DEVELOPING MOBILE, ELECTRONIC AND POSITIONING APPLICATIONS FOR EMERGENCY SITUATIONS INSIDE THE HOSPITAL

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Biotechnology

by
Özge Sevin KESKİN

July 2018
İZMİR

We approve the thesis of **Özge Sevin Keskin**

Examining Committee Members:

Asst. Prof. Dr. Engin KARABUDAK
Department of Chemistry, İzmir Institute of Technology

Asst. Prof. Dr. Nur Başak SÜRMELİ
Program of Biotechnology and Bioengineering, İzmir Institute of Technology

Assoc. Prof. Dr. Engin ÖZÇİVİCİ
Program of Biotechnology and Bioengineering, İzmir Institute of Technology

Prof. Dr. Gökhan AKBULUT
Department of General Surgery, Tepecik Education and Research Hospital

Assoc.Prof.Dr. Şeyda ÖRS KAYA
Department of Thoracic Surgery, Dr. Suat Seren Chest Disease and Surgery Training Hospital

9 July 2018

Asst.Prof.Dr.Engin KARABUDAK
Supervisor,Department of Chemistry
Izmir Institute of Technology

Asst.Prof.Dr.Nur Başak SÜRMELİ
Co-Supervisor,Program of Biotechnology and Bioengineering
Izmir Institute of Technology

Assoc.Prof.Dr.Engin ÖZÇİVİCİ
Head of the Department of
Biotechnology and Bioengineering

Prof.Dr.Aysun SOFUOĞLU
Dean of the Graduate School of
Engineering and Sciences

ACKNOWLEDGEMENT

There are a lot of person without whom it would not have been possible to complete this master thesis and to whom I am grateful.

To my supervisor, Asst. Prof. Dr. Engin KARABUDAK, for his full support, expert guidance, understanding and encouragement over the five semesters. He has been always an open source of guidance for my Master research. I have been motivated by his warmth and valuable experience. I also appreciate his patience and flexibility in allowing me to pursue research and experiments I found interesting.

Besides my advisor, I would like to thank to rest of my thesis committee: Prof. Dr. Gökhan AKBULUT, Assoc. Prof. Dr. Şeyda ÖRS KAYA, Assoc. Prof. Dr. Engin ÖZÇİVİCİ, Asst. Prof. Dr. Nur Başak Sürmeli.

To members of KARABUDAK Research Group, Emre Yusuf GÖL, Ahmet AYTEKİN, İpek HARMANLI, Mert KOÇ, Uğur SOĞUKKUYU and Mehmet KIVANÇ, and for their help and motivation. They always support me to make right decisions when I feel desperate and they are always there for me when I need them.

I would thank to dear doctors; Dr. Zeynep SOFUOĞLU and her husband Dr. Turhan SOFUOĞLU and the whole emergency staffs of Tepecik Education and Research Hospital. They always suggest new ideas so that I improved my perspective in Biomedical Research.

I thank my parents, Meral and Mustafa Zeki KESKİN; my peerless sister, Ezgi Gül KESKİN; without the love and support of my family I could not have completed this thesis. Without them, I would not be where I am today☺ Thanks to my lovely family, I love you!

I would thank to my other half, Cengiz GÜLMEZ; who always believed and encouraged me to complete this project.

I am grateful for the friendship of Gizem GÜLEÇ, İlhan KESKİN, Kaan ECE, Alper İNAN, Fetiye Esin YAKIN, Mehmet Onur CİRİT, Yasemin BİLGİ, Elif GÜREL, Aslı ADAK, Müge BİLGEN who always inspired me with their positive attitudes.

ABSTRACT

DEVELOPING MOBILE, ELECTRONIC AND POSITIONING APPLICATIONS FOR EMERGENCY SITUATIONS INSIDE THE HOSPITAL

The aim of this project is to prevent mortality by providing emergency response on time. The communication inside the hospital is provided via developing application for Android and Ios operating system. The content of application is suitable for everyone's use. The emergency call button makes it easy to call, and if the button is disabled, the heart rate rhythm starts for emergency intervention. This application is associated with bluetooth low energy systems in terms of help to locate the patient. Beacon systems provide data transfer through bluetooth by this means location is detected without internet. The project will be tested primarily at the laboratory following these steps. If it succeeds statistically, the project will be tried at the Tepecik Education and Research Hospital. The success of the project will be reviewed depends on the scenarios in the hospital.

ÖZET

HASTANE İÇİNDE ACİL DURUMLAR İÇİN MOBİL, ELEKTRONİK VE KONUM BELİRLEYİCİ UYGULAMA GELİŞTİRME

Projenin amacı hastane içindeki herhangi bir kişi için acil müdahaleyi zamanında sağlayarak hayati kaybı önlemektir. Android işletim sistemi ve Ios işletim sistemi için geliştirilen uygulamalar aracığıyla hastane içi iletişim sağlanmaktadır. Bu uygulamanın içeriği ise herkesin kullanımına uygundur. Acil arama butonu sayesinde kolaylıkla arama yapılabilmekte ve buton devre dışı kaldığında ise acil müdahaleye uygun kalp atım ritmi başlamaktadır. Bu uygulama hastanın yerini belirlemekte yardımcı olması açısından beacon sistemleriyle bağdaştırılmıştır. Beacon sistemleri bluetooth üzerinden veri aktarımı sağlar bu sayede internete gerek olmaksızın kişinin lokasyonu belirlenir. Projenin bu adımlarından sonra öncelikle laboratuvar ortamında testleri yapılır. İstatistiksel olarak başarılı olan bu proje hastanelerde denenir. Hastanede yapılan senaryolar üzerinden projenin başarısı gözden geçilir.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

Cardiopulmonary resuscitation and respiratory arrest is defined as the privileged situations. Resuscitation is the name of inspiratory and cardiac activities studies (Çete 2000). With timely and effective intervention in cardiopulmonary resuscitation (CPR), it is aimed to save both the patient's life and provide a functional quality of life while protecting the patient's previous health condition. For this purpose; Blue code teams, triggering and monitoring systems are constituted. Blue code teams have studied multidisciplinary to increase the effectiveness of resuscitation and give response to the emergency calls in time. Triggering and monitoring systems are known as “outreach service” in England, “rapid response” in USA, but basic logic and and the purpose of establishment are the same. Pre-determined teams with a call announcement system, known as "Blue Code" all over the world, aimed to respond to medical emergencies both quickly and organically.

CPR regularly updates its algorithm and training of health units leads to a better point every day. The training of people other than health personnel with the courses organized and the withdrawal of the attention of the society to this point with publications and trainings increased the visibility of the CPR ". Currently, only 20% of cardiopulmonary arrest cases in developed countries are performed by people with CPR training(Pearn 1999). In addition, the survival rate in clinics with more than five cardiac arrest per year was found to be higher when compared to clinics with fewer cardiac arrest (Hou et al. 2007).

It is important for the success of resuscitation that the age of the patient, accompanying diseases, physical conditions where arrest is seen, cardiac rhythm, rescuer's skill and awareness at the time of arrest (Eisenberg and Psaty 2009). All of this creates the basic dynamics of the blue code implementation. In general, the purpose of the Blue Code application is to interfere with the correct code implementation. On this basis the organization will reach its target with a higher average. We also work for recovering the current system to save life and improve the current system to help hospital staff with decreasing the interference time.

1.1 General Informations

1.1.1 Historical Process

When we look at not so distant histories, it is seen that the CPR efforts are started by conscious and instinctive foresight and experience. The cases of drowning in the water during the 1700s and 1800s are among the common causes of death due to the increase in maritime trade. To remove the water entering the lungs of the strangulated people, people were hanged from their legs, pressed upside down on their chest cages or put on a horse's back and ridden. This resuscitation effort, which is quite primitive and empowered by experience, contained some basic principles (Çete 2000).

In 1740, the Paris Academy of Sciences recommended oral mouth breathing resuscitation for drowning events, including oral smoking and flogging, which began to gain a more scientific structure in the following years. Again in 1767, the Dutch Union was established to revitalize the strangulated people and the British Royal Humanitarian Association was established in 1774 (Cooper, Cooper, and Cooper 2006). Marshall Hall, who lived during these periods, began to make changes in 1856 on the point of view of the world of science and the revival of society. He pointed out that the revival effort had to start at the scene, the unnecessary time and loss of time lost due to transfer (Çete 2000).

