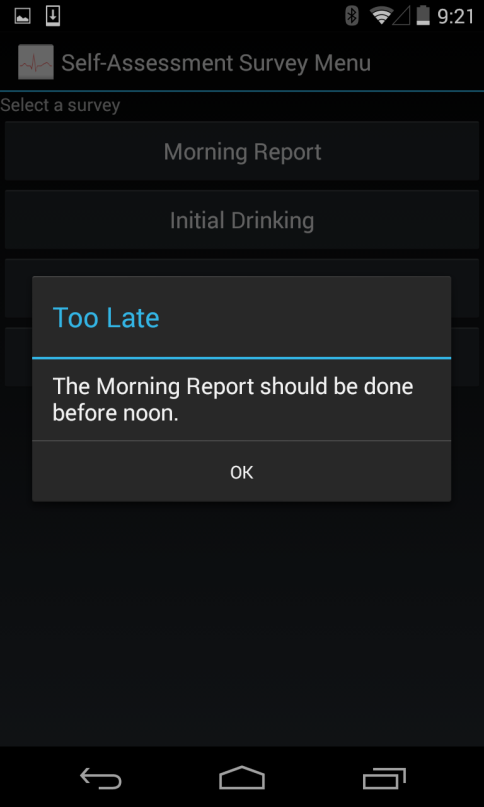
*Figure 4.9 Question Layout*



*Figure 4.10 Survey Menu Activity*

Furthermore, there is a particular group of question in all surveys which are emotion state questions. The sequence of these questions is designed to be in a random manner when the XML Survey Activity is generated, which is implemented using the random function since all the questions are stored in the List Object. In addition, subject could browse back by pressing the “previous” button shown in Figure 4.9. After all questions are finished and the “submit” button is pressed, all the answers will be collected by the Data Collection Module and sent to server managed by the Networking Module.

Moreover, there are two conditions when the XML Survey Activity is brought to the screen. Self-reported survey is triggered by manually pressing the survey button in Survey Menu Activity as shown in Figure 4.10. The Survey Menu Activity could send the intent containing the survey type to background service to construct the corresponding XML Survey Activity. The other condition that starts the XML Survey Activity is the system triggered survey: In a similar way, the intent containing the survey type information is sent to the XML Survey Activity class. The only difference from the self-reported survey is that in the case of the system triggered survey, the sender of the intent is the survey broadcast receiver, which sends the intent after receiving the signal. The time signal triggered is controlled by the Scheduling Module, which is discussed later in the thesis.



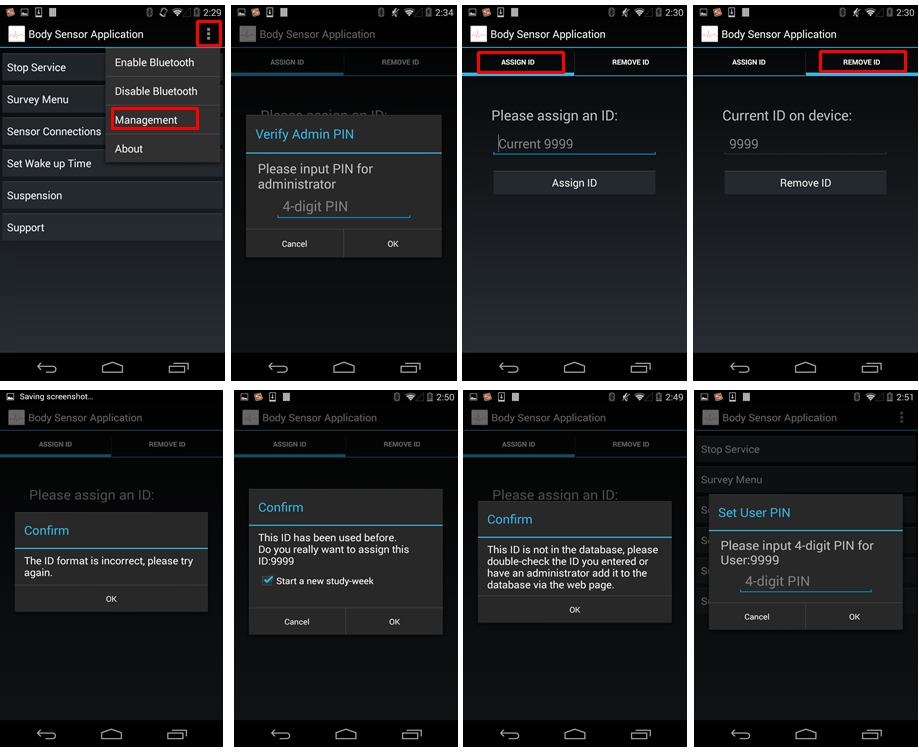
*Figure 4.11 Alert dialog for the Survey*

Also, Survey Module provides alarm sound and vibration warning for two situations: the survey is not completed after a long time (alarm interval is every five minutes), or the system-triggered survey is shown on the phone screen for the first time as a reminder to encourage the subject to complete the survey.

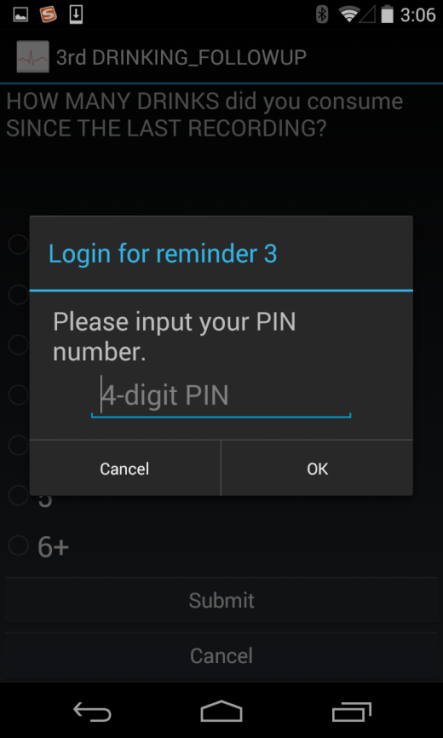
Last but not least, the Survey Module provides time checking for the survey, as some survey is not supposed to be taken at a certain time. For example, the Morning Report is not supposed to be taken after 12:00 P.M. Thus if the Morning Report Survey is manually triggered by the subject after noon, the alert dialog will pop up to remind the subjects and prevent unreliable survey data collection.

### Account Management Module

Account Management Module is designed for two types of users: administrator (mainly the researcher) and subject. The administrator management entry point is accessed through clicking the right corner of the action bar and pressing the “Management” item in the drop-down list in the Main Activity Layout. The system provides password verification for the administrator. After the identity is verified by the system, the administrator could assign ID (when issuing a phone for a subject) or remove ID (when the subject is done with his recording period). Figure 4.12 shows the interfaces of the Admin Management Layout. Furthermore, in the assign-id-tab activity, the Account Management Module, collaborating with Network Module and Server Subject Monitoring Module, checks the validity of the administrator input of the subject’s ID.



