Wireless ECG Software Team

by

Dylan Griff, V00711902 - dcgriff@uvic.ca
Kurt Matzanke, V00732379 - kurtm@uvic.ca
James Parry, V00682552 - jdparry@uvic.ca
Peter Scholten, V00702726 - peter91@uvic.ca

URL: www.web.uvic.ca/~mchester/
Supervisor: Xiadai Dong
Group: 12
Due: December 3, 2012

Table of Contents:

1.1 Introduction.. 1
1.2 Project Goals.. 1

2.1 The Electrocardiogram – an Overview.. 2
2.2 The Normal ECG and its Characteristics 4
2.3 The Classification of Common ECG Irregularities... 5

3.1 The Software Solution.. 6
3.2 How to Receive and Analyse the ECG Data... 7
3.3 Program Adaptability.. 8
3.4 Responding to the Determined Situation... 9

4.1 The Limitations and Purpose of Using a Smartphone....................................... 10
4.2 Mobile Device Platform Comparison... 11

5.0 Conclusion... 14

6.0 References... 15

Appendix A: Algorithm Code Outline... 16
Appendix B: iPhone 5 Specifications.. 19
Appendix C: Galaxy S III Specifications.. 23

Summary:

The purpose of this project is to investigate the development of the software required for a wireless Electrocardiogram Device. The motivation for this project is to allow for twenty four hour ECG monitoring for a person with known heart problems. This constant monitoring is desirable because, if a major heart complication occurs medical help can be sent immediately. Also, valuable data on the behavior of the heart leading up to the incident can be recorded so that researchers can learn to spot warning signs and stop complications from ever happening.

The first area of research in this project was determining what an Electrocardiogram signal looked like, and how it corresponded to the behavior of the heart. Common heart conditions were analyzed to determine what they looked like on an ECG graph and an algorithm was created to identify the key variables on the graph. These variable values were then compared with the values indicating various heart conditions to determine the patient's current heart status.

It was found that to fully analyze the ECG signal it would require more computing power than a smartphone's processor could provide. At this point it was decided that only a partial analysis of the ECG signal would be completed and only some of the more serious heart anomalies that required immediate medical assistance would be searched for. To make up for this it was decided that the complete ECG signal would be recorded so that doctors and researchers could fully analyze the data at a future date.

Two different smartphone app development options were investigated, iOS and Android. It was found that development of the ECG software on the Android platform was more desirable for multiple reasons. One of the main reasons is that iPhones - which use iOS - do not have the option to have external SD cards. This is undesirable for recording the ECG signal because the data would need to be saved to the phone's internal storage which is untenable. The other main reason for choosing the Android platform is that it provides the option to run the app as a 'service', which allows it to run constantly in the background while iOS does not provide this option to developers. After taking these factors into consideration, developing the app for Android is the recommended option.

1.1 Introduction:

A wireless Electrocardiogram Device, referred to as an ECG, that can monitor a person's heart anywhere at anytime is a valuable tool. If someone has a heart attack or other heart irregularity and they are unable to call for an ambulance, the wireless ECG can use their smartphone to send an alert to a nearby hospital. They can then be tracked down using the GPS built into the phone. Not only will a device like this directly save lives, it will also save more lives indirectly.

Currently, if someone has a heart attack that requires a trip to the hospital, there is no ECG data outlining what happened leading up to the heart attack. If a person is being constantly monitored then this data can be captured and studied. This could provide doctors and researchers with valuable information to help proactively detect and prevent heart attacks before they happen. Unless a person is already in the hospital being monitored by an ECG - there is no current way to capture this vital data.

This report discusses the plausibility of designing a wireless ECG machine and its everyday operation in a real-world environment. There are many factors to consider when designing an ECG machine but this submission focuses more on the mobile device and app software that will be used in conjunction with the hardware designed by the other team in the project. Below you will read an outline of what types of signals we will be analyzing, an overview of how the app will read and analyze the data, and a discussion concerning mobile devices with their corresponding operating system platforms.

1.2 Project Goals:

The goal for our proposed project is to design the software side of a wireless ECG device that can interact with a smart phone. The intent is to allow a person to be monitored at all times and contact the necessary party (hospital, family member, etc.) in case of a medical emergency. The ECG data can either be stored directly on the phone or sent to a medical facility to be reviewed later by a doctor.

In order to achieve these larger and broader goals, we must first accomplish goals that are smaller and more specific. We need to establish the nature of the signal we're receiving and determine how to read and analyze the received ECG signal. We will need to determine what indicates an irregularity and what characterizes a normal safe heartbeat. Finally, determining the appropriate response depending on the results is very important because if there is an emergency a medical facility will need to be contacted. Otherwise, we will need to determine what to do with the resulting ECG data.

2.1 The Electrocardiogram – an Overview:

Electrocardiography is the non-invasive medical procedure for measuring the electrical conductivity of the heart. The record of this measurement is known as an electrocardiogram (or ECG). The result is a measure of voltage difference plotted against time. The difference is created by providing two or more electrodes (or leads) positioned on the surface of the body. The ECG is used to investigate many types of heart function abnormalities.

The basic premise for the modern ECG was invented in 1901 by Willem Einthoven. Using containers of electrolyte solution, Einthoven was able to record the prominent electrical deflections related to the heart beat. He named these waveform deflections the P, Q, R, S, and T waves. For this invention he was awarded the 1924 Nobel Prize in medicine.

Modern ECG tests are comprised of 12 leads although only 10 physical leads are used. Two additional leads are created by crossing the polarity of the arm and leg leads. The placement of these leads is found in the following figure:

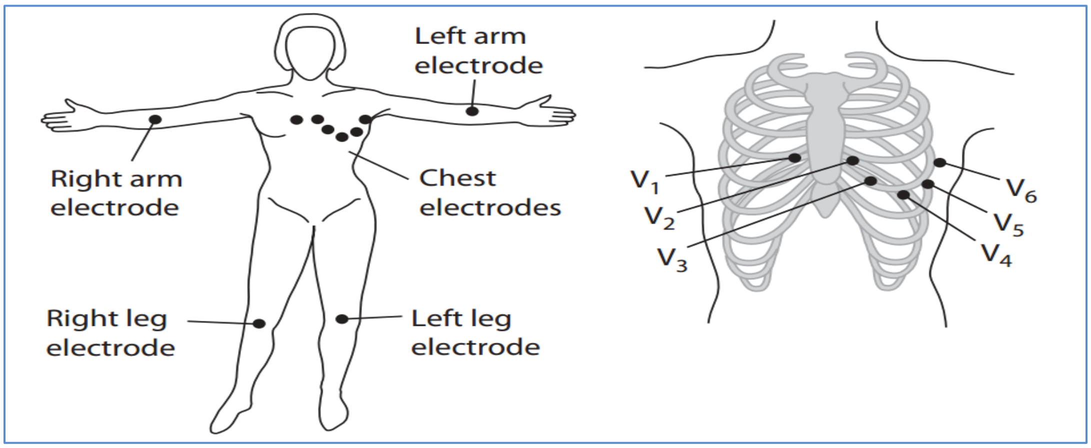
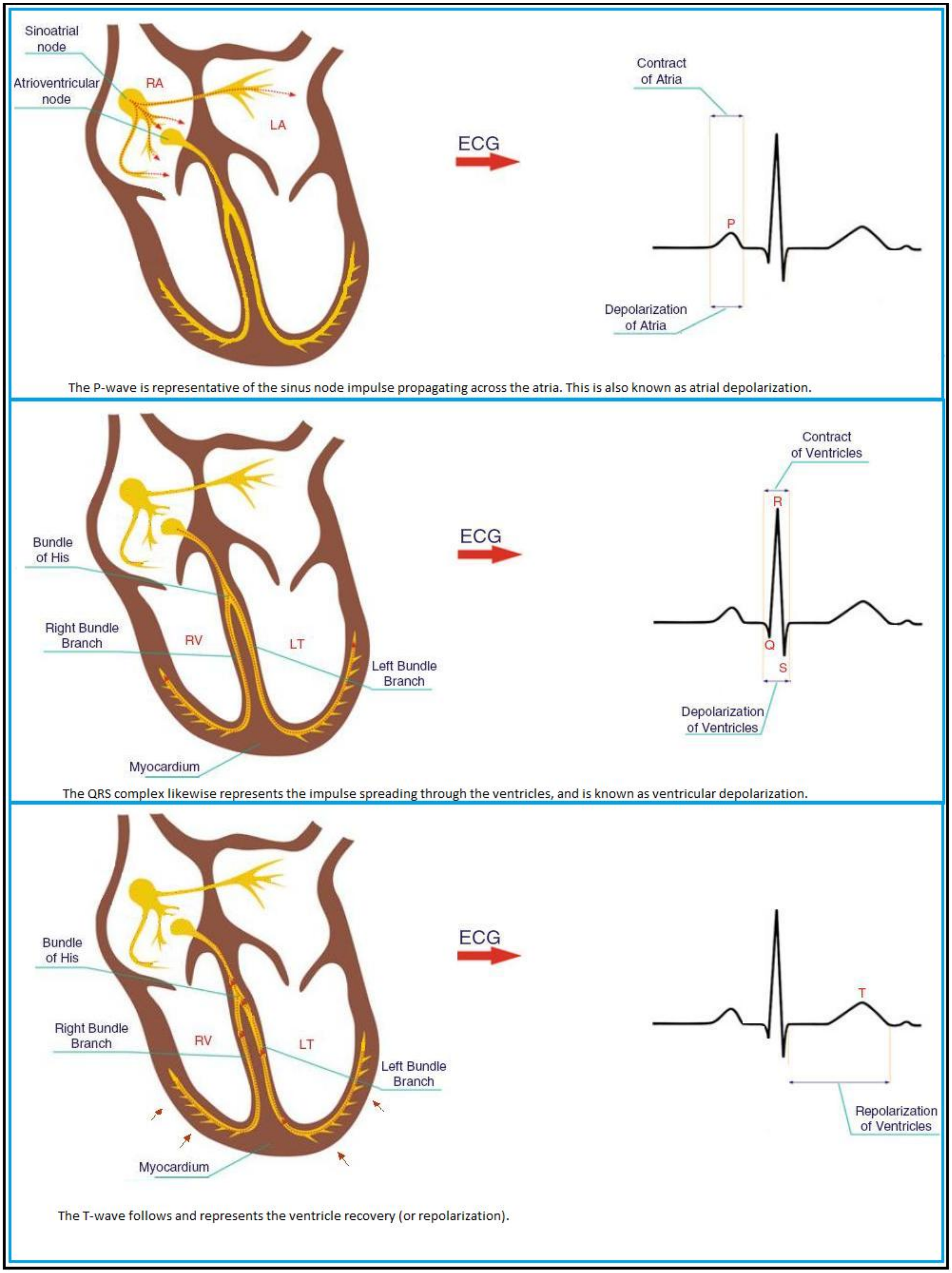


Figure 1 - Proper Electrode Placement

This placement is used to create the three dimensional cardiac axis. This leads to additional available information, which in turn leads to a more accurate diagnosis.

The figure found on the following page (Figure 2) depicts the ideal propagation of the electrical impulse responsible for controlling the heartbeat. Initially, the cells found in the sinus node exhibit spontaneous depolarization which propagates to the atria causing it to contract. This depolarization can be seen on an ECG as the P wave deflection. The subsequent depolarization and contraction of the ventricles results in the large deflection named the QRS complex. The slight pause between the two waves is due to the slow impulse propagation through the atrioventricular node. Finally, the ventricles repolarize and expand the heart which is seen as a T wave deflection on the ECG.



The P-wave is representative of the sinus node impulse propagating across the atria. This is also known as atrial depolarization.

The QRS complex likewise represents the impulse spreading through the ventricles, and is known as ventricular depolarization.

The T-wave follows and represents the ventricle recovery (or repolarization).

