

OULU UNIVERSITY OF
APPLIED SCIENCES



Woubshet Behutiye

ANDROID ECG APPLICATION DEVELOPMENT

ANDROID ECG APPLICATION DEVELOPMENT

Woubshet Behutiye
Bachelor's Thesis
Spring 2012
Business Information Technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Degree Program: Business Information Technology

Author: Behutiye Woubshet

Title of thesis: Android ECG Application Development

Supervisor: Alakiuttu Tapani

Term and year of completion: Spring 2012

Number of pages: 36+6

This paper describes design and implementation of android ECG application that works together with the VS100, a wearable Vitalsens bluetooth ECG monitoring device. The main objective of the thesis was to discover development in the android Platform and produce a prototype ECG application. The thesis was commissioned to Domuset Oy; Oulu based wireless health Solutions Company.

The theoretical part of this paper discusses the android platform, its architecture and the development environment and tools that are used in android mobile application development. Basic components and procedures of android application development are described in the development process. The Vitalsens VS100 device is also explained in this part of the thesis. Emphasis was given on implementing bluetooth API to communicate with the Vitalsens device.

Tools that were used in the development included Eclipse, Android SDK, ADT plugin for Eclipse.

The result of the thesis project was an application that communicates with the Vitalsens device via bluetooth connection. The research process can be extended to a design of efficient and reliable application that can be used with the vitalsens, Vs100 device.

The thesis can be used as a reference to carry out further developments in android mobile health care solutions.

Keywords: Electrocardiograph, Software Development Kit, Android Operating System, Android Development Toolkit, Virtual Machine, User Interface, Application Programming Interface, Near Field Communication, Dalvik Debug Monitor Server

CONTENTS

ABSTRACT.....	3
CONTENTS	4
1 INTRODUCTION	5
1.1 Development and Task goals	6
1.2 Research methods	7
2 THE ANDROID PLATFORM AND VITALSENS ECG DEVICE	8
2.1 Android	8
2.2 Android Architecture	9
2.2.1 The Linux Kernel.....	10
2.2.2 Android Runtime	11
2.2.3 Libraries	11
2.2.4 Application Frame Work.....	11
2.2.5 Applications	12
2.3 Android Application and Development Environment.....	12
2.4 The Vitalsens VS100	13
3 DESIGN AND IMPLEMENTATION.....	16
3.1 Requirement specifications.....	16
3.2 Analysis and Design of the Application.....	18
3.3 Implementing the Android ECG App.....	20
3.3.1 Androidmanifest.xml file and its purpose	20
3.3.2 User Interface	21
3.3.3 Implementation of Functions	24
3.3.4 Testing.....	30
4 CONCLUSIONS	33
5 DISCUSSION	34
REFERENCES	36
APPENDIX.....	37

1 INTRODUCTION

In recent years mankind has witnessed a revolution in the smart phone industry and emerging growth in the usage of mobile applications that range from entertainment and educational apps to simple games, health care apps and more.

As the advancement in wireless sensor technology made remote monitoring easier, the number of smart phone applications utilizing this opportunity has become increasing. As a result it is now more common to see Mobile Electrocardiograph (ECG) applications that work with wireless health monitoring devices or remote control apps in our daily life.

The Android platform as part of the most widely used operating systems on smart phones and hand held tablet devices is important target for mobile application developers and hardware manufacturers.

The aim of the thesis is to introduce the android platform and develop a prototype android ECG application that works with the Vitalsens ECG monitoring device from Intelesens. The application will be used for the realization of ECG data signals that are sent from the vitalsens heart rate monitoring device via bluetooth communication.

Significant importance and contribution of mobile and wireless health solutions makes the choice for research and design of ECG monitoring apps sensible. The use of ECG mobile apps will help to monitoring heart conditions in real-time and take perspective measures.

Taking in to account, the biggest market share and wide popularity of the android OS also makes the platform the best choice from business and research perspectives. Android OS is widely deployed in different handsets and tablets around the globe.

This thesis is done on behalf of Domuset Oy; Oulu based company that is mainly involved in the research and development of wireless technologies and solutions. Throughout the years, the company has carried out different researches in the wireless health care solutions and electronic solutions in the health sector.

The ECG application can be used to monitor electric activities of the heart in a cheap, efficient and easy to use manner from the users' mobile phone or tablet. It is possible to use the application during exercises or in monitoring one's heart conditions, these indeed makes it suitable to be used as a fitness and health care solution.

I believe that the paper layouts a foundation for present and future research and development works that are carried out at Domuset Oy.

1.1 Development and Task goals

The main objective of the thesis is to design and implement a prototype ECG application for the android platform. For the development, deployment and test purpose the vitalsens ECG device (VS100) together with Samsung Galaxy S is to be used as a primary test device.

The application will act as a client that will pair and connect with the vitalsens ECG device via bluetooth technology. Visualization of real time signals that are sent from the Vital Sens device will be important functionality of the app.

In addition accuracy and efficient use of resources are important aspect of the design of the application. And therefore, the design will incorporate these aspects.

The Java programming language, Eclipse and the android Software development Kit (SDK) are used as the development tools and environment. Integrated development with the Eclipse IDE is selected for the development as it offers direct invoking of tools that are used for developing applications through the eclipse Android Development Tools (ADT) plug-in.

Use case and activity diagrams will be used to model activities and processes of the application in the requirements specification phase. The Visual paradigm UML (Unified Modeling language) is used to draw use cases and activity diagrams.

1.2 Research methods

Different articles and books are primarily used in the research and development process of the report. Manuals and information gathered from users of the vitalsens are also part of the resource that is essential for the development work. In particular the documentations of the android SDK and the Vitalsens ECG device manual are invaluable resources in the development work. Quantitative data that will be acquired from the VS100 device is used in the analysis and development work. As part of the research the information gathered from different discussions carried out with the commissioner and thesis advisor is also to be used in different parts.

2 THE ANDROID PLATFORM AND VITALSENS ECG DEVICE

2.1 Android

Referred as the first Open source mobile application development platform (Ableson, Sen, King & Ortiz 2012, 3), android OS has increased its popularity since the first SDK release back in November 2007. It was developed by Google in cooperation with the Open Handset Alliance following the acquisition of the original android Inc; a California based mobile software start up, in 2005 by Google. (Elgin 2005, date of retrieval 6.2.2012)

Android OS is a Linux based operating system that is targeted for mobile, smart phone and tablet devices. It is characterized by its openness and explicit features that enable developers to take full advantage of features of the hardware devices.

Android is a software Environment that includes a Linux based OS, different application frameworks; end user applications, libraries, integrated multimedia support and more. Tools and different APIS that are used for developing mobile apps are provided through the android SDK.

Android offers simple and powerful SDK that can be used for development in multi-platform environments. Users of different operating systems like Linux, Windows or Mac OS have the possibility for developing in any of these environments.