During the following years, many changes were made in both CPR and general medical knowledge. In 1958, Peter Safar brought oral breathing to the literature (Çete 2000). Kowenhoven, June and Knickborker redefined closed breast massage in 1960 (Çete 2000). The introduction of breast pressure and respiratory resuscitation at the Maryland Medical Association meeting in 1960 (Field et al. 2010, Eisenberg et al. 1980) and the definition of direct current monophasic waveform defibrillation in 1962 constituted the basic building blocks of the modern CPR (Lown et al. 1962).

1.1.2 Life-Saving Chain

Cardiac arrest is the cessation of mechanical activity of the heart.Non-palpable pulse can be seen in cases of agonal respiration (apnea) or unresponsiveness (Cummins et al.1997)

Cardiogenic shock for patients in hospital, such as advanced hypotension respiratory and circulatory disturbances cannot be assessed as cardiac arrest but at the same time point to

respiratory and circulatory disturbances. All the practices we do to turn the sudden cardiac arrest cases back to life are called the life-saving chain (Nolan et al. 2010). Benefits of using Blue Code systems in hospitals;

- Preliminary identification of emergencies in the hospital,
- Increased chances of success for both trained and experienced teams in specific situations, such as cardiopulmonary arrest, due to pre-selection and regular training of the teams,
- to prevent panic that may occur between patients' relatives as well as patient interventions in emergency situations and to provide a work environment.

Rings two and three show CPR and early defibrillation steps. Early onset CPR and defibrillation survival can increase two to three times (Valenzuela et al. 1997, Waalewijn, Tijssen, and Koster 2001). Each minute of delay in defibrillation can reduce the survival rate by 10-12% (Waalewijn et al. 2001). The fourth ring refers to the first aid process and urgent orientation of out-of-hospital arrests to the nearest appropriate center. It is the most critical need today for this step which is able to take long time and qualified personnel and adequate equipment. In the last period, post-resuscitation care is observed. In this section, it is aimed to bring all organ functions together with brain and cardiac function to the nearest performance to pre-arrest functions (Cummins et al.1997).

Time is as valuable as gold in all this chain system, which begins with the determination of the critical case (Hazinski et al. 2010).

1.1.3 Golden Hour

It defines a crucial time frame in which any intervention immediately following any acute illness or life-threatening injury has a marked effect on medical outcomes (Wu et al. 2012). In the study conducted between May 1991 and March 1998, which deals with the intervention of the Scotland Ambulance Service to arrest cases out of the hospital, the effect of shortening the intervention period on the discharge rates was investigated (McGrath 1987).10.554 of them were included in the 13.822 unsolved arrest cases. The survival rate was 6% in the patient group with a 15-minute intervention time, and when the intervention time was reduced to 8 minutes, the survival rate increased to 8% at

discharge and to 10% at discharge when the intervention was reduced to 5 minutes (McGrath 1987). Another factor that is precious in intervening in time is the basic life support shaped by experiences and works.

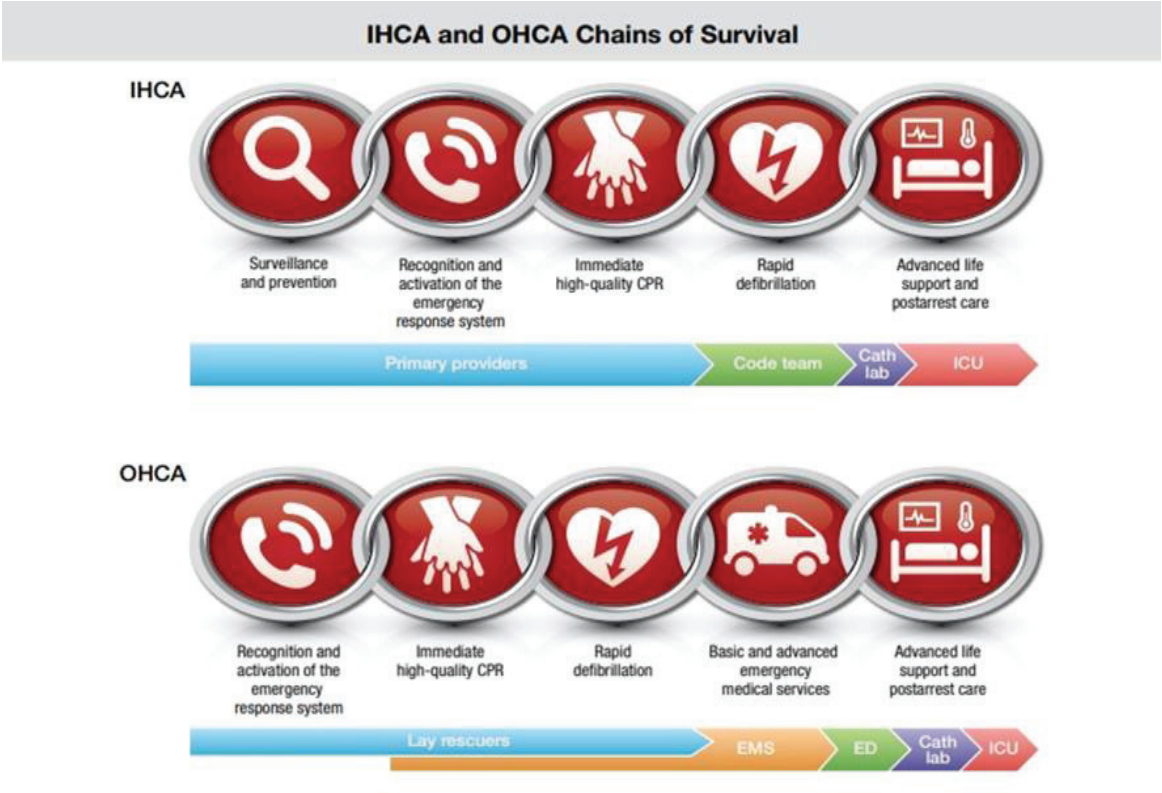


Figure 1.1. AHA survival of chain
(Source: Hazinski et al. 2010)

1.1.2 Basic Life Support

The International Liason Committee on Resuscitation was established in 1993 with the aim of making joint decisions on how to monitor, identify and treat scientific issues and international information related to CPR. Members of the committee are listed in the following list:

1. American Heart Association, AHA
2. European Resuscitation Council, ERC
3. Heart and Stroke Foundation of Canada
4. Resuscitation Council of Southern Africa
5. Inter American Heart Foundation
6. Resuscitation Council of Asia (Weil and Fries 2005)

The information contained in the released guidelines has been a recommendation for healthcare professionals. Certainly not. All practices and recommendations have been specified for this purpose according to the level of evidence and recommendation.

Level of evidence (Level of evidence description)

A: Randomized clinical trials or multiple clinical trials with significant treatment effects meta-analyzes of studies.

B: Randomized clinical trials with low or moderate treatment effects.

A: Prospective, controlled, non-randomized, cohort studies.

D: Important, non-randomized, cohort study or case-control studies.

E: Case series; compiled case series of patients without control group.

F: Studies with animal or mechanical models.

G: Assumptions based on other reasons, resulting in analysis of the results obtained or estimates.

H: Logical estimates (common approaches); without accepting blood-based protocols common daily practices classifications (Weil and Fries 2005).

Suggestion classification

Class I Benefits >>> Risk: Includes interventions, treatment or diagnostic tests, evaluations that are strictly required to be performed or performed.

Class IIa Benefit >> Risk: Includes intervention, treatment, or diagnostic testing, assessments that are recommended for implementation.

Class IIb Use ≥ Risk: Includes intervention, treatment or diagnostic tests, evaluations that may be available when practiced.

Class III Risk ≥ Benefit: Includes intervention, treatment or diagnostic tests, evaluations that should or should not be performed strictly. When applied, it is not beneficial, and it can be harm.

Indefinite class: Studies of relevance include evaluations that have just begun, studies are ongoing, are not recommended until more studies are completed (e.g. not implemented or not recommended) (Weil and Fries 2005).