*Figure 4.12 Admin Management Layout*



*Figure 4.13 Subject Identity Verification Layout*

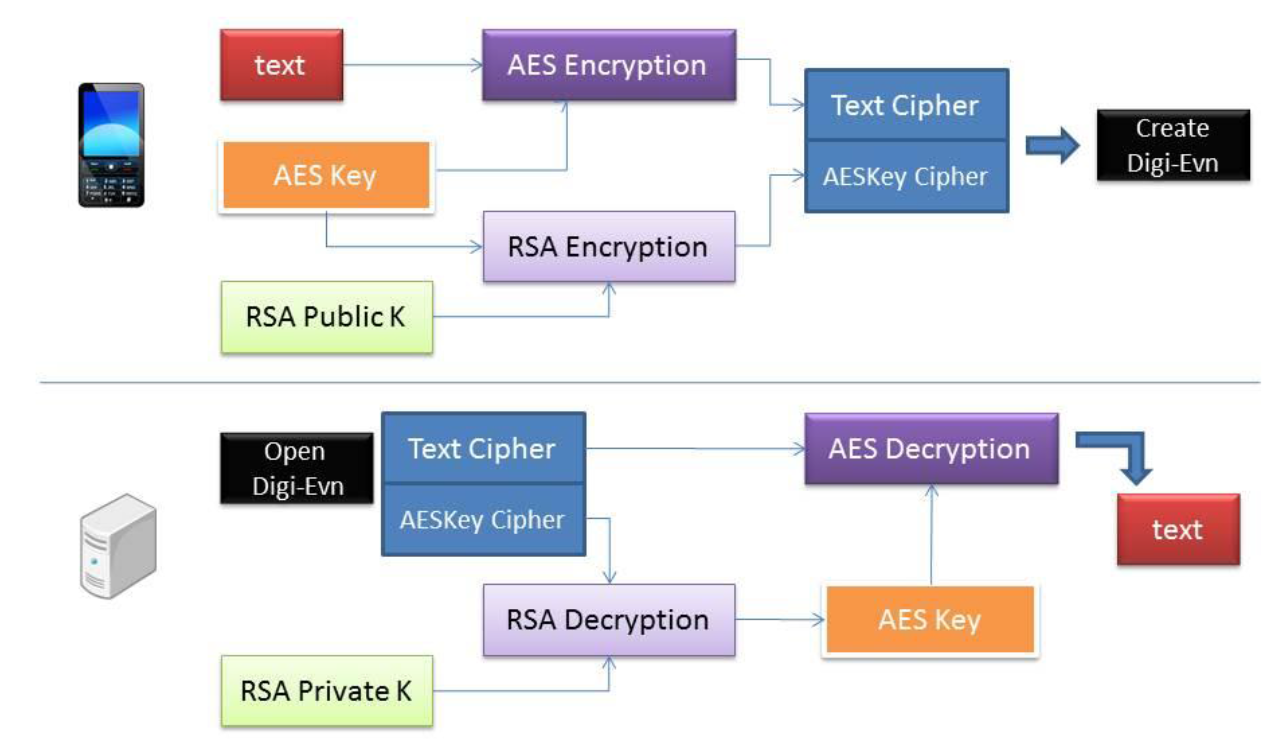
The first three screenshots on the second row of the Figure 4.12 are the three scenarios after the subject ID is checked against response information retrieved from the database on the remote server. If the input format (must be a 4-digit number) is wrong or the input ID does not exist in the database on the remote server, alert dialog will pop up with details and prevent the user from taking further steps. Otherwise, the confirmation dialog with the check question will pop up to let the administrator assign the subject ID in a proper way. Then administrator will let the subject set a pin number for himself, shown in the last graph of Figure4.12, which prevents other phone users from accidently filling the survey supposedly to be finished by the subject himself, to increase credibility of the survey data. The Account Management Module performs subject identity verification check before the subject starts to fill the survey, shown in Figure 4.13.

### Data Collection Module

Data Collection Module is also another important module in the Mobile Computing Module. It is responsible for collecting all the sensor readings, preparing data in blocks as a buffer for future posting and writing the entire sensor data and all other data retrieved from other modules, such as Phone Monitoring Module and Survey Module, into local storage as backup files.

For sensor data connation, the Data Collection Module mainly consists of one broadcast receiver and two Java Runnable Objects serving as data collection thread. The Runnable Objects follow the singleton design pattern to make sure there is only one instance of the Runnable object running. The broadcast receiver is designed to monitor the GPS data. By using the Android Location Manager API, when the phone location is changed, a message is then broadcast and as a result broadcast receiver will capture the GPS value and export the most accurate location data within a 30-second moving window. Two Runnable Objects are utilized for the external sensor data collection and internal sensor data collection, and they are instantiated by the Background Service Module. External sensor recording thread receives the data every 5 seconds. Internal sensor recording thread uses the sensor manager provided by android framework to retrieve new data point when sensor value is changed based on to the preset sensitivity level. In addition, when it comes to the accelerometer data, a median filter is implemented to filter the data with the window size of 5 seconds and export the data in every 5 seconds. All the single data entry labeled with current timestamp are stored in buffer as a preparation data set for future posting to the remote server, which saves much more power compared to sending the data in a higher frequency. The Module use File Writer instance to write data into local storage.

Moreover, all the sensor data buffers are encrypted by a hybrid cryptosystem, as shown in the Figure 4.14, which combines the convenience of an asymmetric public-key cryptosystem with the efficiency of a symmetric-key cryptosystem. In this Module, RSA algorithm is implemented as the asymmetric cryptography and AES is implemented as the symmetric cryptography. The data is encrypted by AES while the unique AES key is protected by RSA key-pair (public key and private key) cryptography. Thus, even if someone other than the researcher retrieves the backup data from the phone; the encrypted data is extremely difficult to be decrypted, which increases the data security level.

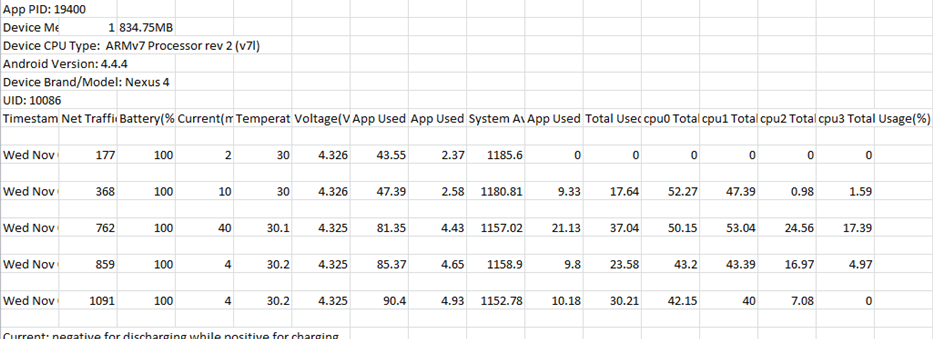


*Figure 4.14 Hybrid Cryptosystem Implemented in the System*

### Phone Monitoring Module

Phone Monitoring Module is responsible for monitoring the device properties, system usage and application performances. By default, the module detects the battery level, phone connection state and phone shut-down/boot behavior in every five seconds. For shut-down/boot behavior detection, two broadcast receivers are registered to monitor the system broadcast intent sent when the phone is shutting down and booting. To detect the battery level, a static broadcast receiver is registered in the android application. Whenever the battery level is changed, system broadcast intent will be broadcasted and the receiver then captures the intent and updates the current battery level. For the phone connection state, Connectivity Manager and NetworkInfo Object in Android Framework are used to detect the connection state, which are “Hardware Problem”, “Connected” or “Not Connected”. For detecting battery level and phone connection state periodical, another broadcast receiver called recording receiver, an alarm manager and a service are implemented. The dynamic broadcast receiver is registered by the background service and embedded with an alarm manager which could call system alarm to schedule the broadcasting intent, which is only recognized by the recording receiver itself within the next five seconds. In this way, repeated monitoring is implemented.

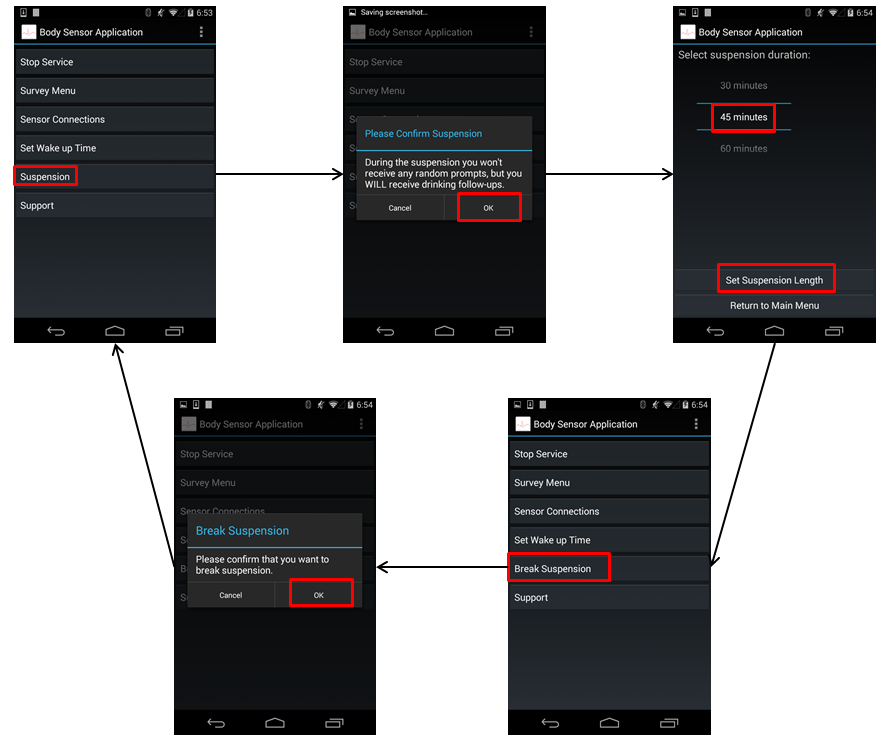
Phone Monitoring Module implements an advanced monitoring mode, mainly used for the purpose of developer debugging, which boasts the capability of monitoring many attributes and properties. Firstly, the tool uses the Package Manager in Android Framework to retrieve application Process Identifier. For the CPU, Network Flow and Memory Usage information are retrieved by reading the system files in the Linux File System Layer. It could not only detect the overall system performance Data such as CPU and Memory Usage, but also application CPU and Memory Usage based on the Process Identifier retrieved from the beginning. The local statistics files are shown in Figure 4.15.