Android's rise to a major mobile platform was accompanied with the release of different updates and software features. The latest update to the day this paper was written is the incremental release of the Android 4.x (Ice Cream Sandwich) which was announced on Oct, 2011 and updated in January 2012 to Android 4.0.3, Revision 2 (Ice Cream Sandwich_MR1). (Android Developers, Date of retrieval 6.2.2012)

The following picture shows the distribution of different platform versions that are accessed through the android market as of February 1, 2012 based on a data collected 14 day period.

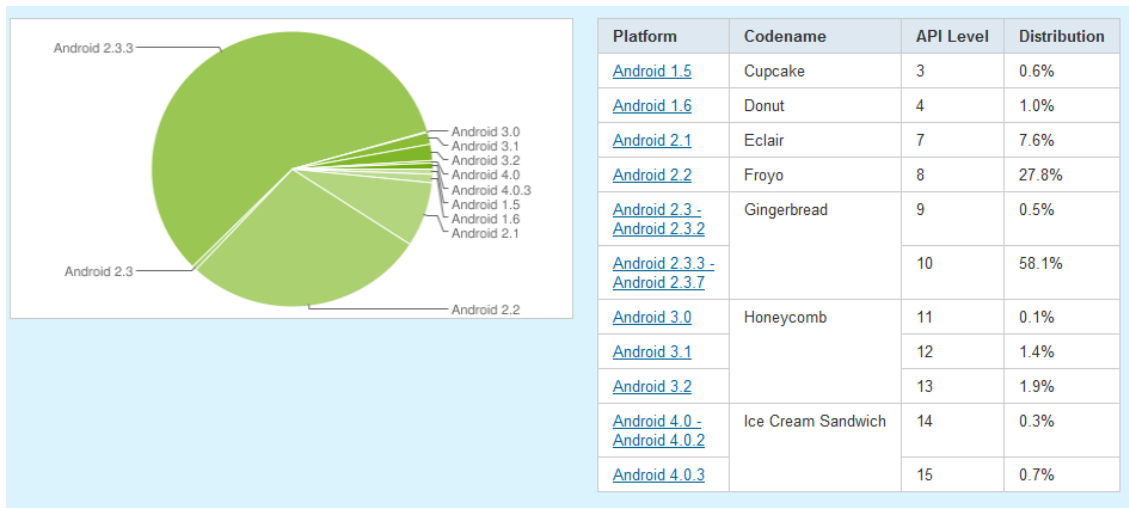


FIGURE1. Android Platform distribution

Android, in contrast to its rival Apple's IOS (iPhone OS), can be deployed to a variety of hardware devices .As a result, a variety of handsets and hand held tablets running android OS , are distributed by different mobile carriers and manufacturers that share the majority of the global market. As of February 2012 there were about 450,000 android applications available in the android market and over 10 billion downloads. (Google Play, date of retrieval 4.4.2012)

Google Play was introduced in March 2012, by merging the Android market and Google Music services together. As a result, all android apps are now accessed from Google's multimedia content service together with music, books and etc.

2.2 Android Architecture

The Android operating system follows a fault-tolerant, secured and efficient design. It consists of Core applications like browser, contact and phone, an Application Framework that enables reuse and replacement of components; different libraries for multimedia, database, security and more; and a Linux kernel that is responsible for memory management and security issues.

The main components of the android OS can be showed in different layers as shown in the next page.

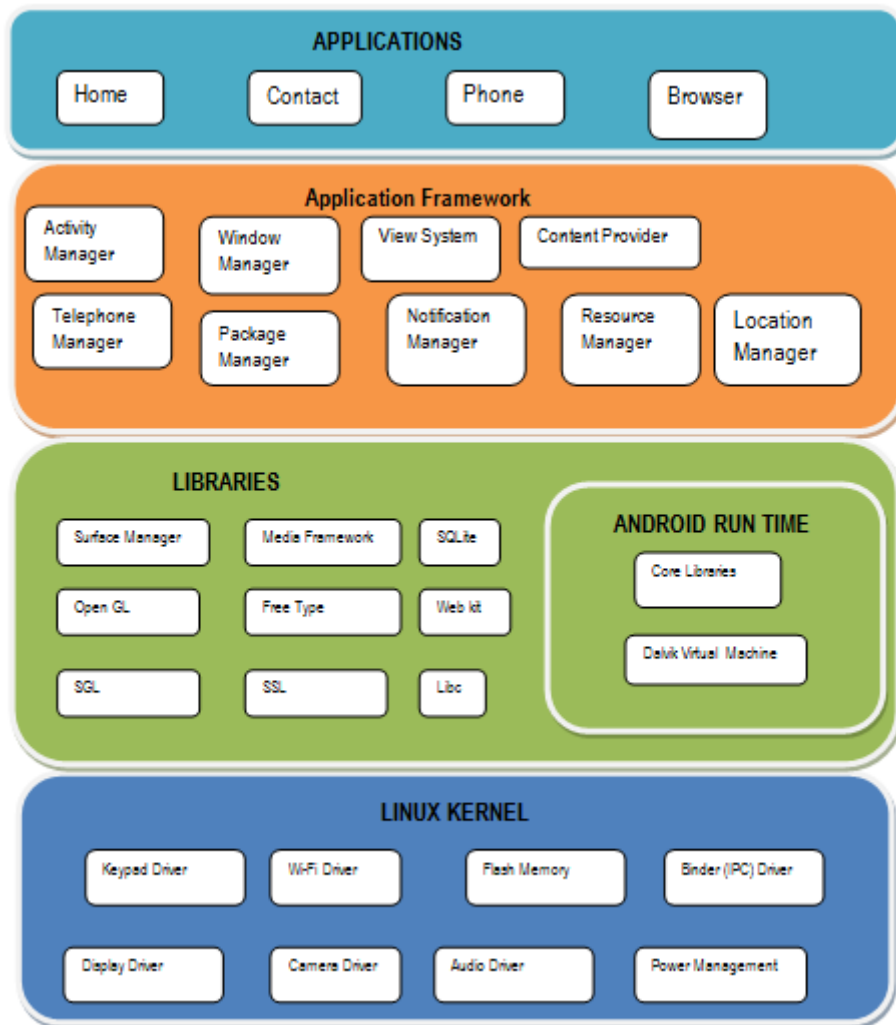


FIGURE2. Android Architecture

2.2.1 The Linux Kernel

This is part of the layer that is responsible for services like memory and process management, security, network stack, threading, camera, and audio drivers and more. It is also referred as an abstraction layer between the hardware and rest of the software stack. Android uses Linux 2.6 version for handling the above mentioned processes. (Android Developers, date of retrieval 6.2.2012.)

2.2.2 Android Runtime

The Dalvik Virtual machine and a set of core libraries that handle different functionalities found in core java libraries is included in this layer. Each android application runs separately on its own Dalvik virtual machine instance. The Dalvik Virtual Machine is a register based architecture that is designed for systems that are constrained in terms of memory and processor speed.