1.1.5 Cardiac Arrest Algorithm in Hospital

Significant changes for 2010 in advanced cardiovascular life support (ACLS) are noted below."Quantitative waveform capnography" is recommended in terms of correct placement and follow-up of the tube. The traditional cardiac arrest algorithm has been

simplified to create an alternative model to emphasize the importance of high quality CPR.

Atropine pulse electrical activity (PEA) / asystole management is no longer a routine practice recommendation. In the case of unstable and symptomatic bradycardia, the application of chronotropic drug infusion as an alternative to pace application has been included in the proposals. In the case of regular monomorphic wide complex tachycardia, "adenosine" is suggested to be reliable and effective in terms of diagnosis as well as treatment. Furthermore, the definition of the large complex tachycardia has been changed and redefined in pediatric cases over the old state of 0.08 sec, the new state of which is over 0.09 s.

In pediatric case recommendations, it is reasonable to have a first defibrillation dose of 2-4 J / kg with a monophasic or biphasic waveform; For ease of teaching, 2 J / kg can be used in practice (same as 2005). At the second and subsequent doses it can be increased to 4 J / kg. Doses exceeding 4 J / kg can safely be used without exceeding the adult dose of 10 J / kg, especially if biphasic defibrillators are used.

1.1.6 Defibrillation

In the studies performed, the rhythm encountered during the first monitoring of cardiac arrest cases was determined as 25% ventricular fibrillation (VF) or ventricular tachycardia (VT) (Bradley et al. 2010, Iwami et al. 2009). When cardiac arrest is detected, CPR should be started immediately, and rhythm should be assessed as soon as the defibrillator is reached. When resuscitation is performed under the guidance of the advanced life support algorithm, it is necessary to pay attention to the time between chest compression and defibrillation. This period should be kept as short as possible, even short delays affect the success of the defibrillation in the worst way (Edelson et al. 2006). After defibrillation, CPR should be continued at a rate of 30: 2 without attempting to perform pulse or rhythm control. This is especially due to the time required for circulation to achieve adequate perfusion, even after successful defibrillation (Bradley et al. 2010). If the rhythm does not turn, the time losing when trying to get a pulse increases myocardial damage (van Alem, Sanou, and Koster 2003) In the VF, which lasts for a few minutes, the sources, oxygen, and metabolic substrates that myocardium has are rapidly consumed. Applying chest compression allows the oxygen and the substrates to reach the myocardium again, increasing the chances of success of resuscitation. A second

shock should be attempted if the rhythm is still VF / VT after an average of 2 minutes following CPR. The shock level to be selected for biphasic defibrillators during shock application is 150-360 J. The second and third shock tests are repeated as necessary with CPR and rhythm controls continuing in the same manner. If the appropriate vascular route was opened during this period, 1 mg of adrenaline and 300 mg of amiodarone should be administered via the vein after the third shock. Despite this third defibrillation, myocardial blood flow increases in the unturned patient, increasing the likelihood of success in the next defibrillation. If rhythm is detected as asystolic or pulseless electrical activity during resuscitation, the algorithm is translated into a non-shocked rhythm algorithm (Eftestøl et al. 2004).

1.1.7 In-Hospital Cardiac Arrest Prevention

As in all areas where human life continued, there are various risks at health care services. These risks are pre-determined and realized what to do and how to manage risk is important. A significant proportion of intra-hospital cardiac arrests are predictable. When we looked at the cases taken to the intensive care units from the services, it was seen that the mortality was higher than the patient population which was intensively examined from the operating or the operating room. The same height was also seen in APACHE 2 scores. However, the intensive care mortalities of patients with longer hospitalization times have also increased (Hillman et al. 2002). Physiologic abnormalities are important, especially in the last twenty-four hours changes are of extra importance in terms of prevention of cardiac arrest. Clinical findings, such as 80% of the cases, began to deteriorate hours before the arrest developed. It is especially seen in cases of cardiac arrest in which the hypoxia and hypotension are avoided by the healthcare personnel without being treated slowly or progressively or without appropriate treatment. Attention physiological abnormalities are frequently hypotension, tachycardia, decreased urination, tachypnea and mental status changes (Smith and Wood 1998). It has been observed that the survival rates of patients with chronic diseases are lower in cases of in-hospital cardiac arrest. Closer follow-up of the patient group that was mistakenly reported as cardiac arrest will affect survival positively because it has been determined that 1/3 of this patient group died during hospitalization. In addition, arrests with a rotatable rhythm and cases with rapid defibrillation have the best survival rates (Buist et al. 2002, Bellomo et al. 2003).

Regardless of the underlying cause, the clinical manifestations are often similar in critical illnesses. Because these diseases eventually manifest themselves in cardiac, respiratory and neurological deficits. Abnormal physiological findings are frequently encountered in the services, but careful attention is not given to those who have acute illnesses or when the physiological observation measurements are not made at the desired frequency in the case of acute illnesses. Many hospitals use early warning scores (EUS) or call criteria to increase the percentage of early detection of critical illness. Previously identified parameters before arrest in these findings are also used in prognosis after arrest. Scores such as post-resuscitation prognosis score, prearrest morbidity score (PAM) and modified PAM score are examples. There are studies showing that scoring systems shorten cardiac arrest, intensive care hospital stay and mortality (Buist et al. 2002, Bellomo et al. 2003). The goal of the system is to identify risks in time, to conduct surveillance, and to act quickly if necessary. Identifying risks the patient should begin to sleep in the hospital (Goldhill et al. 2004).

1.1.8 Medical Emergency Code Systems

Hospital medical emergency code systems; procedures outside of the hospital or in order to respond to the responsible team at the earliest possible time in cases where the persons who will be in the request of medical assistance. Several codes are occurred to use inside the hospitals which are named with colors. Color codes are determined to emergency situations for;

-Inform the hospital workers

-Opportunity to communicate during risky situation

-Give short and clear message

-Gain time to true interference

-Prevent the panic

-Provide security for patient and hospital staff members

-Enable to ready for emergency situations

The health institutions and organizations give courses to hospital staff members for acceptance of these color codes. In Turkey, code blue is brought in 2008, code pink is brought in 2009, code white is started in 2011 and code red is actualized in 2015 as part of health quality studies (Demirci 2016).

1.1.9 Code Pink

It is a system that sends and records all event information after the relevant staff in the hospital in case of baby abduction. The Code Pink staff is in charge during duty hours and the duty officer and security personnel and duty technician are on duty outside duty hours. A pink code alarm is given by the service personnel to the telephone number 3333 for the Pink Code, or a Pink Code alarm is given by reading the card for the emergency call and patient follow-up button if the event has occurred in the clinics. As soon as the Pink Code alarm is heard, all hospital doors are closed for entry and exit. The code management team is notified, the investigations are started, the measures are taken, and the doors are not opened until the event is resolved. Later on, the police department will be contacted via line 155. Subsequently, the notification form is filled in responsibly and delivered to the quality management unit. The practice for Pink Code application is done twice a year (Demirci 2016).

1.1.10 Code White

In case of a possible attack or harassment in and around the office hours, the number 1111 assigned to the White Code is searched in the hospital's central office. If the event occurs in the clinic, the card is read, and the White Code alarm is given. The security officer informs the emergency management physician and White Code management team by telephone. White code management team during working hours, security team, administrative physician and duty officer also come to the scene when outside working hours. The hospital police are notified, the necessary judicial process is initiated by the hospital police, the on-duty deputy director interviews the police headquarters and other judicial authorities and conducts the necessary judicial appeal work and transactions on behalf of the hospital. The scene provides backup of the camera records and handles it to law enforcement officials. The White Code event notification form is completed by the team and forwarded to the Employee Rights and Security Units. The unit shall provide legal assistance and psychological support to the employee and his / her family, if necessary, by making the necessary declarations.