*Figure 4.15 Statistics file for the application and phone performance*

### Scheduling Module

The Scheduling Module closely corresponds to the Survey Module and Phone Monitoring Module. To keep the Survey running, there are scheduling tasks for the different type of system-triggered survey and suspension/breaking suspension operations. Restoring tasks for restarting the application and rescheduling all the scheduled operations after device is booted are also crucial. Most of the scheduling is implemented in alarm manager instead of timer task object because the scheduling tasks must be brought even if the application is stopped. Only alarm manager could meet this requirement as it calls the system level service, whereas the timerTask is only active when the application process runs. To perform the time-based operation which is not periodic, an Android API, set(….) method in alarm manager class, can implement the job. Yet, to utilize the alarm manager setting repeated task in the android device whose OS version below 4.4 (kitkat), there is a simple API, called setRepearting(…) method, that could implement this job. However, when the device’s operating system version is above 4.4, in order to save more battery power, this method could not schedule the task at the exact time. Instead, it collects and enforces different alarm tasks scheduled within a range of time to be performed at a same time. Thus, in the application targeting the recent version, the system assembles background service, broadcast receiver and alarm manager to meet the requirement of performing the time-based tasks repeatedly. The background service registers the dynamic broadcast receiver and then sends initial intent to activate the alarm manager inside the broadcast receiver, so as to perform time-based operation and schedule sending the broadcast intent recognized by the broadcast receiver itself. The dynamic broadcast receiver’s life cycle is binding with service, which means the broadcast can keep receiving the intent and performing the tasks as long as the background service is not destroyed. Suspension/break suspension tasks are triggered by human-device interaction in a variety of activities, as shown in Figure4.16. A Boolean flag is used for detecting the suspension state and intent, which are sent to the daemon receiver to cancel or reschedule the surveys. All the scheduled time is stored in the sharedPrefernce components discussed in chapter 4.1.2. In this way, system could restore the schedule time by reading through the sharedPrefernce (key-value pair structure).



*Figure 4.16 Suspension/Break Suspension Flow Dialog*

The above implementation is applicable in most cases. However, the above implementation cannot perform satisfactorily when the phone is shut down and rebooted. Unlike other mobile operation system, the android system is much more high-level than the corresponding hardware component (especially the real time clock, or RTC). This hardware component wakes the CPU by raising an interruption and calls the CPU to run a dedicated program. Thus, after the phone is rebooted, the previously defined scheduled time is destroyed. To resolve this problem, the system uses a startup broadcast receiver registered in Android Manifest XML file, with the corresponding permission to allow the broadcast receiver to revoke the running application, after which the above methods can be safely used to restore all the time-based operations.

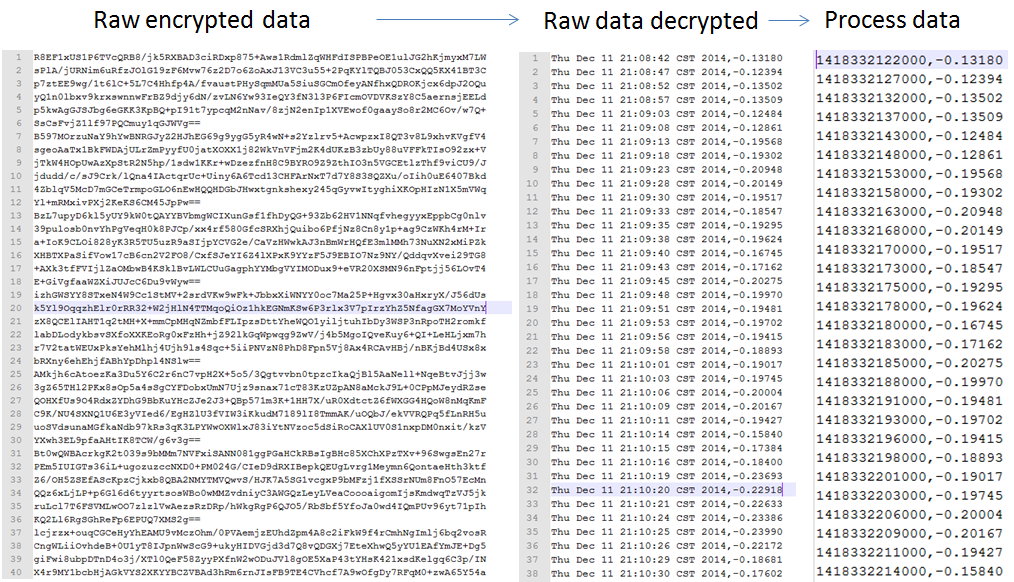
### Background Service Module

Background Service Module connects all the other Modules in the Mobile Computing Module, which is described in the previous sections. The Background Service is composed of two Service components. Recording service is started by the beginning of the application, while Sensor service is activated after the external sensor is connected to the smartphone. Service components can perform long-running operations in the background even when the subject switches to another application. Thus, data communication between the different Modules takes place mainly through the service; most dynamic broadcast receiver is registered and binding with the service; starting or terminating the thread is also performed in the service. In addition, this module is responsible for certain background operations such as audio playing and vibration.

### Utility Module

The Utility Module consists of several JAVA class containing the static final variables and common used functions, a Logger class to create logs serves as a debug tool for the application, sharedPrefence and other resource file in the application folder. KeyStore files and all the Survey XML resource files are put into the assets folder; all the layout XML files are under res/layout folder; all the icons used in the application are under res/drawable folder while all the strings shown on the activities are defined in res/values/strings.xml file; all the system preference is stored in shardPreference; In addition, An AndroidManifest.xml file is in the root directory of application folder. The manifest file presents essential information such as package name, registered components and corresponding hosted processes, application permission, target API version range and etc. In this way, all resource files and utilities are following MVC pattern and easily to be handled and modified.

### Data Recovery Module



*Figure 4.17 one example of data recovery*

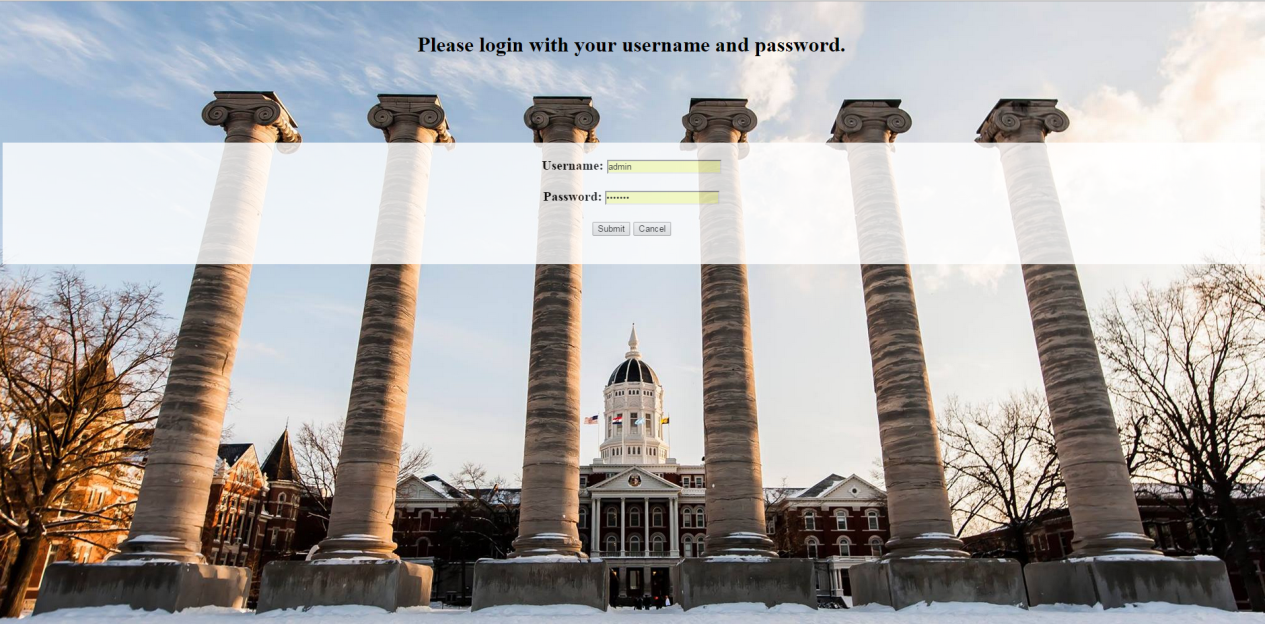
Sometimes data on the server can be deleted accidently by the researcher (A real case happens in practice). Data Recovery Module is built for this reason. An additional android application is developed to simulate the decryption process of the “craving study app” and collaborate with the Networking Module to interact with remote server to restore the data on the Server. Figure 4.17 shows one example of server data recovery from the encrypted backup data on the phone local storage. The first graph is the encrypted backup files while the others two are the restored files.