2.2.3 Libraries

Android contains a set of C and C++ libraries that are responsible for performance optimization and efficiency. This includes

- System C library that is an implementation of the standard C system library (libc), tuned for embedded Linux-based devices (Android Developers, date of retrieval 6.2.2012.)
- Media Libraries that support playback and recording of popular video and audio formats
- SQLite that is a lightweight relational database engine
- LibWebCore, a modern web browser engine
- SGL a 2D graphics engine, 3D libraries based on OpenGL ES 1.0 APIs, Surface manager and more

2.2.4 Application Frame Work

These are a set of managers that provide services for views, activities, content providers notifications and more. Native libraries that are responsible for performance optimization are accessed through the application framework.

Some of the components of the application framework layer are listed as follows:

- Content Providers-which manages access to a central repository data
- Resource manager-that provides access to resources like graphics ,layouts and more
- Notification Manager-which is used for displaying custom alerts
- An Activity manager-that handles the lifecycle of applications and more

2.2.5 Applications

These are a set of applications that are written in the Java programming language .Some of the examples include an email client app, SMS, browser, Contacts and more.

2.3 Android Application and Development Environment

Android applications are written in the Java language, compiled into byte codes which will be converted to a .dex file (Dalvik executable file) using the dx converter. This will further be compiled in to android package file (apk file), that can be installed on the android devices.

Taking in to account the multi user Linux system aspect of android OS, every Android application is treated as single Linux user. Each of the applications on the system is identified by a unique Linux user Id. Each process has also its own virtual machine.

Android applications are composed of one or more of the four basic application components. The main building blocks of an android application are listed and discussed as follows.

Activity

It is a core component of the android applications; Activities represent a given task that can be done by the application. It can also be represented as a single screen User interface that interacts to do functions like dialing, taking photo and so on. A single android application is composed of multiple activities. For example an android ECG app can have an activity that shows the heart rate samples and other activity for settings or splash screen.

Services

Designed to keep running in the background, services are responsible for tasks like updates and remote processes. Services can be implemented for time taking operations and processes that need to be scheduled and carried out regularly. Services do not provide UIs.

Content Providers

These are used as interface to data .A content provider helps to maintain shared data between applications. Data can be shared and modified according to permissions using the content providers. Data encapsulation and data security is provided using content providers.

Consistent and secured data access is managed and maintained using content providers and provider clients.

Broadcast receivers

These are broadcast announcements made by the system. Examples include a broadcast for announcing that battery is low or to inform an update is ready to be downloaded. Broadcast Receivers can be used to initiate services to do something.

One of the merits of developing for android is the cheap and easiness of the development environment. The Android SDK and tools are provided free of charge from the developer's site (<http://developer.android.com>). In addition to this, well organized documentations make it easy for developers to startup and dive in to the platform quickly. While developing android applications, Tools from the SDK can be invoked through command Line or the ADT. In this project, the Eclipse IDE is used for the application development .The ADT (Android Development Tools) is Eclipse plugin that is recommended for developing android applications since it offers direct invoking of tools during application development. (Android developers, date of retrieval 6.2.2012.)

2.4 The Vitalsens VS100

The Vitalsens VS100 System is designed to ease the measurement of different vital signals. The device's size and flexibility makes it comfortable to be worn and used by patients or anyone. ECG signals from the system can be used to calculate data like heart rate, skin surface temperature and motion. (Intelesens. Vitalsens Vital Signs Monitor VS100.2)

Bluetooth technology is primarily used for handling data transfer between the receiving stations (device). The VS100 uses Bluetooth Class II for the data transmission.

The Vitalsens VS100 is made up of a body-worn unit with an ECG sensor, thermistor for skin temperature, and tri-axial accelerometer for activity detection. The sensor establishes Bluetooth

connection with any Bluetooth enabled receiving station e.g. PC/laptop, PDA, Smart Phones, tablets etc. (Intelesens, 6)

The Vitalsens Front End desktop application is also provided with the system. This software enables users to monitor Heart rate signals. It displays the live heart beat readings and ECG graph of the signal. The Following picture shows live heart beat readings taken from the Vitalsens Front End software.



FIGURE3. Logged Readings of streamed ECG and heart beat signals from a user on the Vitalsens Frontend Software

The Vitalsens device is recommended to be operated between the temperatures of 0 °C and 40 °C. Adherence to the specified operational temperature range helps to avoid impaired system performances.

The Vitalsens VS100 device can be used with the Vitalsens Front End software on a laptop or pc computer; however there seems to be no existent application software for smart phones and PDAs from the manufacturers according to the discussions made with the commissioner and the thesis advisor. The ECG applications can be optimized to be used on different mobile devices smart phones and tablets.

A typical message (data) sent from the VS100 device has the following basic format.

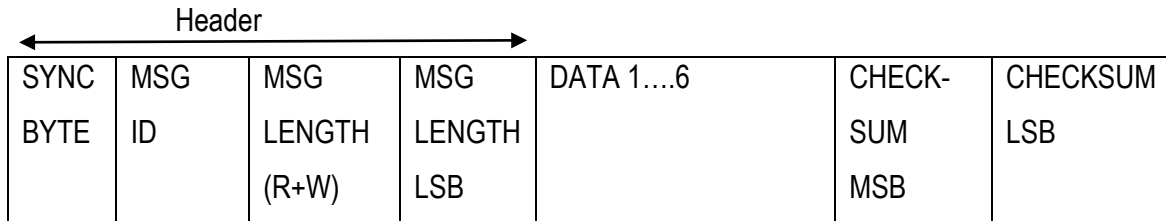


FIGURE.4. Packet format for communications protocol

Visualization of the entire VS100 vitalsens device and supporting devices that are used together can be overviewed as follows.



FIGURE.5. System overview of Vitalsens Vs100 wireless network area and possible

However, as part of the thesis project this paper will emphasis on the development of android end device application. In other words, implementation of the aforementioned system with an emphasis on mobile device (smart phone) end device.

3 DESIGN AND IMPLEMENTATION

This chapter describes the design and implementation of the android ECG application .Use cases and activities will be used to analyze the requirements and specifications, which will be followed with the actual implementation of the specified functionalities of the application.

3.1 Requirement specifications

The app is to be presented in an intuitive manner and provide easy to use functionalities to calculate the heart rate sample and display ECG signals that are sent from the vitalsens VS100 device. Bluetooth communication is used primarily to carry out the data communication between the vitalsens device and the android tablet.

A bluetooth connection to the VS100 device is initiated from the ECG application on the android device while the vitalsens device is turned on. In case bluetooth is not supported on the user's android device, the application will exit with a notice. However, if the device has bluetooth capability, user will be prompted to connect with the vitalsens device.

After authentication and establishing the bluetooth connection the application will be directed to choose the next activity to display heart beat.

Use Case

The user activities are depicted as follows using UML tool for the use case.