Figure 2 - Electrical Impulse Propagation of the Heart

2.2 The Normal ECG and its Characteristics:

The proper diagnosis of a heart irregularity requires that a normal "healthy" baseline is established. This poses a problem, as the normal shape of an ECG is dependent on the age and gender of the patient. Despite this issue there are common ranges for the waveform components of an ECG. The following figure (Figure 1) graphically displays the normal shape of an ECG, and includes the prominent component and interval labels:

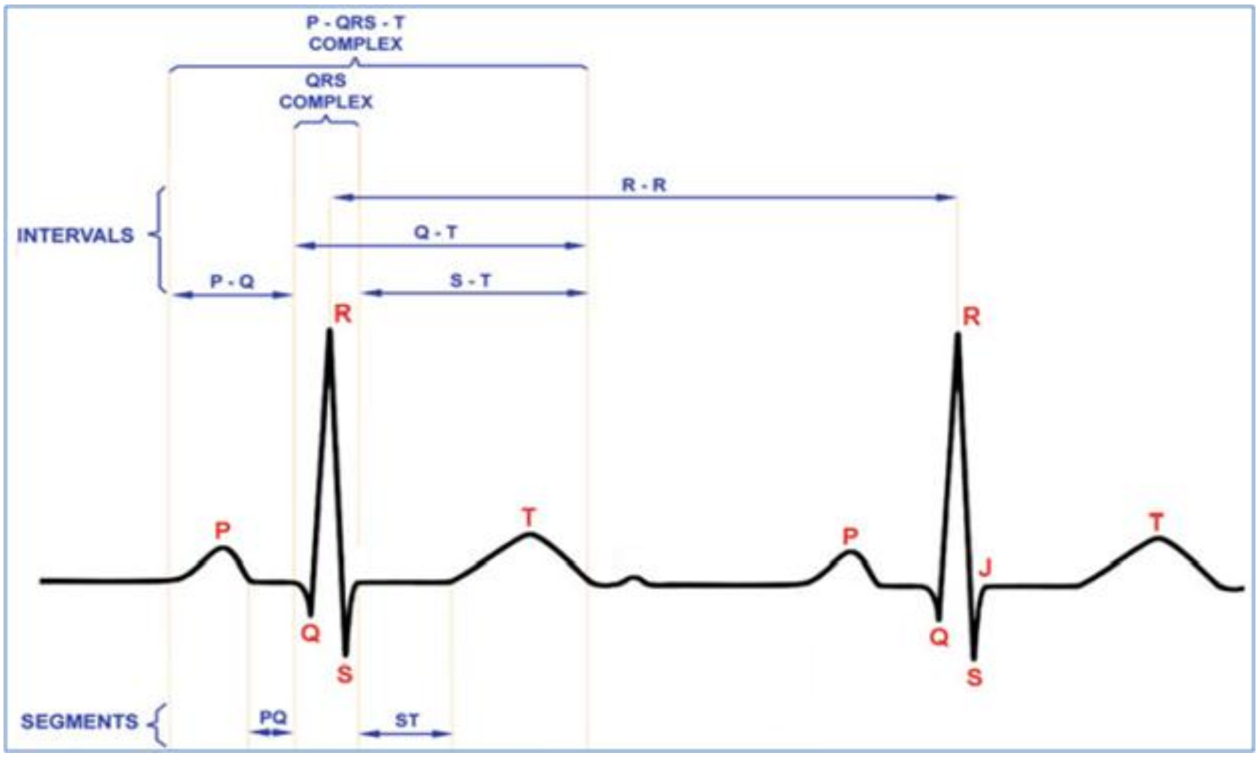


Figure 3 - Normal ECG with Labeled Intervals and Segments

The following table (Table 1) contains the nominal values of the components in Figure 1:

Table 1 –Variants of a Normal ECG

Name: Interval/Segment	Typical Duration (s)	Typical Peak Voltage Reading (V)
R-R	60 – 100 beats per minute	-
P-Wave	< 0.08	< 0.25 mV
P-Q	0.14	-
QRS Complex	< 0.12	< 1 mV
Q-T	0.37	-
T-Wave	0.16	0 <normal voltage < 0. 4
S-T	0.28	-

2.3 The Classification of Common ECG Irregularities:

Once a normal baseline has been set any deviations from the norm may be the result of a dangerous irregularity. Since its invention, the ECG has been utilized to characterize these irregularities. A correlation must be found between the ECG signal and a harmful physical condition. Once a correlation is established an initial diagnosis is possible.

There are many problems that can be diagnosed with an ECG, but the focus of this project was to narrow the scope to emergent and urgent conditions. Thus, our goal is to identify and handle the most commonly occurring heart problems. These problems can be divided into two categories:

- arrhythmia (abnormal heart rate)
- myocardial injury (heart attack)

Approximately 90-95% of all emergent cases that require immediate assistance can be evaluated by determining the heart rate and the S-T interval (elevated or depressed?). The most common irregularities and their key diagnostic features are listed in the following two tables:

Table 2 - Conditions requiring medical attention

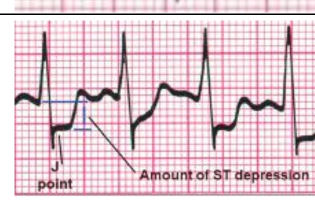
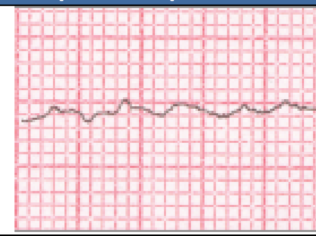
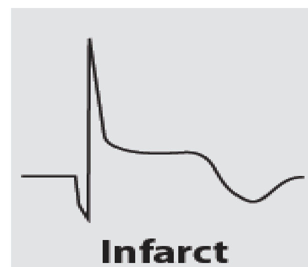
Condition	Description	Threshold/Key Features	Graphical Representation
Atrial Fibrillation	Irregular contraction of atrial, unable to generate enough energy for the heart to pump blood into ventricles - results in blood pressure drop.	No P-wave Chaotic baseline Irregular QRS rate R-R > 100 bpm	
Heart Block (1st Degree)	Conduction from the atria to the ventricle is blocked. Condition may deteriorate to potential cardiac arrest.	P-Q > 0.2 s	
Ischemia	Condition in which insufficient oxygen is reaching the heart muscle.	Inverted T-wave	

Table 3 - Conditions requiring immediate medical attention (i.e. ambulance)

Condition	Description	Threshold/Key Features	Graphical Representation
Ventricular Fibrillation	Life-threatening arrhythmia. This is caused by chaotic impulses, essentially stopping the heart.	No P-wave Chaotic baseline No QRS complex	
Myocardial Injury	A precursor to a heart attack, this injury indicates a tear or lesion.	QRS < 0.12 s S-T : 0.1- 0.6 mV	
Myocardial Infarction	Commonly known as a “heart attack”. Caused is caused by a loss of oxygen to the heart.	QRS < 0.12 s S-T: 0.1- 0. 4 mV	

The tables above were divided based on the severity and required response of the irregularity. Half the cases were deemed to require immediate medical attention, whereas the other three were not as pressing, but still required medical attention within approximately two days. An algorithm to capture these key factors must be created to differentiate the six cases from each other and the nominal values of a “normal” ECG. The framework for this solution is discussed next.