1.1.11 Code Blue

Basic life functions are an early warning system that enables the immediate intervention of the severely threatened or stopped individuals, that is, the implementation of CPR, which enables rapid notification of previously prepared crews and helps the team to arrive at the scene quickly. The Code Blue caller is responsible for providing immediate basic life support wherever he or she is present, until the patient has advanced to the intervention. Code Blue is applied to all units except emergency service. When Code Blue is required, 2222 telephone from the nearest extension telephone is searched or emergency card and patient follow button in hospital clinical and intensive care units are read and the Code Blue team is brought to the scene. When the incoming team arrives, the emergency call is terminated by reading the card to the emergency call and patient follow-up button. In the case of Code Blue, the shortest time (maximum 3 minutes) is reached with the emergency intervention bag (Demirci 2016). Blue Code applications are followed by intensive care and other treatment modalities, resulting in varying survival rates (Pepe, Abramson, and Brown 1994). In the compilation of 44 studies, the survival rate was 3-27%. This change in survival rates is due to differences in study inclusion criteria and study outcomes. In the United States, 750,000 cases of cardiac arrest are registered annually(Sener and Yaylaci 2010). 1-4% of these patients are hospital workers and visitors(Eisenberg and Mengert 2001). The teams that intervene in the cardiac arrest cases in the in-hospital and out-of-hospital areas are composed of physicians, nurses and paramedics from different branches and specialization levels(Keys and Selzler 2011, Henderson and Ballesteros 2001). The leadership of this team is influenced positively by the staff of advanced life support trained. Another very important factor for the team and the hospital is keeping records properly. In many institutions, a member of the Blue Code Rapid Response team is responsible for filling out the intervention assessment form, where the quality of the resuscitation process is assessed. With these forms, specific problems such as delayed access to the defibrillator, aspirator and other devices, lack of intubation and resuscitation equipment or malfunctions are recorded in addition to the effectiveness, duration, drug doses of the basic life support application. KPR training is given by the Anesthesiologist in every semester (at least 2 times a year) and in training courses for new staff. At least two times each period for the blue code application is performed and the record of how long the exercise is reached is recorded. The Code Blue notices to the quality unit are filed and

archived, and staff members who exceed the time of reaching the scene (max. 3 min) and who cannot attend as a team member are determined. Corrective action is initiated if necessary. These statistics are also reviewed by the patient safety committee and the Code Blue team, and the implementation of the Code Blue is followed by taking healing decisions (Demirci 2016).

A problem is recognized inside hospitals that the person, who needs immediate treatment, cannot reach emergency team easily or the person, who have recognized the urgent situations, cannot juridically call the urgent call number in the hospitals so that mortality rate inside the hospitals are bigger than outside the hospitals. To prevent mortality rate, a system was developed to interfere emergency situations when someone has life-threatening issues. Some systems were developed based on triangulation, GPRS but these systems have a problem with defining indoor location.

There is a current system which includes call center, emergency team and a button which provides easily communication between people inside the hospital to prevent mortality inside the hospitals.

1.2 Current Situation

In medicine literature, code system is known as Code Blue. The system is still functional to decrease of mortality when someone has a life-threatening issues. For example; heart attack, epileptic attack, cerebral hemorrhage etc. In Turkey, the reason of loss of life is that people fill the hospitals so emergency team can not interfere the person on time so that the situation concludes with death. During emergency; someone calls the emergency team member.

1.3 Problems of current situation

The current system has some deficiency, so it is realized by emergency team. These deficiencies are loss of time during interference, undetected location of person who needs urgent interference, wrong cardiac massage to the patient.

For example, you have had a heart attack in toilet and someone recognized you then he wants to call emergency team. The procedure will be these following steps ;

-Firstly, he must find a payphone

-He will be dial the 2222 which is the code blue number

-Call center answered the phone call. (If call center employee is there during call)

-Employee try to learn your location and direct code blue team

-Code blue team interfere the patient.

This process must be taken maximum three minutes, but our hospitals are crowded so that code blue difficultly find the location.

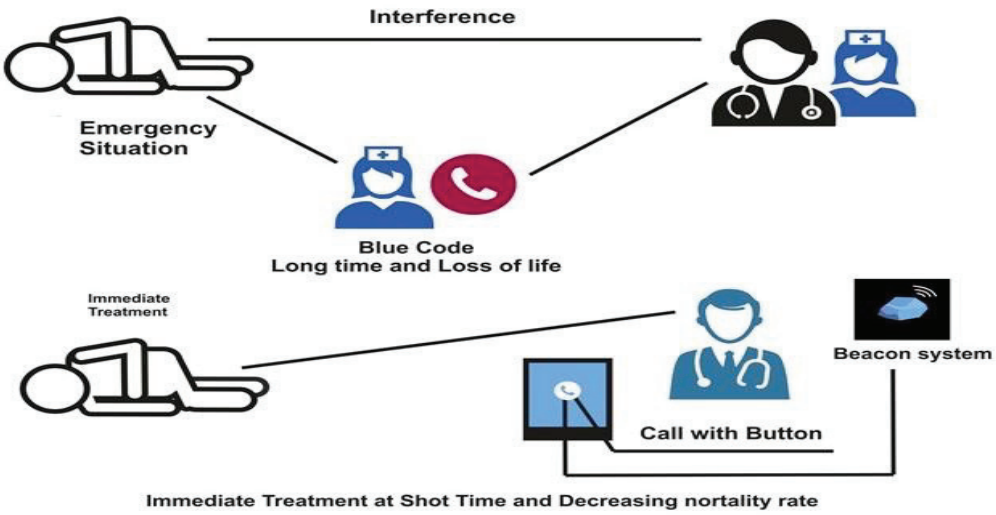


Figure 1.3. Our contribution to the problem

1.4 Our contribution to the problem with our thesis

To solve these problems, we have created a project which includes three basic stages which are developing mobile applications, defining location with Beacon system and set up the system hereby communication will be provided inside the hospitals. Mobile applications provide the emergency call and metronome to apply right cardiac massage, beacon systems provide to find place where the patient is. Application scans the bluetooth low energy devices and detect the closer one then sends text message automatically to the code blue team. The system will prevent to lose inside hospital and it is the most important part of the project. The aim of our project is providing easily communication between people. However, emergency team interfere on time and prevent the death inside the hospitals.

Our system firstly will be tested at the chemistry lab of Izmir Institute of Technology. After that it will be set up to Izmir Tepecik Training and Research Hospital and will be tested with test scenarios in laboratory at the last stage. These tests are

important in that we cannot see and improve the problems of the system. At Research hospitals, each building will have Blue Code team, call center so incoming calls will be transmitted to the basic call center which steer the nearest team to the patient. The team work coordinately to prevent the loss of life inside the hospitals.

CHAPTER 2

MATERIALS AND METHODS

2.1 Abstract

The aim of developing mobile application is to report the emergency situation to the staff inside hospital. Mobile applications will be created on Android and Ios operating system, so it is available for whole mobile phones and application will be created to easily use. Android application is created on Android studio which is used java programming language and Ios application is developed on XCode that is used swift programming language through every person can communicate with emergency team.

2.2 Problems of Current Situation

In hospitals, there is an emergency call center which means all calls are accumulated at there. Sometimes the working people at emergency call center can go out the hospital with a reason for a little. A call can reach the center, but nobody replies so emergency team cannot interfere the situation and people die.

2.3 Our Contribution to The Problem with Our Thesis

Mobile application will be completed on Android and Ios operating systems which known as Hybrid system hereby will be attained plug-compatible system with all mobile phones. Mobile application has a button that provides emergency call to team then they can easily interfere the patient on time. Also, application has a metronome which supply the heartbeat sound to do cardiac massage rightly.