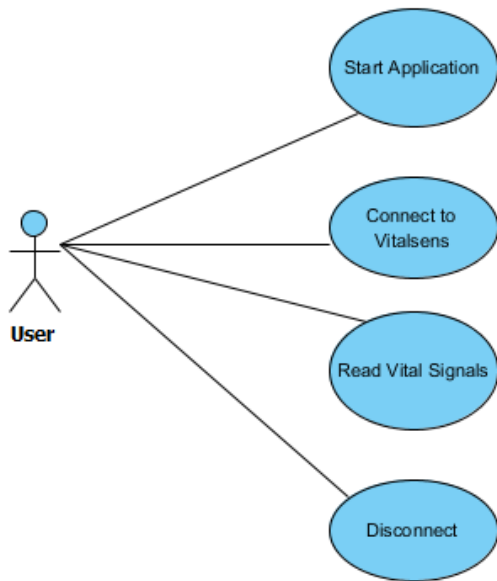


FIGURE.6. Use Case Diagram of the application

The Use case- Start Application, represents launching the application which is the first activity. The connect to Vitalsens use case depicts the action taken for bluetooth device discovery and query functionalities which will lead to establishment of RFCOMM bluetooth communication. In addition, the use cases, Read vital signals and Disconnect will be defining data reading and disconnect functionalities in the application.

TABLE1. Start Application use case

Name	Start Application
Actors	User
Pre-Condition	Android device and bluetooth support
Description	A start up (First) activity ,welcoming activity

TABLE2. Connect to Vitalsens use case

Name	Connect Vitalsens
Actors	User
Pre-Condition	User starts Application
Description	Prompting for device discovery and connecting

TABLE3. Read Vital signals use case

Name	Read Vital signs
Actors	User
Pre-Condition	-Authentication PIN code(0000) -Bluetooth communication is established
Description	Initiating to read data sent from the Vitalsens

TABLE4. Disconnect use case

Name	Disconnect
Actors	User
Pre-condition	Bluetooth connection is established
Description	User can initiate disconnection to the vitalsens By exiting app

The design of the system is constrained to a simple demonstration of the android ECG application, and therefore the functionalities of the application and activities have been maintained to match these aspects.

However, it is acknowledged to comprise additional functionalities in future works. This can be achieved by integrating data like GPS and more.

3.2 Analysis and Design of the Application

The Overall architecture of the system can be visualized to comprise a Patient or user of the application, the Vitalsens wearable heart monitoring device, the android 2.3.3 running device. The patient or user of the application is supposed to be wearing the Vitalsens device on the Chest. After turning on the device, the user should proceed by launching the BluetoothHr application from his or her device.

When bluetooth connection is established with the device the user should be able to see and analyze the real time heart beat rate and ECG graph on their mobile device.

The application acquires real time data sent from the Vitalsens device and analyzes it with the appropriate algorithm and presents user with the real time signals. The following activity diagram shows simplified overall activities in the application.

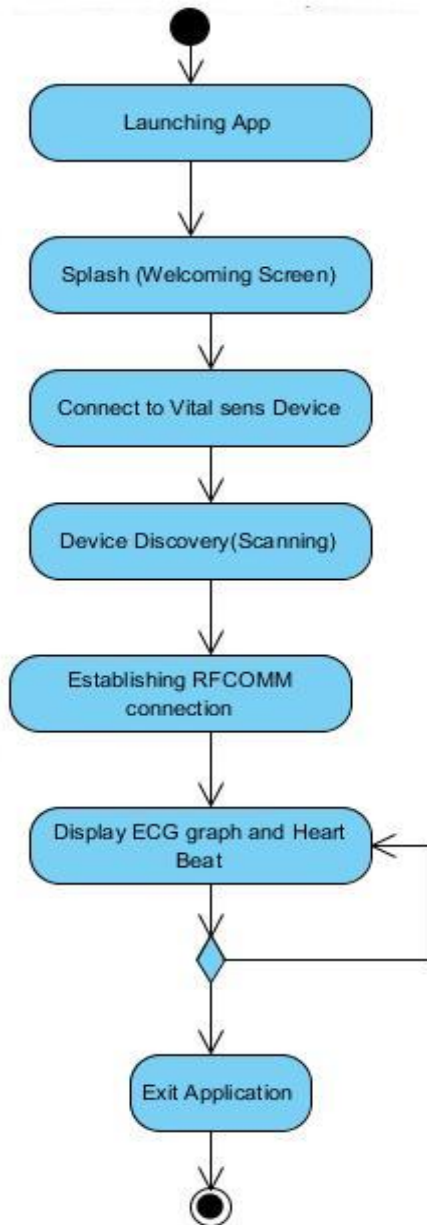


FIGURE.7. Activity Diagram of the application

The application launches from the apps category on the phone and a splash or welcoming screen is presented to the user. Following that activity to initiate connection between the phone and the VS100 is started, accompanied by device discovery and bluetooth RFCOMM establishment.

The next activity would be to display the ECG graph and heart rate data, at this stage the user has an option to exit or stay using the application.

3.3 Implementing the Android ECG App

This section describes the implementation of the android ECG application. The different resource and source files that make up the application are explained here.

3.3.1 Androidmanifest.xml file and its purpose

The Android manifest .xml file depicts important information of the application to the Android System that is required by the system before it can run any of the applications code. Different components of the application like activities, services, content providers and etc are defined in the androidmanifest.xml file. It is also used to declare different kinds of permission such as bluetooth access, camera access, NFC (Near Field Communication) access and so on. Minimum API for applications is also declared in the manifest file. In the BluetoothHr application the android manifest is used to set the API level of the application and to enable bluetooth and Bluetooth Admin permissions.

In addition, the different activities of the application are also defined in the manifest file. The following is a code snippet of the Android manifest file of the application.

```
<?xml version="1.0" encoding="utf8"
?><manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.woubshet.bluetoothhr" android:versionCode="1"
    android:versionName="1.0" ><uses-sdk android:minSdkVersion="10"
/><uses-permission android:name="android.permission.BLUETOOTH"
/><uses-permission android:name="android.permission.BLUETOOTH_ADMIN"
/><application android:icon="@drawable/ic_launcher"
    android:label="@string/app_name" > <activity android:label
="@string/app_name" android:name=".BluetoothHr" ><intent-filter >
<action android:name="android.intent.action.MAIN" /> <category
android:name="android.intent.category.LAUNCHER" /> </intent-filter>
</activity> <activity android:name=".HeartMn"
    android:screenOrientation="portrait">
<intent-filter><category
    android:name="android.intent.category.LAUNCHER" />
```

FIGURE.8.Code Snippet of the androidmanifest.xml

3.3.2 User Interface

This section describes the user interface implementation of the android mobile application .In Android application development the View and View Group objects are used to build the user interface.

A view object represents data structure whose properties store the layout parameters and content on a given specific area of the Screen. (Android Developers, date of retrieval 11.3.2012)

The layout for different activities in an application can be defined using the XML based layout file.

The application is comprised of three activity classes and a service class. The first activity class that is implemented as the welcoming screen is represented by the main lay out view. The main View is defined in the main.xml layout file of the resource of the application. The file resides in the “resources/layout” folder.

The following picture shows the main layout view that represents the starting point of the application.