3.1 The Software Solution:

There are two main purposes to the software that must be written for this ECG device. The first part simply records the data and stores it. The second part will do dynamic analysis on the ECG wave to determine the patient’s current heart condition. If there is a serious heart irregularity then the software can react by either informing the patient they need to set up a doctor’s appointment soon, or by immediately calling an ambulance.

3.2 How to Receive and Analyze ECG Data:

To properly determine the patient's heart condition, the software must obtain values for the five main variables in the ECG graph: P, Q, R, S, and T. Each variable needs both a voltage value along with its associated time value. These can then be compared against each other in order to identify a variety of heart irregularities. We have created an algorithm that scans the current input value, seen in Appendix A, and compares it against previous input values to identify peaks and valleys in the graph. If the previous two values are greater than the current value by a certain threshold and the slope of the graph was previously positive, then it has hit a peak. If the previous two values are less than the current value by a certain threshold and the slope of the graph was previously negative, then it has hit a valley. This threshold value is in place to make sure miniscule fluctuations in the signal are not identified as peaks or valleys. The value would have to be determined through experimental testing. Five peaks and their associated time values are recorded in chronological order. These peaks will correspond to the P, Q, R, S, and T values that are needed for the dynamic analysis. While this is running, an interrupt service routine will be triggered every two milliseconds that will keep track of the time, update the current input value, keep track of the previous input values, and store the current value along with its time in to a data file.

After the necessary values have been recorded, a series of comparisons are made to try and identify certain heart irregularities. There are four examples given in the algorithm: fluttering, a heart attack, an ischemia, and a heart blockage. Fluttering is detected by detecting an insufficient R peak for a series of heart beats. This is indicated by an R peak that is less than two times the height of the P peak. A heart attack can be determined by observing that the S wave is above a certain value. For the purpose of writing the algorithm, a common value of 0.1 mV was used. To detect an ischemia, the algorithm must identify an inverted T wave. This can be done simply by seeing if the peak value recorded for the T wave is below the baseline. And finally, to detect a heart blockage, the time difference between the P and Q wave must be determined. If this is too large, in our example above 200 milliseconds, then it is indicative of a heart block. There are hundreds of different heart abnormalities that can be determined using the ECG graph data. For the purpose of this report, a few key examples were looked at to show how the algorithm will detect these events.

3.3 Program Adaptability:

The numbers used in the algorithm are common values, and they will not be sufficient to use for every person. The graph generated by the ECG will be substantially different from each person. What may look like normal and safe ECG data for a middle aged man could look like heart failure for an elderly woman. In order to make sure this program can work with any person, regardless of sex or age, it first must be calibrated for each unique individual. This can be done in two ways.

The first method would be to have each patient enter all their necessary personal information. This would include age, sex, height, weight, and known heart conditions. Then, a normal range of values for each variable and condition could be assigned to the individual. This method is quick and easy, allowing the patient to calibrate the device without requiring a doctor’s appointment. It could however run into problems if an individual has a unique heart pattern that might not fit well enough into the pre-set values.

The second method would be to experimentally determine each individual’s nominal values. This could be done by having the patient hooked up to the ECG and then perform a series of physical actions. The resulting ECG data could then be used to create the range of normal expected values for the various peaks and cases indicative of heart failure. However, if for some reason the heart is not giving a healthy signal during this calibration time, then the stored normal range of values could actually be indicating unhealthy heart conditions. It could also mean that a healthy heart beat could be interpreted as dangerous. In order to prevent this from happening, the individual must have the program calibrated while hooked up to an external ECG being monitored by a doctor. This would ensure that the ECG software is calibrated properly using healthy ECG data. This could more of an inconvenience for many patients, but the patient is likely going to need frequent doctor’s appointments if they have resorted to having constant ECG monitoring.

The first method would be more convenient for the patient as it would not require a doctor’s appointment to calibrate the ECG device. The second method is more inconvenient to calibrate, yet should yield more personalized and accurate values. This could result in a much smoother program with less false positives when identifying heart abnormalities. Experimentation must be done to determine if the first method will provide enough information to determine the nominal values.

3.4 Responding to the Determined Situation:

Once the situation has been determined, the program must then decide what action to take. There are four different responses depending on the severity of the current heart condition. If it has been determined that there is no emergency, then the program will just store the ECG data and continue to monitor the incoming ECG signal. If a heart condition has been detected that will need medical attention, but not immediately, then a notification can appear on the screen informing the patient that they must set up a doctor's appointment soon or visit the hospital. This response could also send a notification to their doctor informing them of detected heart problem. In the case of a potentially serious heart condition, a notification could pop up on the screen asking the patient if they require an ambulance. A simple button press on the screen could either dismiss the notification or immediately contact emergency services. A safeguard could be implemented in case the patient is not able to respond where the program will automatically contact emergency services if there is no response by a certain amount of time. If a major emergency has been identified, the program can contact the local emergency services automatically.

In recent years updates have been made to 911 call centers in certain regions which make it possible to send SMS (short message service) messages to the call center in cases of emergency. This allows us to create automated emergency SMS messages that can alert the appropriate authorities if a serious anomaly is detected when analysing the received ECG signal. Though not all regions have had these upgrades made yet so we will need a secondary method of alerting authorities, an example of this is British Columbia which still only accepts phone calls to 911. In the case an anomaly was detected the ECG software would go through the following procedure to make an SSM message. First, the software would verify that the anomaly detected was not caused by a hardware malfunction, like a temporary loss of signal or data corruption, by running through a self-diagnostic. It is important to complete this verification, because making multiple false emergency calls can lead to legal fines. If the software verifies that the anomaly is a genuine emergency it will begin the process of creating and sending an emergency SMS message. The SMS message would be created using a pre-programed template that would contain the generic information used by all possible messages and the specific information being filled in would depend on the type of anomaly detected. Some of the generic information included in the message would be that it is an automated message from an ECG device where medical assistance is required, as well as the patient's personal information like name, age, build, and medical history. The specific information that would be added to the template would be location of patient using the phones location data, and tell what type of problem was detected so the dispatcher can inform the responding paramedics. In regions where SMS messages still cannot be received by 911 dispatchers the ECG software could make an automated phone call instead and play a pre-recorded message stored on the smartphone which would consist of much of the same information sent in the SMS message.