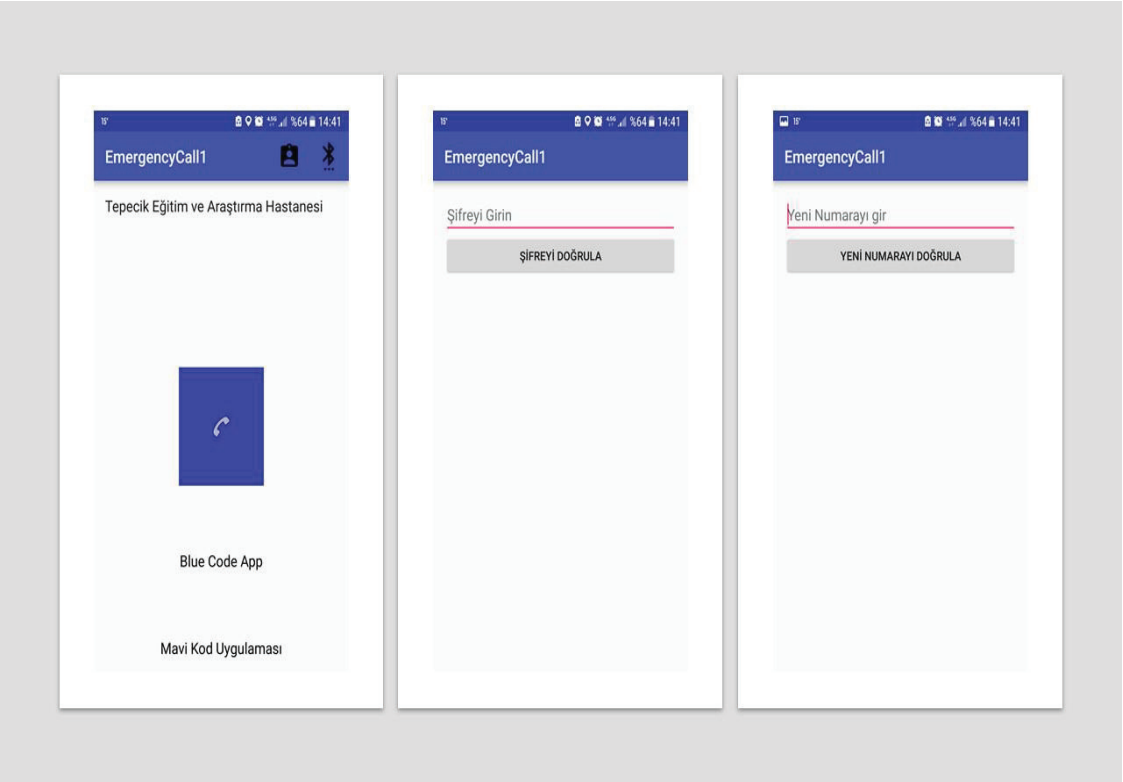


Figure 2.3. Our mobile application designs
(Source: AndroidStudio emulator)

They can change phone number easily which provides coordination between people. Before changing number, application will be asked password to change number. If password is true, new page will be seen which easily enter new phone number. After changing the phone number, application will be called the saved number.

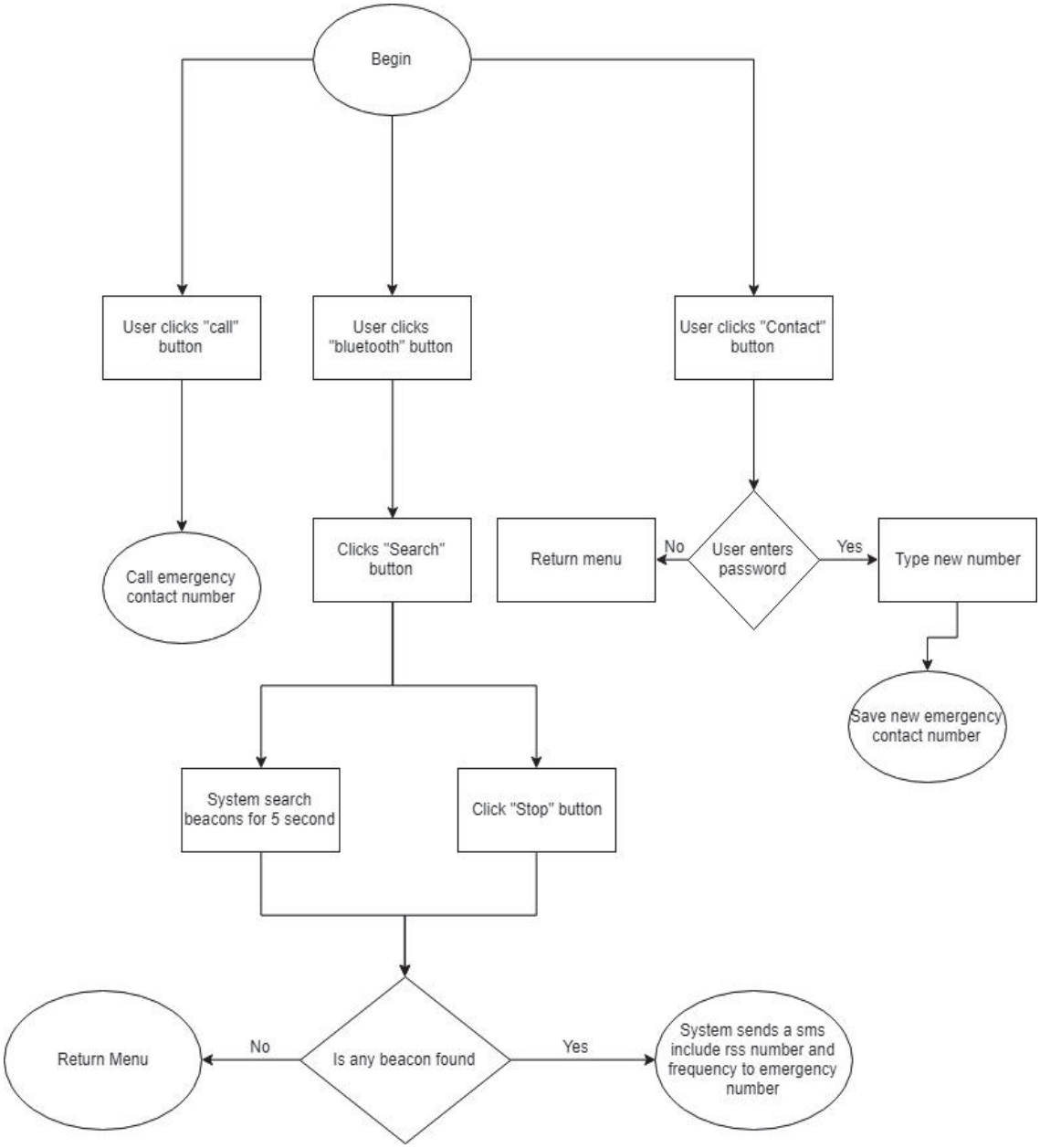


Figure 2.4. Flowchart of our application

CHAPTER 3

ANDROID OPERATING SYSTEM

(Source: developer.android.com)

3.1 Android Operating System

In this chapter, we will talk about android operating systems in details. Android operating system is a mobile system which developed by Google. It is developed to use on smartphones. Many developers created application for android operating system that they generally developed these applications on Android Studio.

On this platform, applications are written in Java language using the Android Software Development Kit (SDK). This SDK provides utilities such as debugger, software libraries and emulator. Android uses a variety of methods to get the most out of power usage and to use memory well. Taking unused applications into standby mode; Some of these methods are to turn off long inactive applications in case of insufficient memory.

Functions such as gyroscope, proximity sensor, accelerometer, which the operating system includes as hardware, are also beneficial for application developers. For example, the rotation of the image when the phone is turned sideways can be applied with the help of these tools.

Android is primarily developed for 32-bit ARMv7 processors. The Android x-86 project also provides support for x86 processors. Google takes care to publish a new version every six to nine months. In terms of security, Android also prevents applications from interfering with all areas of the system to be strong. For each field that each application wants to use, the user must get permission (Gargenta 2011).

3.2 Mobile Application on Android Studio

Android Studio is the official Integrated Development Environment for Android application development, based on Java. It offers even more features that enhance your productivity when building Android apps, such as a flexible gradle-based build system and the project structure in Android Studio includes one or more modules with code files and resources. In Android Studio, you can also customize the view of the project files to focus on specific aspects of your app development and you can search across your source code, databases, actions. Android studio provides an opportunity to create an application without using visual studio. Android is Linux based and open source operating system. We will explain the four main layers of Android.

- Linux Kernel
- Libraries and Android Runtime
- Application Framework
- Applications

3.2.1 Linux Kernel

The lowest level in the Android architecture. This layer contains hardware information and the drivers needed to run applications (keyboard drivers, audio drivers, Wi-fi drivers, camera, image and video drivers, processing and memory control, power control).

3.2.2 Libraries

The ground layer above the kernel is usually written in C ++ and C languages. It includes libraries (libc, SSL). In this layer system libraries, mp3, mpeg4, jpg for media libraries and 2D / 3D graphics for multimedia OpenGL / SGL libraries are available. The Android Operating System can hold its relational data in an SQLite named has a database. SQLite libraries for databases at the library level as well as basic libraries.

3.2.3 Android Runtime

It is the most important layer that distinguishes Android from mobile Linux. Android alt level tasks (such as memory management, hardware drivers), the Linux kernel and contains basic Java libraries. Core in this layer Libraries and Dalvik Virtual Machine.