## Server Subject Monitoring Module

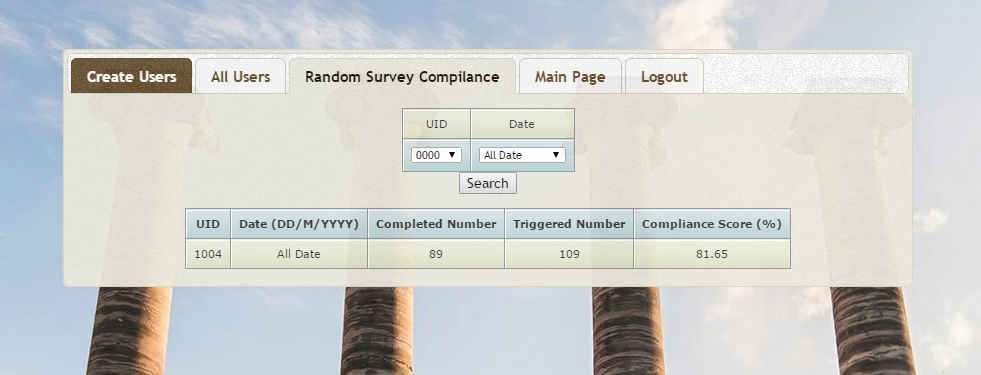
### Data Processing Module

The Data Processing Module is written in PHP language. To begin with, after receiving the data, PHP program calls the pre-complied .jar file to decrypt the files to retrieve the raw data. Then, the module writes the raw data into flat system on the server and processes data in a proper way by detecting the category of the raw data. The processing operation is mainly divided two types: the first type provides temporary file in a suitable format for plotting function of the Data Demonstration Module, while the other type creates new files based on the raw data. One of the most important generated file in this module is the Survey Score file. Mentioned in the Survey Module of the Mobile Computing Module, emotion states questions are embedded into all survey data. And each emotion question could fit into a certain category provided by the physiological researcher. The category includes: Positive, Negative, Fear, Hostility and Sadness. This file contains the score of subjects’ emotion state (five category mention above) aggregated with the corresponding timestamp by calculating emotion items of the survey data. All the processed files could be treated as the initial raw data set for Data Analysis Module.

### Administrator Management Module



*Figure 4.18 Login Page*



*Figure 4.19 Admin Management Page*

Administrator Management Module provides interface for researcher to login the website and maintain the subject condition by connecting with the MYSQL database. In this Module, PHP, JavaScript, CSS, HTML and SQL languages are mainly used.

Figure 4.18 shows the Administrator Login page. In the implementation, all the administrator password entries are stored with the SHA1 value of the password string in the MYSQL database to provide Encrypted Storage Model to increase security level.

Furthermore, regular expression check mechanism and PHP built-in addslashes() function, returns a string with backslashes before characters that need to be escaped, is deployed in this module for the user input to avoid SQL injection Attack.

Figure 4.19 shows the webpage for Admin Management Page, five tabs are in this page, researcher can create or maintain subject information stored in the database and provide random survey compliance score to evaluate the survey completion percentage of the subjects.

### Data Demonstration Module

Data Demonstration Module is responsible for displaying all the human-readable data on the server and graph plotting for survey score data, sensor data and phone state data of the advanced mode. Figure 4.20 illustrates the file check webpage. Administrator could choose the subject ID and its corresponding files and graphs to read and download. All the files can be filtered by the date, subject ID and file category. The .apk file, executable file for android application, is also available for download in this page.



*Figure 4.20 File Check Webpage*

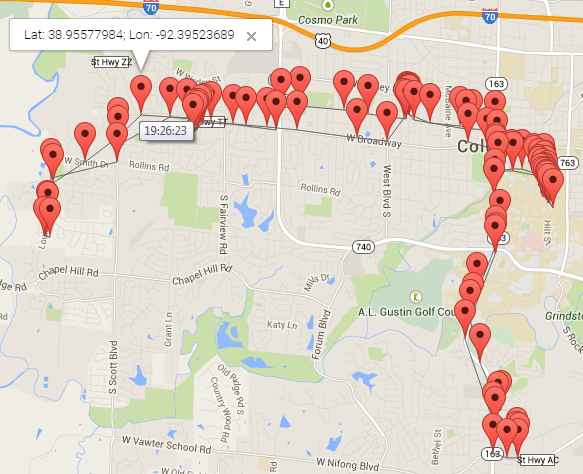
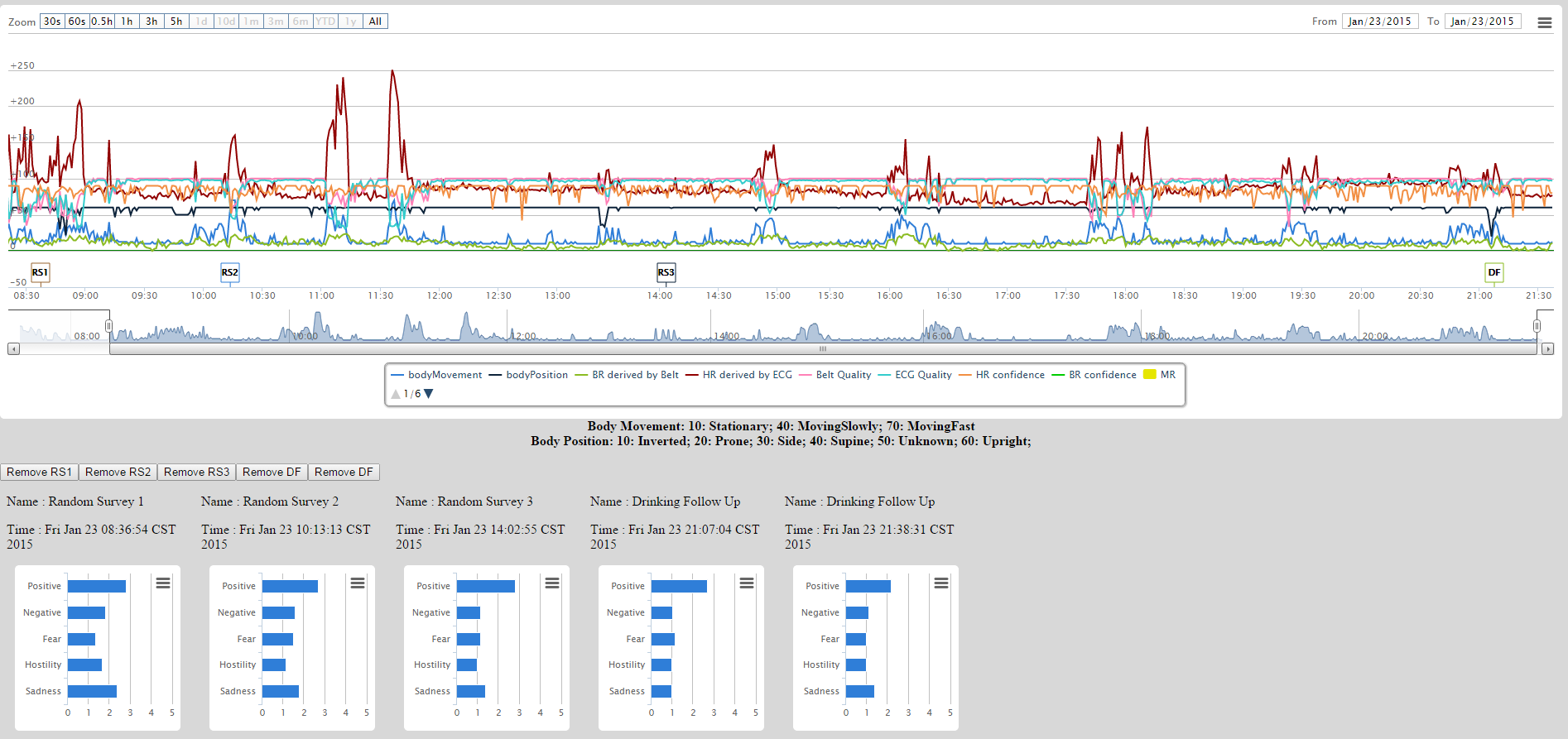