4.1 The Limitations and Purpose of Using a Smartphone:

Since this is a smartphone application that will be running in the background at all times, it cannot be intrusive and resource hogging. The amount of analysis that can be done on an ECG signal is enormous and cannot be completed on a smartphone while still having it operate as a phone. This application is only designed to identify key serious heart conditions that require immediate medical attention. The data recorded from this application is intended to be received by the patient’s doctor and analyzed in depth using an ECG laboratory at a large medical facility. There the signal data can be looked at using dedicated hardware and incredibly complicated software. The results from this will be a complete report on the patient’s heart condition.

The unique feature of this report is that it will be able to show how the heart behaves leading up to various heart abnormalities. This data is crucial for research on heart failure as it currently is not being captured using traditional ECG devices. If a patient experiences some form of heart failure and is then observed by an ECG later in the hospital, the heart pattern leading up to the event is never seen. Using the portable constant ECG monitoring, this data can be captured and used by researchers.

In order to develop this application, a platform must be chosen. To do this, a number of factors must be looked at including how to create an app for the various marketplaces and distribution services, as well as the hardware behind each device. The chosen platform must be able to handle all the requirements by the application.

4.2 Mobile Device Platform Comparison:

When considering the functionality and implementation of our program as an app for either the iOS or Android market it is important to look at which services are needed to produce effective results and what kind of costs we could expect to incur.

To register for iOS development, one must pay $99 annually to keep their app on the market and Apple takes 30% of all sales revenue. The 30% fee on sales revenue will not affect our operations because the app will be offered free of charge with the purchase of our wireless ECG machine. Alternatively, the Android market has a one-time fee of $25 to publish an app on the market. This is vastly more beneficial than paying the $99 annual fee from Apple.

In order to acquire contrast between the two options of iOS vs. Android we will compare Apple's latest product, the iPhone 5, against a competitive high-end Android phone, the Samsung Galaxy S III. These devices can be seen in Figure 5. Firstly, the hardware will be compared to gain knowledge of what actually sets these devices apart other than their operating system. While in many respects these products are similar, there are a few integral characteristics that set them apart. Both have large touch-screens, loudspeakers, headphone jacks, Wi-Fi, cameras, etc. but these factors do not play any significant role for our purposes. What does matter are the performance related aspects of the devices such as RAM, processor speed, Bluetooth functionality, and SD card slot capabilities. The iPhone 5 and Galaxy S III both have 1GB of RAM which will be sufficient for our needs seeing as any consumer would be hard pressed to find a mobile phone device that has more. What sets the iPhone and Galaxy S III apart is the processors; the iPhone boasts a dual-core 1.2 GHz processor while the Galaxy S III performs with a quad-core 1.4 GHz processor. While it may not look like they differ by much the reality is the Galaxy S III will be considerably superior not only in speed but in multiple operations at one time; twice as many as the iPhone to be specific. This is very important for our app because we need it to run silently in the background while the user still operates their phone as normal without experiencing any noticeable lag or performance issues while multi-tasking. What makes the biggest impact on choosing a hardware platform for our app is the SD card slot. The iPhone has zero support for an SD card slot while the Galaxy S III supports up to one 64GB MicroSD card while still having plenty of internal storage on the phone itself (up to 64 GB). The capability of having an SD card inserted into the phone is crucial for our design because our method of data storage is to store the data on an external SD card in order to avoid using the majority of the phone's internal storage, which is needed for other apps and data. Without an SD card slot we would have to use up the phone's free internal storage, which varies from user to user, so it would be hard to tell how long our app could run before filling all free space.

Using an estimation that a single record of a sensor lead uses two 5-digit values to record a voltage and time value accordingly, a sensor being recorded every 2 milliseconds would produce approximately 20MB of data every hour. Extrapolating this observation for two probes producing 40MB of data every hour, a 64GB SD card would be filled up in 2 months. This is obviously more than enough storage space to perform our task but it is safe to say that even a 32GB SD card would last approximately a month before the user would have to pass the data to his or her doctor to be analyzed. If the patient is swapping SD cards, the program would keep recording data onto the phone's internal storage until a new card was inserted. At this point, the program would dump all the data that was temporarily stored on the phone’s internal memory back onto the SD card and resume recording as normal.

In the case of not having an SD card, the transfer of data would get considerably more difficult because of the further development required to support large data-dumps from phone to computer and furthermore how the client would get the data to the doctor without any discrepancies. It would be devastating to the analysis of the ECG data if it was incomplete or corrupt so using an SD card is the best option. Sending the data over the network to be stored at the doctor's office is not a viable option. Many people have limits on their monthly data plan and uploading 30GB of data per month would end up being far too expensive using any standard data plan. Another feature that could prove useful is the FM radio of the Galaxy S III. This is because ECGs are sensitive to electronic interference and so the FM radio antennae can be used to detect and correct for any signals in the area surrounding the user.

Secondly, on the software side of choosing a platform we need to consider if the operating system itself can supply the services our app will need for everyday operations. It is essential for our app to run while not being actively used by the user; this is so essential because of the need for run-time analysis of incoming data via Bluetooth while not demanding all of the phones resources. It would be unacceptable to have an app that only works while being actively viewed because an ECG requires an uninterrupted data flow in order to properly analyze the data.

Given that the only real requirement from the operating system is the ability to run background operation we use this factor to differentiate between the two different choices - iOS and Android. The Apple iPhone, using iOS, supports multi-tasking in a very limited specific manner by which the app can request to run in the background where it can run code but this time is limited and intended for apps to be able to finish functions before becoming suspended. This background operation is insufficient for our purposes unless the user can remember to view the app before the 10 minutes is up. The only other work-around to our problem is iOS supports an app to receive updates from an external device but seeing as we would be giving updates between 250-500 times per second this option becomes a major burden to the system. In the newer versions of iOS background multitasking is available in a limited way where the app requests it but may not receive from the system. Since there is no guarantee of permission this method is unreliable and not suggested. Also, the use of these services is intended for limited use. Using them on a constant basis would be abusive and demanding on the processing capabilities.

The Android software platform is much more compatible for our purposes as it is able to declare our app as a 'service'. The Android developer reference site defines a service as "an application component representing either an application's desire to perform a longer-running operation while not interacting with the user or to supply functionality for other applications to use". Such straightforward functioning as well as a full online reference for all commands, classes, and functions makes development in Android the more appealing option. This can be seen in Figure 4.