3.2.3.1 Core Libraries

Core libraries are included core APIs for Java, data structures, services, file access, network access and graphical components.

3.2.3.1 DVM – Dalvik Virtual Machine

Dalvik Virtual Machine (DVM) is the most important component of Android operating system. Android is a single device using DVM which allows multiple instances to run efficiently with using a traditional Java VM, such as Java ME. Low-level functionality such as DVM, thread, memory management, operations and security use the Linux kernel on the device to put it in the path. DVM, minimum memory run Dalvik files, an optimized format for use. DVM summarize;

- The memory is efficient.
- Ensures consistency in application mobility and operation.
- Designed for embedded environments.
- Converts the structure of Java class files into the optimized dex format.
- Dex files are run with Dalvik bytecode.
- A separate Dalvik virtual machine is run for each process.

3.2.4 Application Framework

Android offers a very rich platform for software developers. Android application framework that allows the application framework to application services are as follows;

▪Activity Manager: Control the life cycle of your activities would. Includes management of the activity stack.

▪ Views: Used to make user interface for activities.

▪Notification Manager: Notifications made to the user and provide consistent and consistent functionality for warnings.

▪ Content Providers: Allows the app to share data. Telephone book, picture, music etc. access to data They are the interface. SQL-like access interface.

▪Resource Manager: Stores for outdoors and support code-free resources such as graphics.

3.2.5 Applications

It includes local and third-party Android applications built using services and classes in the Android app framework. Local applications include basic applications such as email client, text message program, calendar, google maps, phone book.

3.3 Android Lifecycle Activity

The product's software can be used both during production and during customer use all the stages that it spends are called the software development life cycle. The software development process consists of timed and contented stages. At this point the software is being developed in a planned way. Because of the requirements for software functions are constantly changing and expanding, the phases are handled in a continuous loop. It is in the loop to go back and improve at any stage. Basic software development steps are as follows:

Planning: Software is the first phase of the life cycle. Basic needs are identified, feasibility studies are carried out for the project (costs and benefits of the system and project planning is carried out.(Rogers et al. 2009)

Analysis: The functions and precise requirements of this target system are clear and to document them in a specific format as a result. This work includes client, software engineer, system analyst, business analyst, product manager, etc. It can be done by the groups in which the roles come together. In situations where the needs are unclear, communication and collaboration between the software engineer and the customer should be much more. At this stage in various software development methodologies use documents and test plan documents can also be created.

Design: The system design phase begins with the completion of the requirements. Software product design is the activity of determining the features, capabilities, and

interfaces of the software product to meet the needs and desires of the customer. It is possible to talk about two types of design (high level design - architectural design and detailed design). Architectural design deals with the general structures of software modules and their interactions within the organization. As a result, architectural design documents are created. During detailed design, architectural design documents are generally revised. One of the most important techniques used in software design is Abstraction. Abstraction is the elimination of certain features of objects, events, and situations to facilitate the solution of problems. It simplifies the problem and allows us to focus on the most important parts. Modeling is the basic design tool and it is possible to talk about static and dynamic models. The static model is used to express the unchanging aspects of the program (class and object models), the dynamic model is used to express the behavior of the program during its execution (state and sequence diagrams).

Implementation (Coding and Testing): The design stage has a certain degree of coding phase begins with the arrival. The customer is the stage of programming the product to be delivered. Good code, easy to read and simple to maintain. According to the KISS (Keep it simple) principle, if a new graduate can understand and change the code written in 1-2 days, the code written is a good code. The other important step during coding and after coding is the test. Having a early test approach and having a test perspective from the analysis stage will reduce the error rate and costs (time, money, prestige, etc.). There are many different categories and depth of test types that can be applied according to the process and situation such as unit tests, smoke tests, false test, acceptance tests, usage scenario tests, load tests, user acceptance test, test automation.

Delivery and Maintenance: After all the test steps have been completed, a deliverable version is removed, and the delivery phase is carried out. As a delivery outlet, the product alone is not enough. The user manual and version difference document must be created for end users. The maintenance phase also starts with delivery. There are different maintenance activities such as debugger, preventer, infrastructure optimizer, new feature generator for the product (Rogers et al. 2009).

Android is created around the specific conditions of mobile applications. Resources are restricted on mobile devices and provides mechanisms to descant those resources. These mechanisms have been obviously seen in the Android Lifecycle Activity, which determines the events or states.

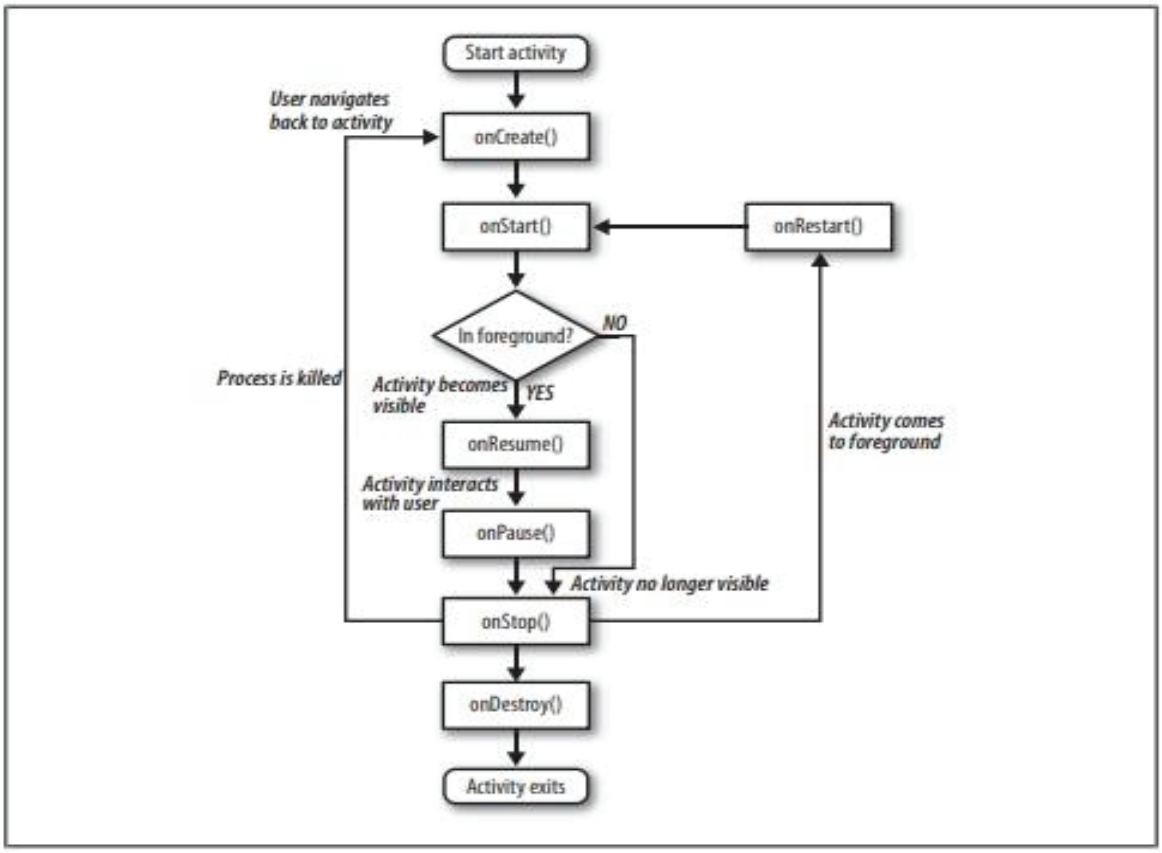


Figure 3. 1. Life cycle is shown
(Source: Rogers et al. 2009)

onCreate:

It creates your first activity and opens what your activity needs to use. In general, it sets to start your activity.

onStart:

Called before your activity print on screen. If your activity can start such as the foreground activity on the screen, control will assign to onResume.

onResume:

It is also called if your activity stops the foreground to another activity, and that activity directly exits.

onPause:

Your activity will no longer have shown to the screen, so you should stop consuming battery and CPU usage unnecessarily.

onStop:

It is to your activity is not visible, because another activity access on the foreground.

onDestroy:

It is the last call to start activity before your activity is ejected.