Figure .21 *An example of a sequence of time-stamped subject locations*



*Figure 4.22 An example of sensor and survey data graphs displayed on the server*

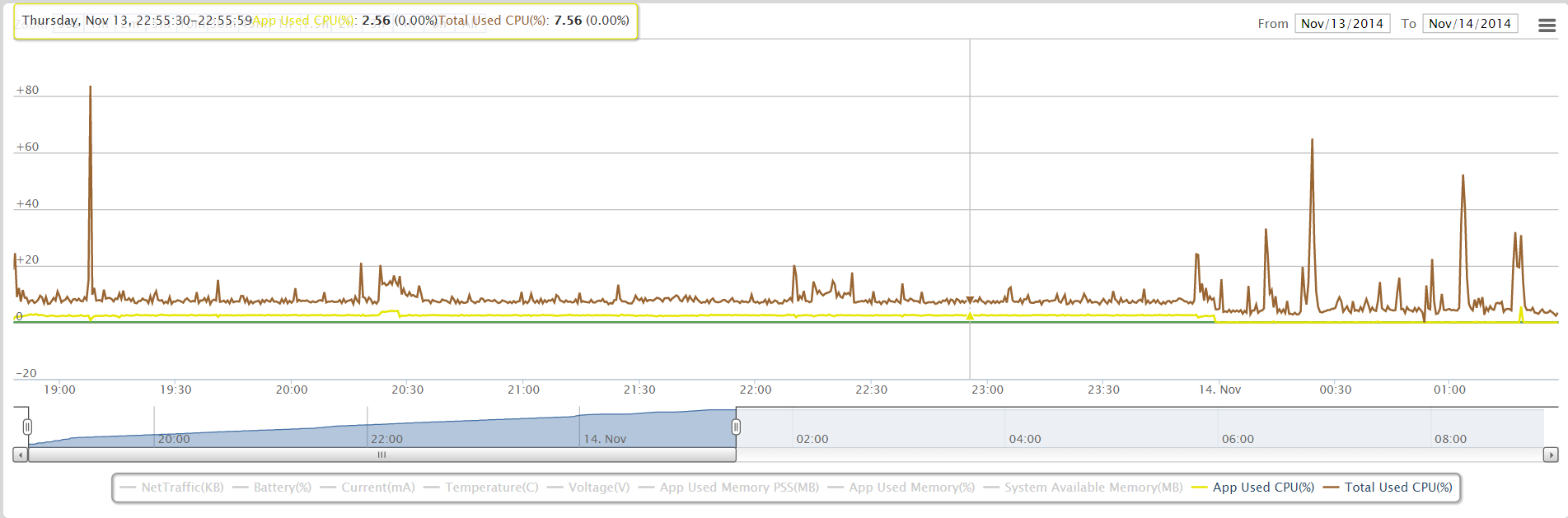


Figure .23 *An example of CPU status of App Used and Overall Used*

Interactive graphs of real-time sensor data are displayed to help researchers study real-time behavior implemented in this module. There are two types of graphs: locations on Google map, as shown in Fig. 4.21, and graphs containing sensor data and survey data, as shown in Fig. 4.22. A location graph shows a sequence of time stamped user locations on Google map. A sensor data graph shows all sensor readings and emotion values derived from survey data. The graphs are interactive, in which each type of items could be displayed or hidden. Interactive data range and duration selection provide a flexible and clear way of visualizing the data.

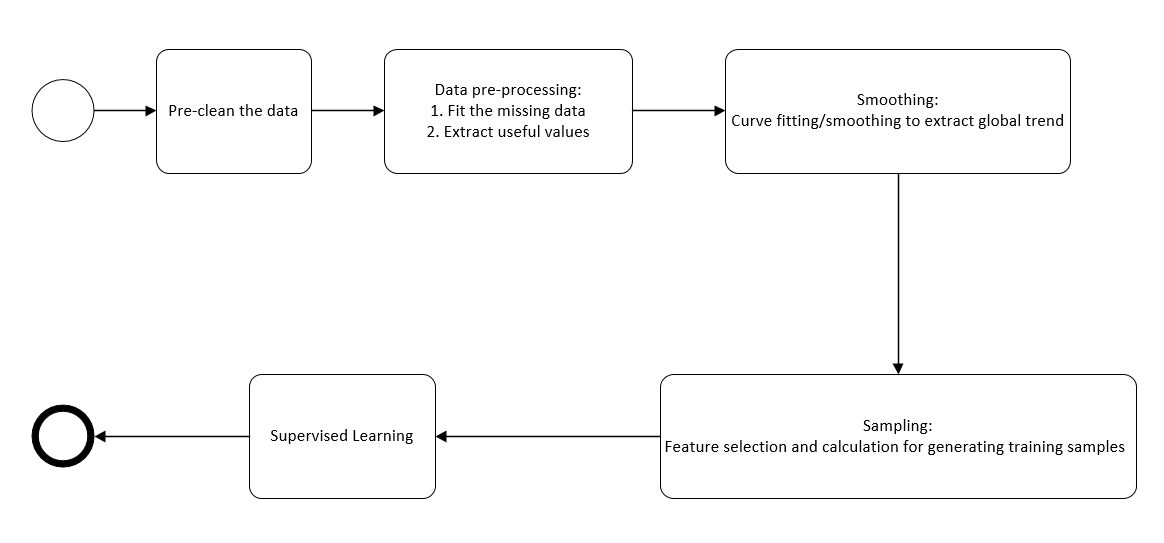
In addition, if the advanced mode of phone monitoring mode is activated, then the Accumulated Network Traffic, Battery Level, Memory and CPU Usage can also be visualized on the webpage. Figure 4.23 presents an example of CPU graph, which shows the application used CPU and Total System Used CPU.

## Data Analysis Module

The Data Analysis Module is mainly developed for research on the drinking craving state in current stage. Based on the survey and sensor data collected from the mobile system, the steps of the Data Analysis Module are as follows:

* Data preprocessing: handling missing data and extracting useful values.
* Data filtering and curve fitting/smoothing to extract global trend.
* Feature selection and calculation for generating training samples
* Supervised Learning model construction

Thus four Modules, Pre-processing Module, Graphic Module, Sampling Module and Supervised Learning Module, are dealing with the four tasks sequentially. The Figure 4.24 shows the pipeline of the Data Analysis Module.



*Figure 4.24 Pipeline of the Data Analysis Module*

### Pre-processing Module

Pre-processing Module mainly has two functionalities: pre-cleaning the data and pre-processing the data, including fitting the missing data and extracting useful values. As a variety of data is collected on the server, selecting the suitable data files and merging them into one file for each data set is important. Since the emotion state research is still in the preliminary stage, the thesis focuses more on alcohol craving state.

Thus, nine types of data entry are need to be collected and merged into one file: timestamp, ambulation, heart rate, breathing rate, heart rate confidence, breathing rate confidence, skin temperature and drink report state. In the pre-cleaning state, some sensor readings are treated as missing data and converted to “Nan” value if the sensor reading itself is not in reasonable range, or if the sensor corresponding confidence is lower than the threshold recommended by the expert. Next, several files are split from one file recording one entire day. Recording is separated by detecting whether the timestamp is continuous or not. The pre-cleaning part is implemented in JAVA while the other parts in this Module are all implemented in MATLAB.

After pre-cleaning the data, Pre-processing Module fits the missing data mainly by applying two one-dimension data interpolation method: Linear Interpolation and a piecewise polynomial interpolation called Spline Interpolation.

If the two points are given by the coordinates ( ,) and ( ,), The Linear Interpolation is the straight line between these points. To fit the value of y, which is the unknown value at x between [ ,], Eq. 4.1 is applied. For this application, the timestamp usually acts as the x value in Eq. 4.1.

(4.1)

The second method is Spline Interpolation. Eq. 4.5 is applied. P(x) stands for the value y which is unknown value at x between [ ,]. All the formula and description below is referenced from “Numerical Computing with MATLAB” [15].