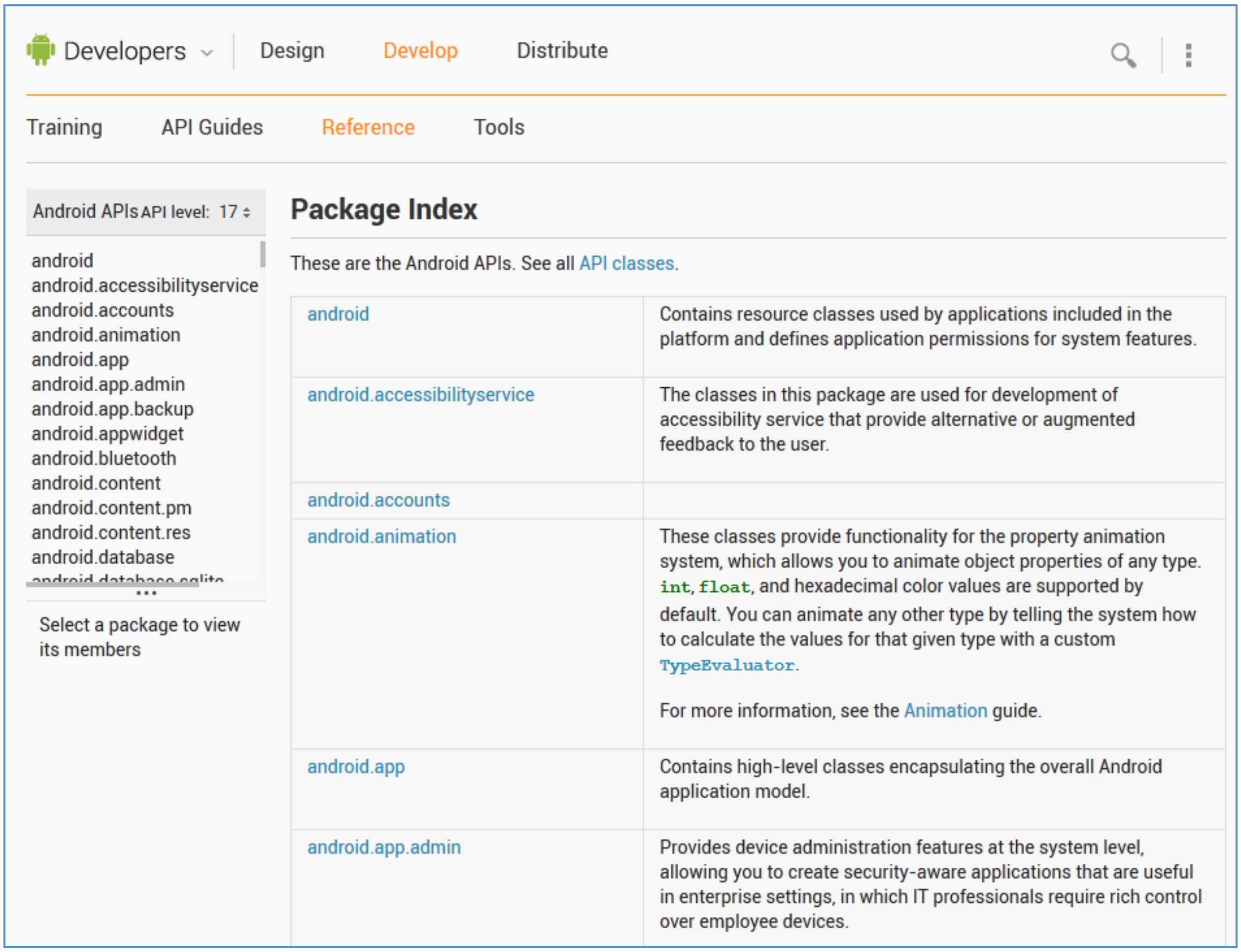


Figure 4 - Android Online Reference Library

With the simplicity of just defining our app as a 'service' and therefore having unlimited run-time on the platform we do not have to fight against the operating system for run-time and can focus more on making the app less processor-heavy. After looking extensively at the Apple product it is obvious that is has shortfalls that simply cannot work for our purposes and needs. Without an SD card slot or sufficient software support we feel that developing our app for iOS is not in our best interests. Looking forward towards Android, it can be seen that there are a larger selection of devices to choose from and a hardware/software platform that is best for our purposes. In summary android is the best choice when considering what phone to use for our wireless ECG machine.

Figure 5 - iPhone 5 and Samsung Galaxy SIII

5.0 Conclusion:

There is a gap in the ECG technology that currently doesn't allow for the data leading up to some form of heart failure to be captured. A portable ECG device that is constantly monitoring a patient's heart could not only save their life directly by calling for help in the case of heart failure, it can also save many lives indirectly by providing researchers with data outlining a heart's activity leading up to heart failure.

An algorithm outline has been developed that can read the signal from a portable ECG and perform two main tasks with it. The first is to store the data for in depth analysis in a laboratory later. The second task is to perform some basic dynamic live analysis on the incoming data to determine if the user is experiencing a medical emergency. If this is the case then the smartphone will take appropriate action depending on the severity of the emergency.

The two smartphone giants, iOS and Android, were taken into consideration for development of this app. Android provided many more freedoms regarding their marketplace, resource allocation on the device, and expandable storage options. Taking these factors into account, developing the ECG app for Android is the recommended choice.

6.0 References:

1. Gacek AC. (2012, October 30th). ECG signal processing, classification, and interpretation a comprehensive framework of computational intelligence.[On-line]. Available: http://ezproxy.library.uvic.ca/login?url=http://dx.doi.org/10.1007/978-0-85729-868-3

2. Rowlands, Angela and Sargent, Andrews. The ECG workbook. (2nd edition). (2012, October 30th). [On-line]. Available: http://ezproxy.library.uvic.ca/login?url=http://site.ebrary.com/lib/uvic/Top?id=10484323

3. Gertsch, Marc. The ECG manual : an evidence-based approach. (2012, October 30th). [On-line]. Available: http://ezproxy.library.uvic.ca/login?url=http://dx.doi.org/10.1007/978-1-84800-171-8

4. Kusumoto, Fred and Bernath, Pam. ECG interpretation for everyone: an on-the-spot guide . (2012, October 30th). [On-line]. Available: http://ezproxy.library.uvic.ca/login?url=http://onlinelibrary.wiley.com/book/10.1002/9781119962168

5. Apple inc. (2012). *iOS App Programming Guide* [Online]. Available: http://developer.apple.com/library/ios/#documentation/iphone/conceptual/iphoneosprogrammingguide/ManagingYourApplicationsFlow/ManagingYourApplicationsFlow.html#//apple_ref/doc/uid/TP40007072-CH4-SW20, Oct. 19, 2012 [Nov. 29, 2012].