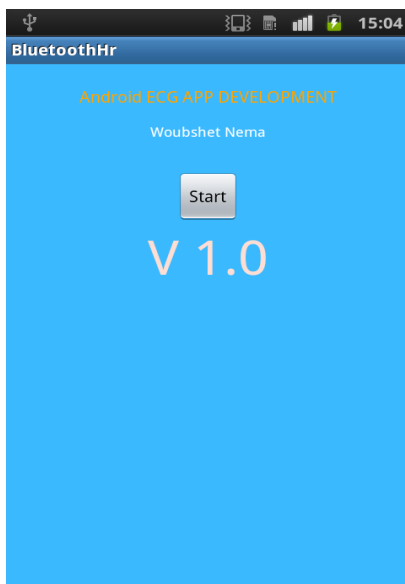


FIGURE.9 .The main layout view of the application

The user interface of this layout consists of a linear layout component, three text views and a button widget that is used to start the application and proceed to the next activity. Different properties

like gravity, background-color, orientation and etc are used to adjust the interface according to our preference. The values for the elements have been defined in the strings folder of the resource files as strings.xml.

The following is a code snippet of the XML file that is used in the main.xml of the main layout view.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:background="#3BB9FF"
    >
    <TextView
        android:layout_width="wrap_content" android:layout_height="34dp"
        android:layout_marginTop="20dp"
        android:layout_gravity="center_horizontal"
        android:text="@string/title"
        android:textColor="#FFA500"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="45dp"
        android:layout_gravity="center_horizontal"
        android:text="@string/author"
        android:textColor="#ffffff" android:textSize="12dp"/>
    <Button
        android:id="@+id/start" android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
```

FIGURE.10. The Code snippets from main.xml layout file

After clicking the start button on the main layout view users will be directed to the next activity which is the HeartMn.java class that is represented with the Customs.xml layout view. Depending on the availability of bluetooth on the user's device and whether or not bluetooth is turned on or off, the user will be directed to either of the following screens.

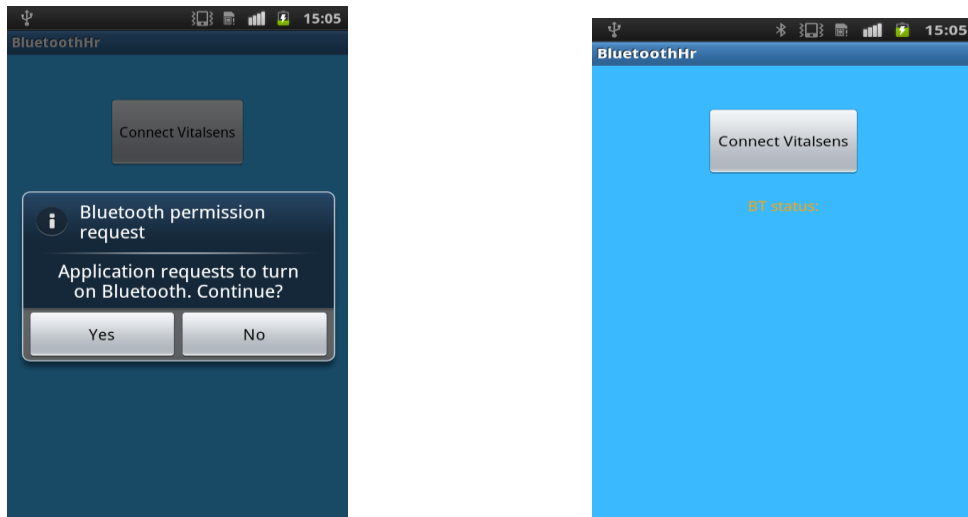


FIGURE.11. The Customs.xml layout

Intent in the HeartMn.java class is used to initiate the request for bluetooth permission in case the bluetooth adapter on the device is turned off at the start of the application. The customs.xml layout file mainly consist a button widget and textView widget that is used to display the bluetooth status. After granting permission to turn on the bluetooth adapter the user is presented with the ConnectVitalSens button. The ConnectVitalSens button is used to initiate bluetooth communication with the vitalsens device.

The last two layout views that are used in the application are the device_list.xml and device_name.xml layout files. These two layouts are used to display a list of discovered devices during bluetooth device (the Vitalsens) search and bluetooth device name respectively.

The device_List.xml layout file is used together with the DeviceList.java class to display and perform device query and listing of paired and available devices.

The device_List.xml file, the device_name.xml and the deviceList.java class are all used under the Apache License (<http://www.apache.org/licenses/LICENSE-2.0>) from the android open source project Bluetooth Chat sample file.

3.3.3 Implementation of Functions

This section explains the implementation of android bluetooth communication with the vital sens device and shows how classes and interfaces of the bluetooth API are used in the application. In addition, classes and components of the application are explained.

The First activity of the application is represented with the BluetoothHr.java class .A code snippet from the class shows how the layout for the activity is loaded with the onCreate () method call as follows. The view for the activity is set using the setContentView () method. Here the main.xml layout is set as the UI of the activity. The setOnClickListener will attach the start button on the activity using the OnClickListener interface that responds to a call back method onClick when clicked. This method will start the HeartMn.java class which is the next activity.

```
public class BluetoothHr extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Button btStart = (Button) findViewById(R.id.start);
        btStart.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent startIntent = new
                Intent(v.getContext(),HeartMn.class);
                startActivityForResult(startIntent, 0);
            }
        });
    }
}
```

FIGURE.12.Code snippet from BluetoothHr.java class

The HeartMn.java class represents the second activity in the application and is responsible for the majority of functionalities of the app. It extends the activity class and utilizes classes in the android bluetooth package to manage bluetooth functionality.

Bluetooth communication is implemented following the bluetooth permission declarations inside the androidmanifest.xml file.

As the first step, the class checks the availability of bluetooth Adapter on the device and ensures that it is enabled. In case bluetooth is not supported by the device the app will exit, however if bluetooth is available and disabled a bluetooth enable request is made to turn on bluetooth. A snippet from the class shows how bluetooth availability is checked as follows.

```
public void onCreate(Bundle savedInstanceState) {  
  
    super.onCreate(savedInstanceState);  
    // Set up the window layout  
  
    setContentView(R.layout.custom);  
  
    // Get the local blue-tooth adapter  
  
    mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();  
    // If adapter is null then Blue-tooth is not supported  
    if (mBluetoothAdapter == null){  
        Toast.makeText(this, "Bluetooth is not Supported",  
  
            Toast.LENGTH_LONG).show();  
    }  
}
```

FIGURE.13.Code Snippet of the onCreate method

While starting the app the isEnabled () method is called to check if the bluetooth is enabled. If bluetooth is disabled, a startActivityForResult () method is called using the action intent for ACTION_REQUEST_ENABLE.

If bluetooth is enabled the setupEcg () method is also called to initialize the mHrService, service class.

In addition the ConnectVitalSens button is attached with the onClick Listener to invoke selectBtDevice () method.

A handler method that is responsible to handle bluetooth connection state is also implemented in this class. The method also manages data byte that is sent from the vitalsens device upon bluetooth establishment.

The HrService.java class is mainly responsible for bluetooth connection and threading purposes. A handler in this class is used to update the User interface activities in the HeartMn.java class activity. Some of the methods that are included in this class are, connect, connectThread, connect-ed and connectedThread methods.

The connect method is responsible for initiating bluetooth connection with a remote device. In order to achieve this, the class first check and cancels any threads that are attempting to make a connection and threads that have already established a connection. As the next step the method starts a thread to connect with the selected device. The following snippets show part of the connect method.