Let donate the length of the kth subinterval:

(4.2)

Then the first divided difference, , is given by

(4.3)

Let donate the slope of the interpolant at :

(4.4)

Consider the following function on the interval, expressed in terms of local variables and :

(4.5)

Also the n linear equations in n unknowns:

Ad = r. (4.6)

The vector of unknown slopes is:

(4.7)

The coefficient matrix A is tridiagonal:

(4.8)

The right-hand side is:

(4.9)

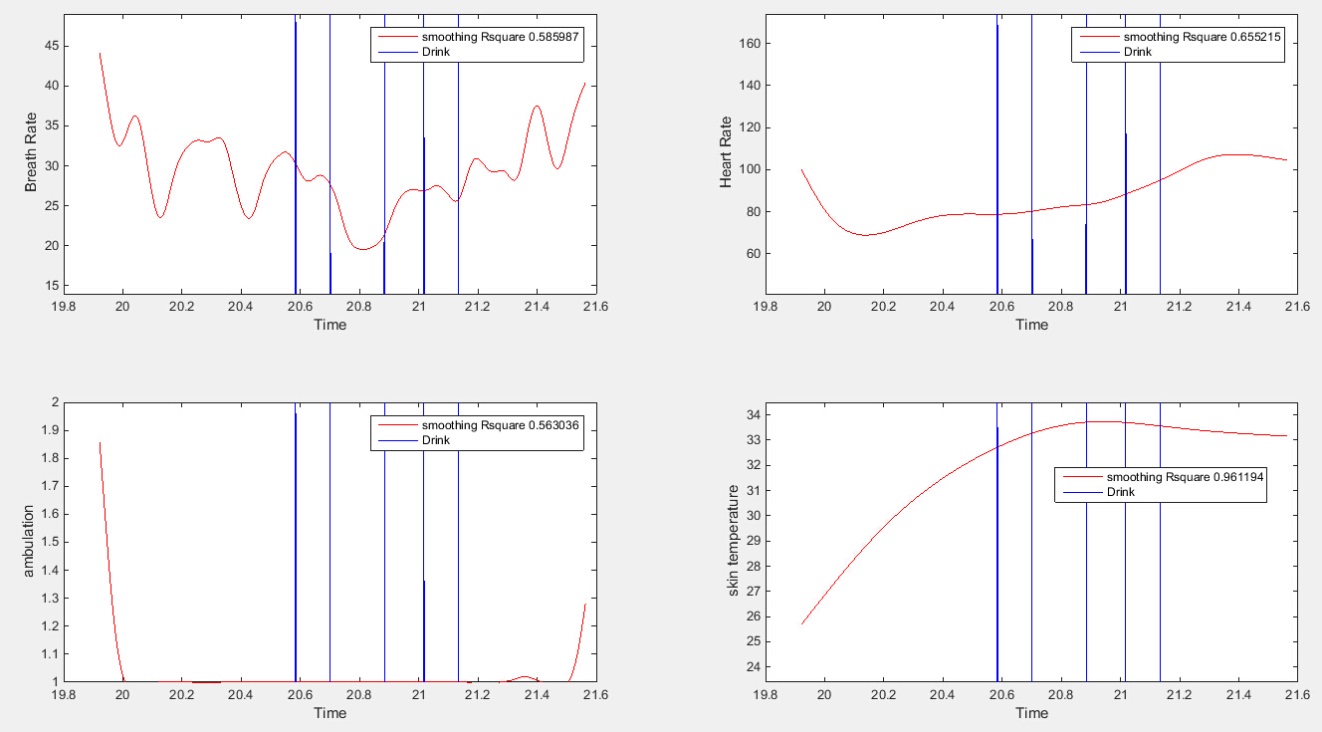
The two values and are associated with the end conditions.

### Graphic Module

To extract the global trend of the data, Graphic Module is developed. In current module, smoothing splines is utilized. To smooth out noisy signals, moving average, median filter, and smoothing spline are attempted. After comparing some results, smoothing spline is chosen for the drinking carving data set collected in Mobile Computing Module. The smoothing spline function s minimizes the following Eq. (4.2):

(4.2)

where p is the smoothing parameter between 0 and 1, is the weight, and is a training example. p = 0 produces a least-squares straight-line fit to the data, while p = 1 produces a cubic spline interpolant.



*Figure 4.25 A drinking sample collected by real subject, from around 19:50 p.m. to 21:40 p.m. The four measurements are breath rate, hear rate, ambulation (1- stationary, 2-slow moving, 3 fast-moving), and skin temperature. Each blue vertical bar represents a time that the patient reported a drink in drinking follow-up surveys.*

Fig. 4.25 shows a drinking sample from real subject in one evening. The four measurements shown are breathing rate, heart rate, ambulation (1-stationary, 2-slow moving, 3-fast moving), and skin temperature. They are not raw data, which are very noisy and rugged. Instead, the curves are results of smoothing spline. Different smoothing parameters p are applied in order to make R-square value greater than 0.5, which means the regression line not only shows smoothed tendency, but also fits all points well. Each blue vertical bar in the figures represents a time that the patient reported a drink in drinking follow-up surveys.

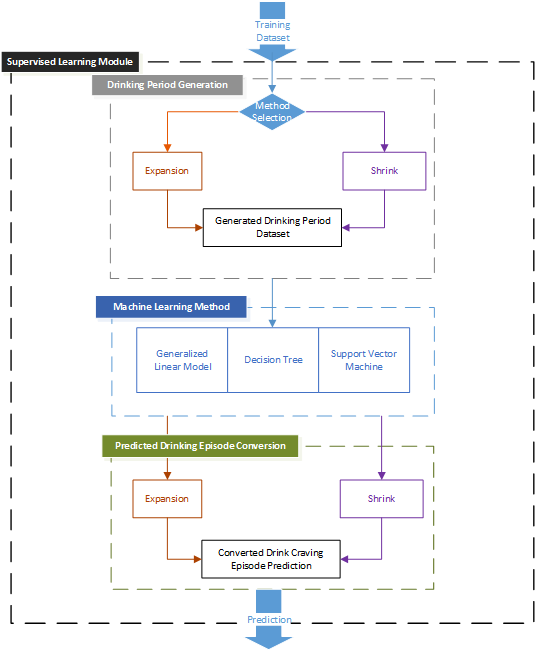
There are some interesting observations. For example, breathing rate is generally lower during the drinking period, whereas the heart rate and skin temperature is generally higher during the drinking period.

### Sampling Module

The Sampling Module is responsible for variable sampling by applying different methods for variable sampling. The first method simply uses all the raw features. The second method adds the two-feature-interaction as new features. Generalized linear model is then used to choose the parameter and corresponding dataset based on the overall accuracy for both methods.

### Supervised Learning Module

First, take a look at data generation part. The current system focuses on the drinking analysis and prediction. In the dataset, the drinking status (0 represents drinking status is false, while 1 represents drinking status is true) is reported by the subjects at some certain point of time, which could not fully represent the duration of the drinking period. And also, the proportion of the drink-data is too low (around 0.25%), which could not predict drink craving episode precisely by directly using some machine learning methods.