6. GSMArena. (2012). *Apple iPhone 5* [Online]. Available: http://www.gsmarena.com/apple_iphone_5-4910.php, [Nov. 29, 2012].

7. GSMArena. (2012). *Samsung I9300 Galaxy S III* [Online]. Available: http://www.gsmarena.com/samsung_i9300_galaxy_s_iii-4238.php, [Nov. 29, 2012].

8. Android. (2012). *Android Developers Package Index* [Online]. Available: http://developer.android.com/reference/packages.html, Nov. 16, 2012 [Nov. 30, 2012].

Appendix A: Algorithm Code Outline

Below is an outline on how the code may look for the algorithm if it was implemented. It is intended to give a general idea on how the algorithm will operate and is not intended to be complete working software.

An interrupt service routine will be used to store the “input_current” variable along with its associated “time” value into a data file. It will also keep track and update the “time” variable. This service routine will be activated every two milliseconds.

Also, there must be some initialization software to make sure that the first peak recorded is indeed the P wave and the device hasn’t started recording mid beat. This could be done by determining which peak “peak5” is on the first initial reading and then waiting the appropriate amount of peaks to begin recording a full individual beat.

```
main(){

	double time = 0;

	double diff1 = 0;
	double diff2 = 0;
	double input_current = 0;
	double input_previous = 0;
	double input_previousprevious = 0;
	double peak = 0;

	double peak1 = 0;
	double peak2 = 0;
	double peak3 = 0;
	double peak4 = 0;
	double peak5 = 0;

	double peak1_t = 0;
	double peak2_t = 0;
	double peak3_t = 0;
	double peak4_t = 0;
	double peak5_t = 0;

	double heart_rate = 0;
	double peak3_t_previous = 0;
	double bpm = 0;

	double th = ?;

	//this is assuming we do that initialization and are starting by
	looking //for P first
	boolean slope_positive = true;

	boolean peak_set = false;
	boolean 1set = false;
	boolean 2set = false;
	boolean 3set = false;
```

```
	boolean 4set = false;
	boolean 5set = false;
	boolean done = false;



	//three levels of emergency:
	//1 = immediate ambulance call
	//2 = popup asking if you need an ambulance
	//3 = popup saying you should see your doctor soon
	//0 = no emergency
	int emergency = 0;

	int fluttering = 0;



	while(1){

		//captures the peaks
		while(!done){

			diff1 = input_current - input_previous;
			diff2 = input_previous - input_previousprevious;

			if(diff1<0 && diff2<0 && abs(diff1)>th && abs(diff2)>th &&
slope_positive==true){
				peak = input_previousprevious;
				slope_positive = false;
				peak_set = true;
			}

			if(diff1>0 && diff2>0 && abs(diff1)>th && abs(diff2)>th &&
slope_positive==false){
				peak = input_previousprevious;
				slope_positive = true;
				peak_set = true;
			}

			if(peak_set){
				if(1set == false){
					peak1 = peak;	//P
					peak1_t = time;
					1set = true;
				}else if(2set == false){
					peak2 = peak;	//Q
					peak2_t = time;
					2set = true;
				}else if(3set == false){
					peak3 = peak;	//R
					peak3_t = time;
					3set = true;
				}else if(4set == false){
					peak4 = peak;	//S
					peak4_t = time;
```

```
                    4set = true;
                }else if(5set == false){
                    peak5 = peak;     //T
                    peak5_t = time;
                    5set = true;
                }else done = true;
            }
            peak_set = false;
        }

        //difference between the time values of the current and previous
        R //peak
        heart_rate = peak3_t - peak3_t_previous;
        bpm =

        //check to see if heart has been "fluttering" for 5 beats
        if(peak3 < 2*peak1){
            fluttering++;
            if(fluttering >= 5) emergency = 1;
        } else fluttering = 0;

        //check for a heart attack
        if(peak4 >= 0.1){
            emergency = 1;
        }

        //check for ischemia (inverted T wave)
        if(peak5 < 0){
            emergency = 3;
        }

        //check for heart block
        if(peak2_t - peak1_t > 0.2){
            emergency = 3;
        }

        //This is where the program will either display a notification or
        //contact emergency services depending on the value of
        "emergency"

        //RESET ALL THE NECESSARY VARIABLES (emergency, done, etc..)
    }
}
```

Appendix B: iPhone 5 Specifications

Apple iPhone 5

GENERAL	2G Network	GSM 850 / 900 / 1800 / 1900 - GSM A1428
		CDMA 800 / 1900 / 2100 - CDMA A1429
	3G Network	HSDPA 850 / 900 / 1900 / 2100 - GSM A1428
		CDMA2000 1xEV-DO - CDMA A1429
	4G Network	LTE 700 MHz Class 17 / 1700 / 2100 - GSM A1428 or LTE 850 / 1800 / 2100 - GSM A1429
		LTE 700 / 850 / 1800 / 1900 / 2100 - CDMA A1429
	SIM	Nano-SIM
	Announced	2012, September
	Status	Available. Released 2012, September

BODY	Dimensions	123.8 x 58.6 x 7.6 mm (4.87 x 2.31 x 0.30 in)
	Weight	112 g (3.95 oz)

DISPLAY	Type	LED-backlit IPS TFT, capacitive touchscreen, 16M colors
	Size	640 x 1136 pixels, 4.0 inches (~326 ppi pixel density)
	Multitouch	Yes
	Protection	Corning Gorilla Glass, oleophobic coating

SOUND	Alert types	Vibration, proprietary ringtones
	Loudspeaker	Yes
	3.5mm jack	Yes

MEMORY	Card slot	No
	Internal	16/32/64 GB storage, 1 GB RAM

DATA	GPRS	Yes
	EDGE	Yes
	Speed	DC-HSDPA, 42 Mbps; HSDPA, 21 Mbps; HSUPA, 5.76 Mbps, LTE, 100 Mbps; EV-DO Rev. A, up to 3.1 Mbps
	WLAN	Wi-Fi 802.11 a/b/g/n, dual-band, Wi-Fi hotspot
	Bluetooth	Yes, v4.0 with A2DP
	USB	Yes, v2.0