```
// Cancel any thread attempting to make a connection
if (mState == STATE_CONNECTING) {

    if (mConnectThread != null) {

        mConnectThread.cancel();
mConnectThread = null;
    }
}
// Cancel any thread currently running a connection
if (mConnectedThread != null) {
    mConnectedThread.cancel();
    mConnectedThread = null;
}
// Start the thread to connect with the given device
```

FIGURE.14.Code Snippet from the Connect method

The connectThread method extends Thread and is responsible for establishing bluetooth connection with a given bluetooth device. The connectThread method first gets bluetooth socket on the user's android device and then initializes the socket to connect with the vitalsens device. A universally unique Id (UUID) that is commonly used in creating RFCOMM connection is chosen in this application. A UUID is described as an Identifier that is used in software development. The UUID used in the application has a string value of " 00001101-0000-1000-8000-00805F9B34FB".

The `connectedThread` extends `Thread` class and handles bidirectional communication between the application and the vitalsens monitoring device. This thread is executed in the existence of bluetooth connection between the application and the vitalsens device.

The `DeviceListActivity.java` class makes part of the application that is responsible for handling the list of paired and detected devices. The activity sets up a window for the list of discovered and paired devices.

A code snippet from this class is illustrated as follows. Array Adapters are initialized for both discovered and paired devices, and are shown using the respective list View.

```
// Initialize array adapters. One for already paired devices and
// one for newly discovered devices
mPairedDevicesArrayAdapter = new ArrayAdapter<String>(this,
                                                    R.layout.device_name);
mNewDevicesArrayAdapter = new ArrayAdapter<String>(this,
                                                  R.layout.device_name);

// Find and set up the ListView for paired devices
ListView pairedListView = (ListView) findViewById(R.id.paired_devices);

pairedListView.setAdapter(mPairedDevicesArrayAdapter);
pairedListView.setOnItemClickListener(mDeviceClickListener);

// Find and set up the ListView for newly discovered devices
ListView newDevicesListView = (ListView) findViewById(R.id.new_devices);

newDevicesListView.setAdapter(mNewDevicesArrayAdapter);