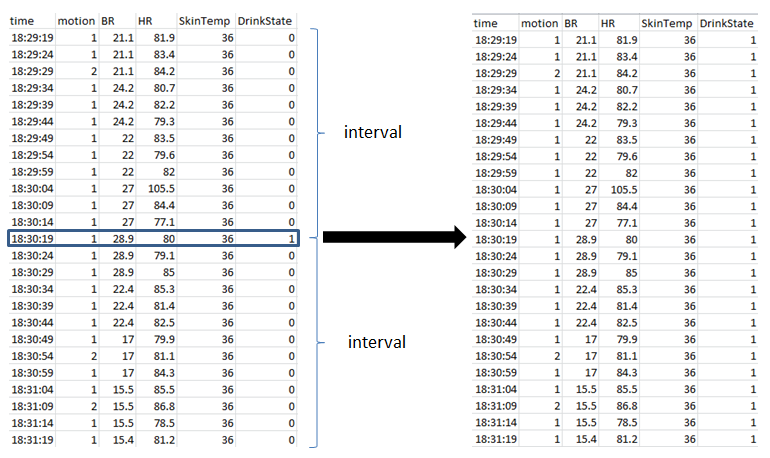


*Figure 4.26 Flow Chart of Supervised Learning Module*

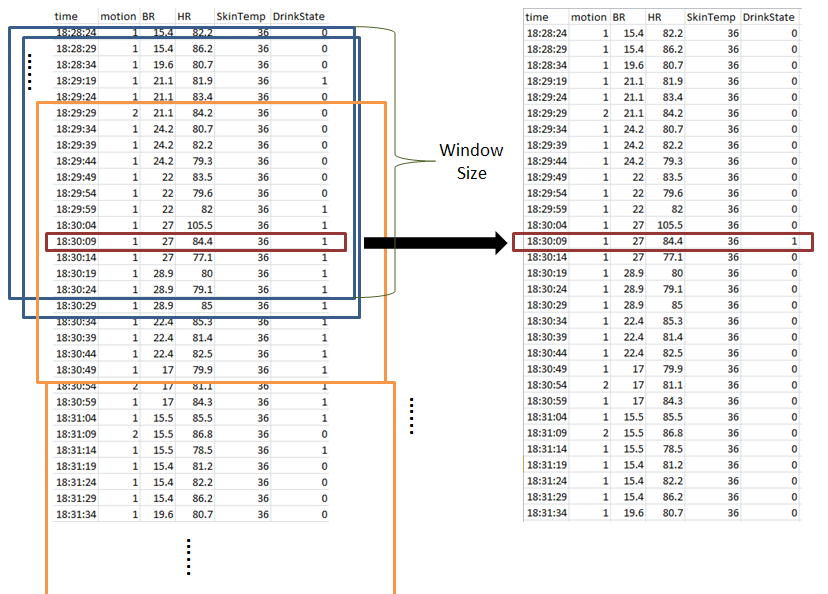
Figure 4.26 shows the flow chart of Supervised Learning Module. Rather than directly predicting the drinking carving episode, the system proposes an idea to firstly generate the drinking period based on the training dataset; next, it predicts the drinking period using some machine learning methods, which are generalized linear model, decision tree and support vector machine (SVM); then the system converts the predicted drinking-period to drinking craving episode prediction; finally, the system compares the result with the converted prediction with the real drinking episode testing dataset. The system applies two methods for the “Drink Period Generation” and “Predicted Drinking Episode Conversion” steps.

#### Expansion Method

The first method is called “expansion”, it is because it first expand the drink-period, then convert it back to drink carving episode prediction after running some machine learning Method.



*Figure 4.27 An example of drinking period generation step in “Expansion” method*



*Figure 4.28 An example of conversion step in “Expansion” method*

In the expansion step, it expands the drink records to drink period. It sets drinking status reported within a certain period of timeframe around the drink-report time, to “true”. The other data in the original dataset is not modified.

In the Figure 4.27, parameter of period is one minute for illustration purposes (In the real case, the period is longer than the one in the example). The system tries different value of period (from five minutes to ten minutes) to generate different samples based on the chosen period time. After running the machine learning method, the system generates the prediction of drinking period.

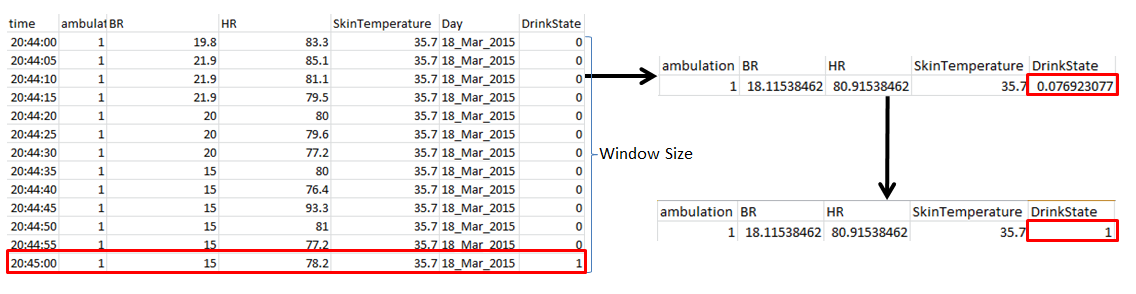
In the conversion step, the system applies a moving window to calculate the proportion of the drink-data, as shown in the Figure 4.28. If the proportion is lower than the threshold, the moving window moves on (like the blue windows in Figure 4.28). Otherwise, the system considers the middle entry as the drink craving episode entry. Then, as shown in red box, the system keeps drink state of middle entry as 1 (means “true”), while other entries is set 0 (means “false”). Then the moving window skips the checked data entries, and start moving on from the first entry not be tracked (shown as the yellow windows in the Figure 4.27). In addition, the windows size is the same of expanded drinking period.

#### Shrink Method

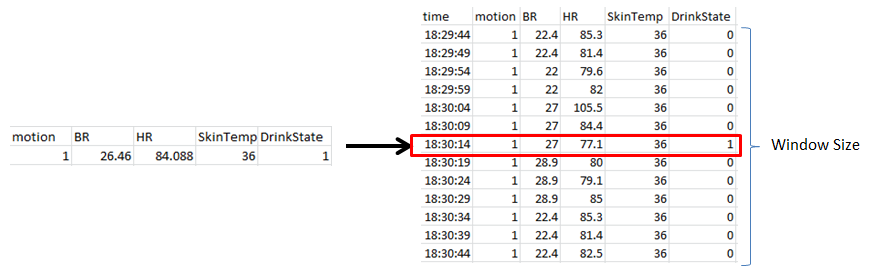
The second method is called “shrink”, which first shrinks the whole data and then expands the prediction after running the machine learning methods.

In the shrink stage, it uses a moving window following the steps below, as shown in Figure 4.29: First, the module divides the dataset into different subsets based on window size (numbers of selected entries). Secondly, the module takes each subset as input and calculates the average of each single data feature. Next, the module aggregates these features and output them as a single data entry. Finally, the module aggregates all the data entries together as new training sample. Also, as long as the drink state of output dataset is more than zero, the drink state is treated as “true” and value is converted to “1”.

After machine learning method is applied, same as “expansion” method, the system goes through the conversion step to expand the intermediate prediction. The Figure 4.30 shows an example of conversion step in the “shrink” method.



*Figure 4.29 An example of shrink step in “Shrink” method*



*Figure 4.30 An example of conversion step in “Shrink” method*

#### Generalized Linear Model

The Eq. 4.10 and Eq.4.11 [16] shows the logistic function of generalized linear model. The system constructs the model based on the training dataset. For the test data, if the value of p(x) is bigger than 0.5, the system treats the drink state as “true” and predicts it as a drinking data. Otherwise, the system treats the drink state as “false”.

(4.10)

(4.11)

#### Decision Tree

The Decision Tree learning is the construction of a decision tree from class-labeled training tuples. A decision tree is a flow-chart-like structure, where each internal node denotes a test on a feature, each branch represents the result of a test, and each leaf (or terminal) node holds a class label. The top node in a tree is the root node [17]. The algorithm for building a classification tree as follows [16]:

1. Use recursive binary splitting to grow a tree on the training data, stopping only when each terminal node has fewer than some minimum number of observations.
2. Apply cost complexity pruning to the large tree in order to obtain a sequence of best subtrees as a function of α.
3. Use J-fold cross-validation to choose α. For each k = 1, … , K:
   1. Repeat Step 1 and 2 on the fraction of the training data, excluding the kth fold.
   2. Evaluate the mean squared prediction error on the data in the left-out kth fold, as a function of α.

Average the results, and pick α to minimize the average error.

1. Return the subtree from Step 2 that corresponds to the chosen value of α.

Also, the model is constructed based on the trained dataset and use the model to predict the drinking state of test dataset. Pruned and unpruned tree are both applied.

#### Support Vector Machine (SVM)

An SVM model is a representation of the dataset as points in space, mapped so that the training data of the separate categories are divided by a gap that is as wide as possible. Test dataset are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on [19]. Three kernels are applied in this Supporting Vector Machine method: Gaussian or Radial Basis Function (RBF) kernel, Linear kernel and Polynomial kernel. Eq.4.13, Eq.4.14 and Eq.4.15 shows the formula for each kernel.