These methods have advantages which simplify the writing code in Android Studio. It provides better user experience and lifecycle activities determine what needs to be done that we do not lose inside the codes. These activities are used in the classes which are occurred the structure. It is up to you what kind of creating java class, but you have to give permission in Android Manifest file(Rogers et al. 2009).

3.4 Visual Components

We have provided visual assets to help us design our apps by Google. The basic elements that come with the Android SDK and will be used in the layout files are listed as follows:

3.4.1 TextView

It used to display text values on the screen

3.4.2 EditText

It is used to send input such as text or number to be taken from the user to the source code side. It shows input-like behavior in HTML. The keyboard type can be selected according to the value that the user will input when logging in.

3.4.3 ImageView

ImageView is used to display pictures on the screen. ImageView files from the drawable folder or from another source. It is suitable for viewing.

3.4.4 Button

The Button item can be used to create a button. When the user presses the button, an action can be assigned to the button by executing code in a listener (onClickListener) created in the code.

3.4.5 ListView

It usued to create a table of rows. It is suitable for showing data fetched from a specific source (database or web service). It comes at the top of the most commonly used visual items in applications. The ability to replace row designs with custom layout files provides a flexible use(Rogers et al. 2009).

3.5 Content of Mobile Application

We firstly created these steps on android studio. First step is to make a call. Android application will be developed on Android Studio which is enable for creating application. It has been created for one month. Android application is created to use Froyo 2.2 android version base so that developing version can be easily use. Firstly, I have constituted making call button. Several codes are typed to provide easily calling, we named that urgent call button. The button does not provide just calling feature that also provides starting metronome which during urgent call, metronome is cut-in which decreases the wrong cardiac massage. We have added a new page to the application that provides changing the number and saving the new number feature. The aim of this feature is to provide coordination people inside the hospitals. Every person saves a number to call someone during urgent situation. The third step is the most important activity which is scanning the bluetooth low energy devices. We can scan the bluetooth low energy devices and list them in ListView. With a method we can sort the their RSSI and pick the closer and send message to the team to determine the location of the patient.

3.6 Tests on Mobile Phones

Mobile application firstly generated at Android Studio after tested in laboratory at several type of phones.

Brand	Version
1.Samsung S7 Edge	Android 6.0
2. Asus	Android 4.4
3.Samsung S7	Android 6.0

4.Lenovo Vibe P1 pro	Android 5.0
5.GM Discovery II	Android 4.4. 2
6.LG G3	Android 4.0
7.LG G4	Android 5.0
8.Xiaomi Max 2	Android 7.0
9.LG G5	Android 7.0
10.Samsung Note 3	Android 6.0
11.Sony Xperia Z	Android 5.0
12.Vestel Venus	Android 4.4.2
13.Samsung S8	Android 8.0
14. Samsung J7	Android 7.0

CHAPTER 4

iOS OPERATING SYSTEM

(Source: apple.com)

4.1 iOS Operating System

iOS operating system is the Apple's operating system for mobile devices which works on iPad, iPod Touch, iPhone. A software development kit named iOS SDK provided by Apple is used. This software development kit is written using C and Objective - C programming languages, providing constructs to allow us to define classes and objects. Once you get the hang of the Smalltalk-style syntax, if you have programmed in an object-oriented language before, things should look familiar.

However, there are some differences. One of the bigger differences is in how Objective-C deals with management of memory, although this has been massively clarified with the arrival of iOS 5(Allan 2013). This operating system is called iPhone operating system until iPad emerged so that Apple Company had to change the name. The letter i at the beginning of the iOS abbreviation represents the prefix Apple has used in its products such as iTunes and iMac. Therefore, iOS is the new documentation provided by Apple.

4.2 Layers of the iOS Operating System

Unix-based operating systems, providing communication with hardware, processor and a component called the kernel that manages the file system and basic

network operations.The iOS operating system consists of four service layers configured roughly on a Unix-based core.

Table 4.1 Layers of iOS Operationg System
(Source: Allan 2013)

4	Cocoa Touch Layer
3	Media Services Layer (Media Services)
2	Core Services Layer (Core Services)
1	Core Operating System Layer (Core OS)

4.2.1 Core Operating System Layer (Core OS)

The bottom layer, the Core OS layer, is the layer closest to the kernel and containing services at the lowest level. The services used on this layer are often confused when the iOS application is being developed. In the case of symptom outbreaks, application frameworks that use these services are used instead of these services. Some of these application frameworks are given below.

Accelerate Framework: This application framework, accessible from the Accelerate framework namespace, includes a variety of interfaces to perform high-precision mathematical operations and digital signal processing calculations.

Security Framework: This framework of applications, accessible from the security framework namespace, contains several libraries that ensure the security of the data used in applications. With this application framework, security policies, public and private keys can be generated and managed.

External Framework: Accessible from the external framework namespace, this framework includes libraries that are needed to communicate with and manage other devices connected to iOS operating system devices.

Lib-System Framework: It contains various interfaces that surround the iOS kernel and communicate with other layers.

4.2.2 Core Services Layer

This layer contains a variety of application extensions that all applications use. Foundation application framework is also in this layer. Some of the application overlays in the core services layer are given below:

Address Book Framework: With this application framework, you can access the iPhone address book from within Objective - C and add, delete and modify entries.

CF Network Framework: This application provides C programming language-based interfaces that provide access to sockets for applications that use the framework TCP / IP protocol.

Core Data Framework: It can provide the requirement to save the application's state information and access the saved state information when needed, in accordance with the MVC (Model-View-Controller) design template.

Core Foundation Framework: Includes functions that make basic data types that we use while programming more useful.

Core Location Framework: This application framework is an application framework that is used for Geographic Information System-like applications that generally require and operate on the geographical location of the device.

Core Media Framework: Allows you to manage operations such as accessing and executing audio and video files from within our own applications.

Store Kit Framework: This application framework allows you to business transactions can be done safely.

System Configuration Framework: from within developed applications access to system configuration and network connection information allow me to provide.

4.2.3 Media Services Layer

The main purpose of this layer is to provide audio, video animation and graphical transaction skills. For this reason, application lovers located in the media services layer essentially operate on audio, video and image files. Some of these application frameworks are given below:

Assets Library Framework: It contains the classes that allow the information of files such as videos, photos in iOS operating system devices to be retrieved and processed on these files.

AV Foundation Framework: On supported devices, we can play audio files, play them and save new audio files.

Core Graphics Framework: We can create and manage 2D graphics.

Core Midi Framework: iPhone / iPad charger interface (dock) relates to the keyboard and similar devices are communicated between our application.

Core Text Framework: Allows advanced text formatting to be performed and font management within text.

Image Input / Output Framework: Contains various classes and methods for manipulating image files.

Media Player Framework: Provides necessary interfaces for executing compressed video files in known formats such as mov, mp4, m4v and 3gp from developed applications.

Quartz Core Framework: The developed application provides various animation capabilities.

Open Audio Library: Makes it possible to produce sophisticated sound effects, especially in games.

4.2.4 Cocoa Touch Layer

This layer is the layer at the top of the iOS operating system design and often contains application defragmentation that are often used by iPhone application developers. Cocoa Touch is known as an API (Application Development Interface). This API; It is a customized version of the Mac OS X Cocoa API for touch devices, which defines various controls used by the user-facing side of the Mac OS X operating system. The app overlays and features in this layer are listed below:

Address Book UI Framework: With this framework, you can access your phonebook from within the application, add new links, edit and delete it.

Game-Kit Framework: With this application framework, it provides mutual connection, data and voice communication in multi-player games.

Map Kit Framework: Gives developers the ability to perform operations on maps supported by iOS devices on developed applications.

Message UI Framework: With this application framework, it is possible to send e-mail messages within the applications we have developed.

iAd Framework: This application framework enables us to banner advertisements

iAd Framework: This application framework enables us to banner advertisements from developed applications and to manage the advertisements.

UIKit Framework: Contains controls and related classes that allow us to interact directly with the user in the application.

Event Kit UI Framework: This application framework allows us to access calendar contents from developed applications.

Push Notification Service: Allows me to generate notifications that users may notice within applications.