newDevicesListView.setOnItemClickListener(mDeviceClickListener);
```

FIGURE.15.Code snippets from the `devicelist.java` class

The next and last activity is the `WaveformView.java` class which is a `View` class that is responsible for real time plotting of ECG signals and displaying the heart rate that are sent from a connected Vital sens device. A Line graph can be used to plot the ECG signals with the `canvas` class. Data bytes are read and analyzed to plot the appropriate ECG signals and Heart rate.

The use of appropriate ECG algorithm is vital for implementing accurate data reading from the Vitalsens device. Further study should be done and implemented to acquire accurate results that will match the corresponding values read with the Canberra desktop application of the VS100 device.

The android developers website offers instructions and sample applications on bluetooth application development. The Bluetooth Chat sample was found helpful in learning about the android bluetooth class.

The following diagrams show screen captures taken from the application while running on the Samsung Galaxy S phone. Screen captures were made on Dalvik Debug Monitor Server (DDMS) Perspective view in the Eclipse environment using, Device => Screen Capture tool.

The DDMS is Android debugging tool that is integrated with Eclipse to provide screen captures from android device or emulator and services like port forwarding, thread, logcat and more information.(Android Developers, Date of retrieval 4.9.2012)

The pictures show user interfaces and functional implementations of the activities and classes discussed in the above paragraphs.

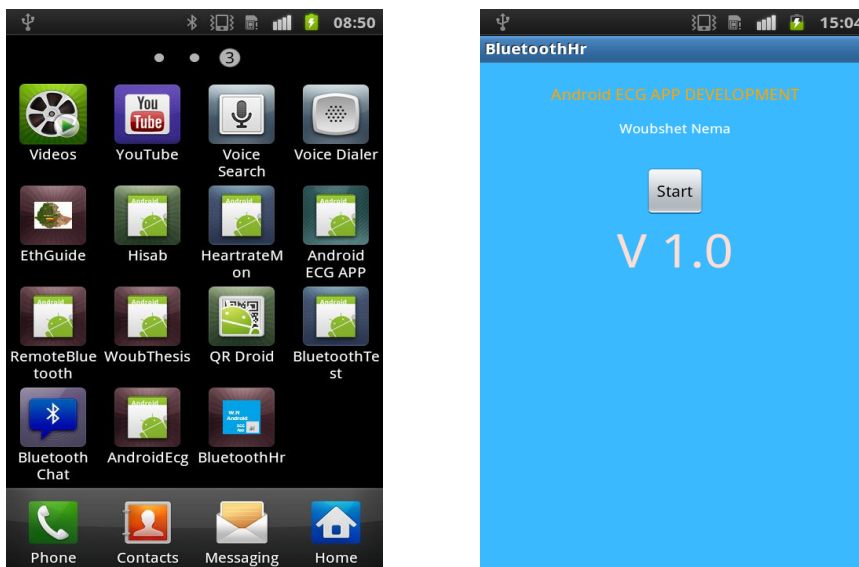


FIGURE.16. Screen capture of Application Installed on the Android device and after launching

After starting the application the user is directed to the HeartMn.java activity which is shown in the next diagram. The screen shot shows the Bluetooth Permission request dialogue being presented to the user whenever the bluetooth adapter is not enabled on her/his device.

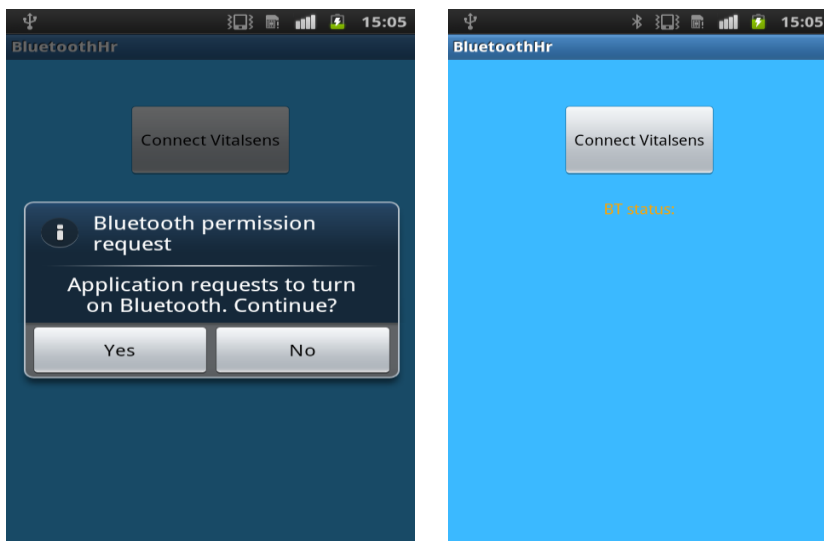


FIGURE.17. Screen Capture of Bluetooth Permission request and HeartMn.java activity

The user will be preceded with the following screens while initiating connection with the VS100, Vitalsens device. Bluetooth pairing is done using the “0000” PIN Code.

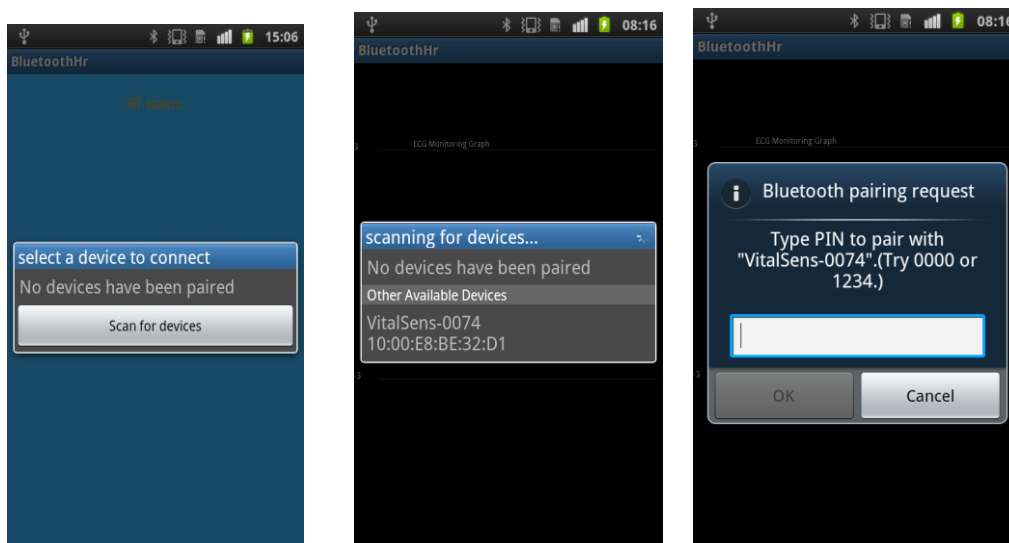


FIGURE.18. Device Scanning and Pairing

Right after the connection is established the application starts reading byte values sent from the device and pass values to the waveform view.

3.3.4 Testing

Android Mobile application testing can usually be done inside the development tools using the emulators and further tested using hardware devices. However when it comes to bluetooth applications it is required to carry out the development and testing on the specific hardware model under target. And It is a must to have the android device.

In this Project ,the black box testing is used to carry out the mobile application testing for the development work .Black Box testing is described as a method of software testing that is focused on functionalities of an application.It is defined as a software testing technique whereby the internal structures (workings)of the item under test are not known to the tester. (Webopedia .Date of retrieval 04.08.2012)

The tests carried out extend from installation of the application on android 2.2.3 Samsung Galaxy phone,and bluetooth functionalities like connectivity to reading ECG.

The following tables show a list of tests carried out on the application.

TABLE.5. Test Case- Installation

Test ID 1.1	Test Title Installation
Test Description	The .apk file must install on the device
Testing steps 1.Open browser on the device and type the Url of the application(http://www.students.oamk.fi/~k8newo00/BluetoothHr.apk) 2. .Connect the typed Url and download application 3. Carry out installation	Result 1.Application installation Was successful
Result	PASS, Successful

TABLE 6. Test Case-Launch Time

Test ID 1.2	Test Title Life Cycle (Launching time)
Test Description	Checking Launching time ,there should be no delays(should launch in a second)
Result	PASS, Successful

TABLE7. Test Case-Application Launch and Bluetooth Permission

Test ID 1.3	Test Title Application Launch and Bluetooth permission
Test Description 1.Application Should exit if bluetooth is not Supported on device	2.Application should prompt a request incase bluetooth Adapter is turned off and supported on the device
Result	PASS , Successful

TABLE8. Test Case-Device Query(Scanning)

Test ID 1.4	Test Title Device Query and scanning
Test Description 1.Application should be able to scann bluetooth devices in the area	2.should list devices that are paired and Found in the surrounding area
Steps 1.Launch App 2.Accept bluetooth Permission if bluetooth was not enabled	3. Press start button from welcoming screen 4.Start query by pressing Connect Vitalsens button
Result	PASS , Successful

TABLE9. Test case-Bluetooth Connection with Vitalsens

Test ID 1.5	Test Name Bluetooth establishment with the Vitalsens device
Test Description Bluetooth Connection between the Vitalsens and Samsung Galaxy should be established	
Steps 1.press connectVitalsens device button and start querying for bluetooth device	2.From the query result select Vitalsens device and connect 3.Enter PIN code "0000" for establishing bluetooth connection
Result	PASS ,Successful

The final test that was carried out is ECG data and heart beat rate analysis test, however the attempts made to plot the ECG graph and calculate heart beat were not able to reflect expected outcomes.

In addition, while carrying out the test it is also important to consider multitasking issues which are great factors in the daily use of the application. For example, the app should be able to work normally in case there is incoming dial or text message or whenever the user is running multimedia or other applications simultaneously.

While developing mobile applications, it is necessary to carry out tests that check the performance of basic applications of the phone while running our new app. In this case, the basic applications from the device should not experience any disruption.

4 CONCLUSIONS

The advancement in mobile and wireless health care solutions is contributing in different aspects of our lives that range from diagnosis to treatment of different diseases like cardiovascular disease. Android applications are also a part of diverse solutions that are offered as healthcare solutions together with a variety of health monitoring devices.

The thesis project has covered mobile application development in the android Platform, basic components of android applications and showed the overall architecture of the android OS. The result of the project has achieved part of the objectives that were set at the beginning. However, the scope has been limited to establishing bluetooth communication with the vital sens device and receiving data. Appropriate algorithm for heart beat calculation and detailed research on the VS100 device should be adapted for future work.

The bachelor's thesis was an opportunity to extend and discover new skills in Mobile application development. The final result of the application can also be adapted as a reference for bluetooth application development in the android platform. The Paper lays out a foundation for future developments that are focused on wireless health solutions in the android platform at DOMUSET OY.

5 DISCUSSION

The thesis project gives an insight in android application development and particularly emphasizes on android bluetooth communication and ECG analysis. The final result of the project was an application that can be improved and developed further in the future.

It has also been part of a great learning process, as there was the need to learn about the android platform and development Environment and tools. This was all accompanied with self studies, tutorials and help from groups and forums like stackoverflow.com. Suggestions and guidance from the thesis supervisor and commissioner were also invaluable and helpful along the way.

Taking into considerations time and resource limitations the scope of the paper was set to a simple demonstration prototype of ECG for the Vitalsens. However, the research can be improved to develop a fully functional app that contains different features optimized for a better performance. Some of the features that can be included are integrated SMS (texting) functions, options for saving the data on the device or data streaming function that can work with a control server in a health center. These features will help clinicians to monitor patients efficiently and therefore improve service.

It is also important to include more options like zooming functionality, which will improve the usability of the app. In addition, the application can be customized to offer more business opportunities. For example; it is possible to add GPS tracker features that can be used during exercises like cycling and running. These can be accompanied with services from a web server at DOMUSET OY, where by users will be able to log their data and track their records.

I recommend that future developments include team of professionals with expertise from different fields in order to develop a successful application. During the development process of the project, I was often facing challenges that result from minor software update issues in the development environment to problems that were caused by programmatic errors.

While considering the target devices running newer Versions of the android OS, like the ice-cream Sandwich (Android 4.0.x), the BluetoothHealth class should be used for development purposes. Bluetooth Health Device profile is officially supported in these newer versions making it easier to create applications for health devices that communicate with bluetooth, such as heart rate monitors and blood meters. This class makes development easier, providing most functionalities and services common to health devices. The support for Bluetooth Health device (BHD) devices is available for Android 4.0 (API level 14) and newer versions.

REFERENCES

Abelson, W.F., Sen, R., King, C. & Ortiz, C.E. 2012. Android in Action. 3rd edit. Shelter Island: Manning.

Android Developers.2012. ADT Plugin for Eclipse. Date of retrieval 6.2.2012.
<http://developer.android.com/sdk/eclipse-adt.html>

Android Developers 2012. Developing. Date of retrieval 6.2.2012.
<http://developer.android.com/guide/developing/index.html>

Elgin,B. 2005. Google Buys Its Mobile Arsenal. Date of retrieval 6.2.2012.
http://www.businessweek.com/technology/content/aug2005/tc20050817_0949_tc024.htm

Intelesens. 07.03.11. Intelesens Communications Protocol. V1.2/0311

Intelesens. Vitalsens Vital Signs Monitor VS100.UserGuide Version 070311

Webopedia.com. Date of retrieval 04.08.2012.
http://www.webopedia.com/TERM/B/Black_Box_Testing.html

Android Developers 2012.Developing.Debugging.Using DDMS. Date of retrieval 4.9.2012
<http://developer.android.com/guide/developing/debugging/ddms.html>

GooglePlay.About Google Play. Date of retrieval 4.4.2012
https://support.google.com/googleplay/bin/answer.py?hl=en&p=play_faq&answ

APPENDIX

//====The AndroidManifest.xml =====//

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.woubshet.bluetoothhr"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="10" />
    <uses-permission android:name="android.permission.BLUETOOTH"/>
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:label="@string/app_name"
            android:name=".BluetoothHr" >
            <intent-filter >
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