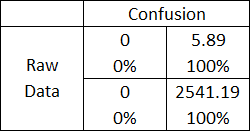
(4.13)

(4.14)

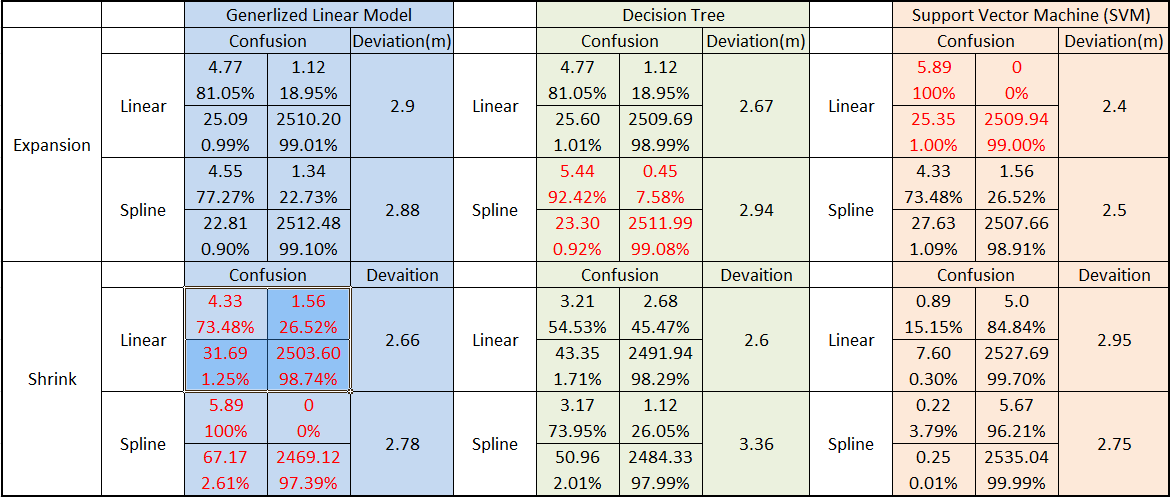
(4.15)

### Preliminary Result

In current stage, the system runs the experiments on the dataset collected from three different days. The size of each original raw data is 2500\*12. After data cleaning and feature extracting, the size of dataset is around 2500\*5 while the number of reported drink data is around 6. The proportion of the drink-data is around 0.23%. Because the raw data is biased, thus no drink craving episode prediction is made by directly applying the Machine Learning Method, as shown in the Table 4-1.



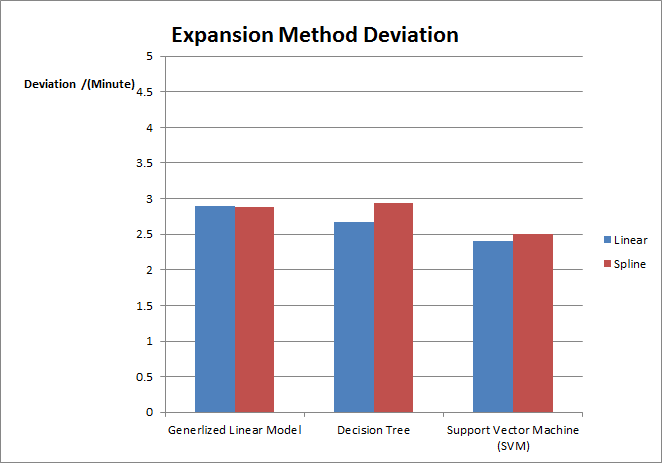
*Table 4‑1 Confusion matrix of the prediction result based on the raw dataset*



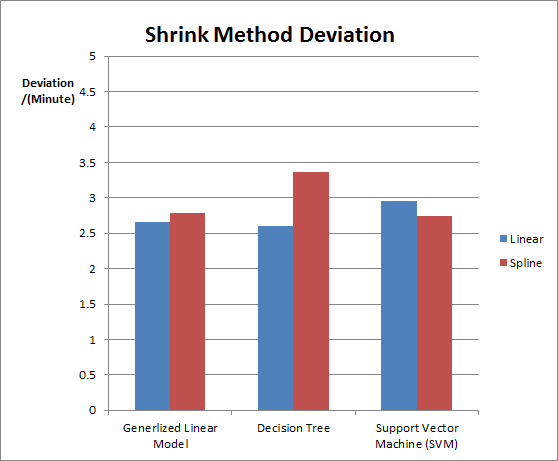
*Table 4‑2 Confusion matrix and deviation of “Expansion” and “Shrink” methods*

For the proposed methods, the evaluation rule of prediction allows some deviation because the drink episode point is always chosen to be the middle one during the conversion of intermedia prediction, which could provide deviation for the predication data. Thus, the rule is that if the drinking craving episode prediction is near to the real craving episode within the range of 5 minutes, the system considers it is a true prediction. Table 4-2 shows the confusion matrix and deviation of each method, which proves that the proposed methods contribute a lot to the drink craving episode prediction. Figure 4.31 and Figure 4.32 shows the graph of deviation result of “Expansion” method and “Shrink” method.

Based on the result, the linear interpolation and spline interpolation methods, for the missing data fitting, provide no big difference on the final result. In addition, overall, the “Expansion” method provides more accurate prediction than the “Shrink” method.



*Figure 4.31 Deviation result of “Expansion” method*



*Figure 4.32* *Deviation result of “Shrink” method*

# . Future Work

For the data collection part, firstly, more external sensors such as Hexoskin sensor will be utilized to improve sensor readings in the future. Thus, alternative collection function needs to be implemented both in the mobile side and server side. Next, the communication function in the Networking module of Mobile Computing Module needs to be improved. Currently, the system sends the data before checking the device connection state which means the server could not receive the real-time data if the device is disconnected from internet. Consequently, Sync data mechanism is next goal for the Networking Module. Correspondingly, Rational Database needs to be developed to make sure the sequence of the collected data is correct. Additionally, the support for the Bluetooth 4.0 is an alternative task to improve Bluetooth performance. In proposed system in this paper, only Bluetooth 2.0 is supported.

For the data analysis part, unsupervised learning method such as Naïve Bayes and K-means is the next task to be completed, which may contribute to the model construction. Implementation of more missing-value-fit method and sampling mechanism is another goal in the future. Additionally, Survival analysis model, deals with analysis of time duration until one or more events happen, needs to be added. For the drinking craving data, survival analysis is a suitable statistics method and need to be further explored more in the future work. Furthermore, after more real data from subject is collected, more features such as subject emotion state could be investigated through this context-aware system.

# . Summary

The thesis describes the development of a Context-Aware enhanced mobile assessment system for alcohol carving studies. The mobile assessment system can provide a new assessment method for alcohol drink study. The system provides real-time environment factors monitoring as well as the physiological factors/drink-report monitoring as opposed to traditional approaches where the subject is just assessed by a survey questionnaire where the subject is required to rate his intensity of alcohol craving without real-time monitoring. In current stage, several real subjects are using this system to provide real data. The mobile computing module has been successfully tested and deployed on variety of Android Platforms and cooperate with external sensor and remote server. And data analysis module could automate the whole process and run preliminary results and provide some invented idea to deal with the drink craving study data. Additionally, this assessment has been deployed not only in the drink craving episode, but also in other fields now such as emotion detection.

# References

1. Society for Ambulatory Assessment, 2012, www.ambulatory-assessment.org.
2. 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.
3. T. Starner, Wearable Computing and Contextual Awareness, Ph.D. thesis, MIT Media Lab., Apr. 30, 1999.
4. 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.
5. 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.
6. L. Constantine and H. Haij, "A survey of ground-truth in emotion data annotation," IEEE Int’l Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), pp.697-702, 2012.
7. G. Miller, “The Smartphone Psychology Manifesto,” Perspectives on Psychological Science, vol. 7 no. 3, pages 221-237, 2012.
8. F.H. Wilhelm, and P. Grossman, “Emotions beyond the laboratory: theoretical fundaments, study design, and analytic strategies for advanced ambulatory assessment”, Biological Psychology, vol. 84, pp. 552–569, 2010.
9. Equivital, EQ02 Life Monitor Sensor Electronic Module Datasheet,

http://www.equivital.co.uk/assets/common/SEM\_Data\_Sheet\_General\_HIDA3330- DSG-02.2\_.2\_.pdf, 2012.

1. Android Application Fundamentals http://developer.android.com/guide/components/fundamentals.html.
2. Nisarg Gandhewar, Rahila Sheikh. Google Android: An Emerging Software Platform for Mobile Devices [J]. International Journal on Computer Science and Engineering (IJCSE), 2010, 12-17.
3. Android Services http://developer.android.com/guide/components/services.html.
4. Sax Parser https://docs.oracle.com/javase/7/docs/api/javax/xml/parsers/SAXParser.html.
5. 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.
6. Cleve Moler (2008), Numerical Computing with MATLAB, Natick, Massachusets: The MathWorks, Inc.
7. James, G., Witten, D., Hastie, T., Tibshirani, R (2013), An Introduction to Statistical Learning.
8. Decision Tree Learning. In Wikipedia, <http://en.wikipedia.org/wiki/Decision_tree_learning>.
9. Random forest. In Wikipedia, http://en.wikipedia.org/wiki/Random\_forest.
10. Support vector machine, In Wikipedia, http://en.wikipedia.org/wiki/Support\_vector\_machine.