CAMERA	Primary	8 MP, 3264x2448 pixels, autofocus, LED flash, check quality
	Features	Simultaneous HD video and image recording, touch focus, geo-tagging, face detection, panorama, HDR
	Video	Yes, 1080p@30fps, LED video light, video stabilization, geo-tagging, check quality
	Secondary	Yes, 1.2 MP, 720p@30fps, face detection, FaceTime over Wi-Fi or

		Cellular

FEATURES	OS	iOS 6
	Chipset	Apple A6
	CPU	Dual-core 1.2 GHz
	GPU	PowerVR SGX 543MP3 (triple-core graphics)
	Sensors	Accelerometer, gyro, proximity, compass
	Messaging	iMessage, SMS (threaded view), MMS, Email, Push Email
	Browser	HTML (Safari)
	Radio	No
	GPS	Yes, with A-GPS support and GLONASS
	Java	No
	Colors	Black/Slate, White/Silver
		- Active noise cancellation with dedicated mic - Siri natural language commands and dictation - iCloud cloud service - Twitter and Facebook integration - TV-out - Maps - iBooks PDF reader - Audio/video player/editor - Organizer - Document viewer - Image viewer/editor - Voice memo/dial/command - Predictive text input

BATTERY		Standard battery, Li-Po 1440 mAh (5.45 Wh)
	Stand-by	Up to 225 h (2G) / Up to 225 h (3G)
	Talk time	Up to 8 h (2G) / Up to 8 h (3G)
	Music play	Up to 40 h

MISC	SAR US	1.18 W/kg (head) 1.18 W/kg (body)
	SAR EU	0.95 W/kg (head) 0.90 W/kg (body)
	Price group	*

TESTS	Display	Contrast ratio: 1320:1 (nominal) / 3.997:1 (sunlight)
	Loudspeaker	Voice 66dB / Noise 66dB / Ring 67dB
	Audio quality	Noise -91.3dB / Crosstalk -76.5dB
	Camera	Photo / Video
	Battery life	Endurance rating 51h

Appendix C: Galaxy S III Specifications

Samsung I9300 Galaxy S III

GENERAL	2G Network	GSM 850 / 900 / 1800 / 1900
	3G Network	HSDPA 850 / 900 / 1900 / 2100
	4G Network	LTE (regional)
	SIM	Micro-SIM
	Announced	2012, May
	Status	Available. Released 2012, May

BODY	Dimensions	136.6 x 70.6 x 8.6 mm (5.38 x 2.78 x 0.34 in)
	Weight	133 g (4.69 oz)
		- Touch-sensitive controls

DISPLAY	Type	Super AMOLED capacitive touchscreen, 16M colors
	Size	720 x 1280 pixels, 4.8 inches (~306 ppi pixel density)
	Multitouch	Yes
	Protection	Corning Gorilla Glass 2
		- TouchWiz UI

SOUND	Alert types	Vibration; MP3, WAV ringtones
	Loudspeaker	Yes
	3.5mm jack	Yes

MEMORY	Card slot	microSD, up to 64 GB
	Internal	16/32/64 GB storage, 1 GB RAM

DATA	GPRS	Class 12 (4+1/3+2/2+3/1+4 slots), 32 - 48 kbps
	EDGE	Class 12
	Speed	HSDPA, 21 Mbps; HSUPA, 5.76 Mbps
	WLAN	Wi-Fi 802.11 a/b/g/n, DLNA, Wi-Fi Direct, Wi-Fi hotspot
	Bluetooth	Yes, v4.0 with A2DP, EDR
	NFC	Yes
	USB	Yes, microUSB v2.0 (MHL), USB On-the-go

CAMERA	Primary	8 MP, 3264x2448 pixels, autofocus, LED flash, check quality
	Features	Simultaneous HD video and image recording, geo-tagging, touch focus, face and smile detection, image stabilization
	Video	Yes, 1080p@30fps, check quality
	Secondary	Yes, 1.9 MP, 720p@30fps

FEATURES	OS	Android OS, v4.0.4 (Ice Cream Sandwich), upgradeable to 4.1 (Jelly Bean)
	Chipset	Exynos 4412 Quad
	CPU	Quad-core 1.4 GHz Cortex-A9
	GPU	Mali-400MP
	Sensors	Accelerometer, gyro, proximity, compass, barometer
	Messaging	SMS(threaded view), MMS, Email, Push Mail, IM, RSS
	Browser	HTML, Adobe Flash
	Radio	Stereo FM radio with RDS
	GPS	Yes, with A-GPS support and GLONASS
	Java	Yes, via Java MIDP emulator
	Colors	Pebble blue, Marble white, Amber brown, Garnet red, Sapphire black, Titanium grey, La Fleur
		- S-Voice natural language commands and dictation - Smart Stay eye tracking - Dropbox (50 GB storage) - Active noise cancellation with dedicated mic - TV-out (via MHL A/V link) - SNS integration - MP4/DivX/XviD/WMV/H.264/H.263 player - MP3/WAV/eAAC+/AC3/FLAC player - Organizer - Image/video editor - Document viewer (Word, Excel, PowerPoint, PDF) - Google Search, Maps, Gmail, YouTube, Calendar, Google Talk, Picasa integration - Voice memo/dial/commands - Predictive text input (Swype)

BATTERY		Standard battery, Li-Ion 2100 mAh
	Stand-by	Up to 590 h (2G) / Up to 790 h (3G)
	Talk time	Up to 21 h 40 min (2G) / Up to 11 h 40 min (3G)

MISC	SAR US	0.55 W/kg (head) 1.49 W/kg (body)
	SAR EU	0.21 W/kg (head)
	Price group	*

TESTS	Display	Contrast ratio: Infinite (nominal) / 3.419:1 (sunlight)
	Loudspeaker	Voice 75dB / Noise 66dB / Ring 75dB
	Audio quality	Noise -90.3dB / Crosstalk -92.6dB
	Camera	Photo / Video
	Battery life	Endurance rating 50h

Log book grade (25%): _______________
Report grade (75%): _______________
Total Grade (100%): ________________

Supervisor’s Comments:

____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____

_________________________ _______________________ ______________________
Supervisor’s name (Print) Signature Date

Notes for the supervisor:

1. Please return the marked hard copy to Prof. Tao Lu by Monday, December 10.
2. Attached additional pages for comments if necessary.