```

```

        <activity
            android:name=".HeartMn"
            android:screenOrientation="portrait">
    <intent-filter>
    <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
        </activity>

        <activity android:name=".DeviceListActivity"
            android:label="@string/select_device"
            android:configChanges="orientation|keyboardHidden"
            android:theme="@android:style/Theme.Dialog"> </activity>
    </application>

</manifest>

```

XML layout Files

```

//====The main.xml layout file =====//
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:background="#3BB9FF"
    >
    <TextView
        android:layout_width="wrap_content"
    android:layout_height="34dp"
        android:layout_marginTop="20dp"
        android:layout_gravity="center_horizontal"
        android:text="@string/title"

```

```

        android:textColor="#FFA500"/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="45dp"
    android:layout_gravity="center_horizontal"
    android:text="@string/author"
    android:textColor="#ffffff" android:textSize="12dp"/>

<Button
    android:id="@+id/start" android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:gravity="center" android:text="Start"/>

<TextView android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:text="@string/version"
    android:textColor="#fcded4" android:textSize="43dp"/>

</LinearLayout>

```

//====The Customs.xml layout file ====//

```
<?xml version="1.0" encoding="utf-8"?>
```

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
        android:background="#3BB9FF" >

    <Button

        android:id="@+id/btnConnect"
        android:layout_width="wrap_content"
        android:layout_marginTop="45dp"
        android:layout_height="73dp"
    android:layout_gravity="center_horizontal"
    android:text="@string/btnConnect"/>

    <TextView android:layout_width="wrap_content"
        android:id="@+id/txt_btstatus" android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
    android:layout_gravity="center_horizontal" android:textColor="#FFA500"
        android:text="@string/status_Label"/>

```

```

</LinearLayout>

```

```

//=====The device_list.xml=====//

```

```

<?xml version="1.0" encoding="UTF-8"?>

```

```

<!-- Copyright (C) 2009 The Android Open Source Project Licensed under the
Apache License, Version 2.0 (the "License"); you may not use this file except
in compliance with the License. You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law
or agreed to in writing, software distributed under the License is distribut-
ed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express or implied. See the License for the specific language governing per-
missions and limitations under the License. -->

```



```

<LinearLayout android:layout_height="fill_parent"
android:layout_width="fill_parent"
android:orientation="vertical"
xmlns:android="http://schemas.android.com/apk/res/android">
    <TextView android:layout_height="wrap_content"
        android:layout_width="fill_parent"
        android:paddingLeft="5dp"
        android:textColor="#fff"
        android:background="#666"
        android:visibility="gone"
        android:text="@string/title_paired_devices"
        android:id="@+id/title_paired_devices"
        />
    <ListView android:layout_height="wrap_content"
android:layout_width="fill_parent"
        android:id="@+id/paired_devices" android:layout_weight="1"
        android:stackFromBottom="true"/>
    <TextView android:layout_height="wrap_content"
        android:layout_width="fill_parent"
        android:paddingLeft="5dp"
        android:textColor="#fff"
        android:background="#666"
        android:visibility="gone"
        android:text="@string/title_other_devices"
android:id="@+id/title_new_devices"/>
        <ListView android:layout_height="wrap_content"
android:layout_width="fill_parent" android:id="@+id/new_devices"
android:layout_weight="2" android:stackFromBottom="true"/> <Button
android:layout_height="wrap_content" android:layout_width="fill_parent"
        android:text="@string/button_scan" android:id="@+id/button_scan"/>

</LinearLayout>

```

Values

```
//====The Strings.xml ====//
```

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <!-- main.xml -->
    <string name="title"> Android ECG APP DEVELOPMENT </string>
    <string name="app_name">BluetoothHr</string>
    <string name="author">Woubshet Nema</string>
    <string name="version">V 1.0</string>
    <string name="start"></string>

    <!-- custom.xml -->
    <string name="btnConnect">Connect Vitalsens</string>

    <!-- string for connection status -->
    <string name="not_connected">You are not connected to device.</string>
    <string name="title_connected_to">connected to: </string>
    <string name="title_connecting">connecting.</string>
    <string name="title_not_connected">not connected</string>
    <string name="status_label">Bluetooth Status</string>
    <string name="bt_not_enabled_Leaving">Bluetooth was not enabled.
Leaving Application.</string>

    <!-- DeviceListActivity -->
    <string name="title_paired_devices">Paired Devices</string>
    <string name="title_other_devices">Other Available Devices</string>
    <string name="button_scan">Scan for devices</string>
    <string name="select_device">select a device to connect</string>
    <string name="none_paired">No devices have been paired</string>
    <string name="scanning">scanning for devices</string>
    <string name="none_found">No devices found</string>
</resources>

```