4.3 Mobile Application on XCode

The most important factor in enabling Objective - C to be revolutionized is the use of the Software Development Kit called iOS SDK, which allows Apple to develop native applications for the operating system for devices using the iOS operating system. This kit, which was opened by Apple company in February 2008 for developer use, allows developers around the world to develop their own applications. XCode is very useful program to develop an application because the program is object oriented that means coding is not complex.

4.3.1 Objective C

iPhone applications are programmed in Objective C. The Objective C programming language is an object-oriented language introduced by adding object-oriented programming features to the ANSI C programming language. Objective-C is an object-oriented programming language produced by a software company named StepStone at the beginning of 1980's. The first design of the programming language was made by Brad Cox and Tom Love.

Objective C classes come in the form of interfaces and implementations. In the interface section, the properties and methods of the class are defined, and the codes of the methods are in the application section. When you define a new class, you need to specify the top class of this class, what kind of data is stored in the class, and which methods can be called on class objects. The classes that are defined in Objective C program files are automatically the top-level class, the non-native object class.

Methods do not execute operations on objects of the class that are defined for the class are the operational code units that are used. If you define a method, Objective C compiler (compiler) must declare if the method can return and return a value, and if so, what type it is. The way to do this is to write the type of the returned value before the method name in parentheses (). The code of the class methods is passed in the

implementation section of the class. Definition section of class defined by interface keyword an application partition is created with the implementation keyword. Once the classes are defined, the main program section to be used for the classes is defined. The main program section has a method called 'main', similar to the Java programming language, and the Objective C compiler automatically invokes this method when the program is run.

4.4 Tests on Mobile Phones

1.iPhone 5S	iOS 7
2. iPhone 6	iOS 8
3. iPhone 6S	iOS 9.0
4.iPhone SE	iOS 9.3
5.iPhone 6S Plus	iOS 9.0
6.iPhone 5S	iOS 7
7.iPhone 6S	iOS 9.0
8. iPhone SE	iOS 9.3
9. iPhone 5S	iOS 7
10. iPhone 5S	iOS 7

CHAPTER 5

BLUETOOTH LOW ENERGY

5.1 Abstract

Beacon system will be built thus the person, who needs to immediate treatment, can be easily determined the location by emergency team. The aim of second stage is to interfere the person on time in this way team will gain time and restore the person to life. Beacon system is worked with Bluetooth which is wireless technology for converting data over short distances from mobile devices. This technology is used since 1994 in the areas of telecommunication, computing, networking. We used Puck.js beacon system which is open source. It includes semiconductor, accelerometer, air pressure sensor, Li-ion battery and also it includes mobile application to control from a distance.

5.2 What is the Bluetooth Low Energy

Bluetooth low energy is part of the Bluetooth 4.0. This Technology is smaller, highly-optimized version of Bluetooth but it has different design and lineage. Nokia designed BLE technology that known as Wibree. The purpose of this technology design is to create radio standard with the lowest power consumption, optimized for low cost, low bandwidth, low power and low complexity. Devices compliant with this standard are expected to consume very low power so that they can operate for months or even years on coin cell or smaller batteries without the need of recharging or replacing batteries. This is very useful in applications where it may be difficult to recharge frequently, and longer battery life is important. Data communication is generally in short bursts that do not need to be very frequent. It is best suited for devices that do not require high throughput or streaming of data. Some of the key features of Bluetooth Low Energy are:

Ultra-low power, which enables months or even years of operation on coin cell or smaller batteries;

Small size;

Low cost;

Short range;

Secure;

Interoperable.

Bluetooth Low Energy finds a huge variety of applications including the following:

Health care devices such as blood pressure monitors, glucose meters;

Sports and fitness equipment such as smart watches, pedometers, GPS locators, and heart rate monitors;

Home automation;

Home entertainment, remote controls, wireless keyboards;

Smart energy (meters and displays);

Advertisements;

Mobile payments;

Automotive devices;

Security.

So many advantages have assembled around Bluetooth Low Energy and we can buy system-on-chip solutions for under $ 3 per chip and in low volumes, which is good below the whole overall price point of similar wireless technologies such as GSM, Zigbee, WiFi, etc. Bluetooth Low energy allows to design achievable products that can communicate with any modern mobile platform via tools, chips and standards that are simple to access.(Heydon 2012)

One of the factors contributing to the accomplishment of Bluetooth Low Energy is that it was created to serve as an extensible framework to exchange data. Bluetooth Low Energy goes a completely different way. It has been optimized for low-power consumption instead of increasing data rates. Wired and wireless technologies increase speeds, as showed in Table 5.1(Heydon 2012).

Table 5.1: Speeds Almost Always Increase
(Source: Heydon 2012)

Modems	**Ethernet**
V.21: 0.3 kbps	802.3:10 Mbps
V.22: 1.2	802.3u:100 Mbps
V.32:9.6	802.3ab: 1000 Mbps
V.34: 28.8	802.3an:10000 Mbps
Wifi	**Bluetooth**
802.11: 2Mbps	v1.1:1 Mbps
802.11b:11 Mbps	v.2.0:3 Mbps
802.11g: 54 Mbps	v3.0:54 Mbps
802.11n:135 Mbps	V4.0:0.3Mbps

This different way has been reached through the understanding that Bluetooth technology cannot reach the low power requisites required for button-cell battery devices. However, to completely understand the requisites around low power, another thought must be taken. Bluetooth Low Energy is created to be applied in high volumes, today's devices do not use Wifi technology. The low cost is come from their compact and small design and it should be efficient.

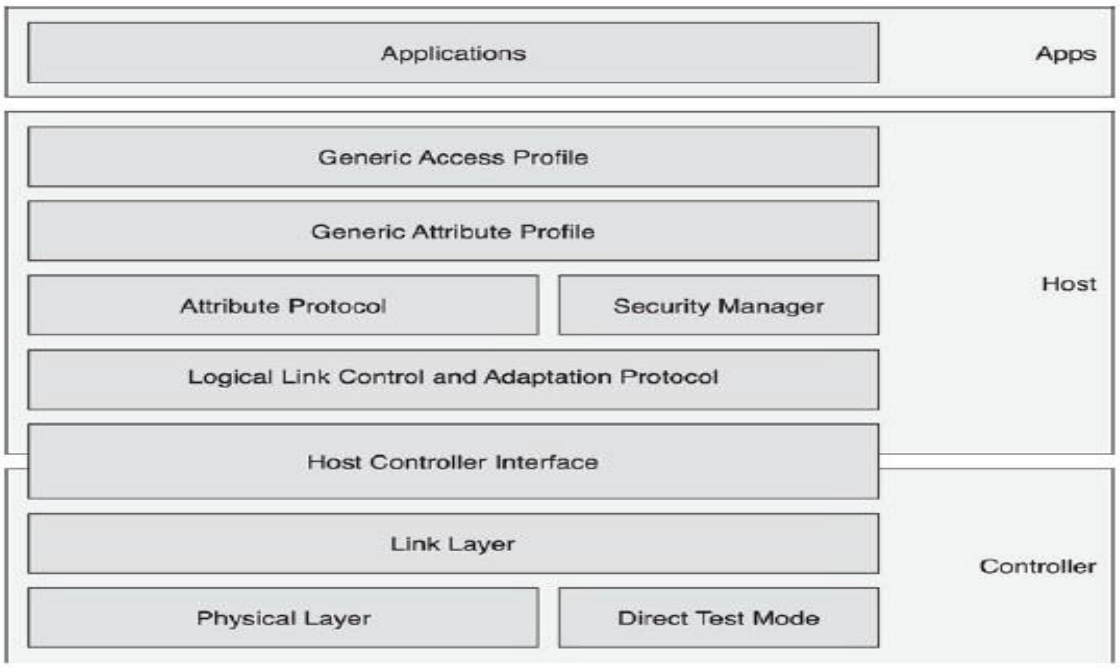


Figure 5.1: Architecture of Bluetooth Low Energy
(Source: Heydon 2012)

Bluetooth Low Energy uses the 2.45GHz bandwidth and it has two device type which are single and dual mode. Device types can interact with each other. Single mode can interact single mode and dual mode. Dual mode can communicate dual mode.

Architecture of Bluetooth Low Energy devices split into three parts: host, controller and applications. The controller is a device that receives and transmits radio signal and explains how these signals can be used. The host is the software part that helps to manage the communication between devices. The applications use the host.

There is both the Physical Layer and Link Layer as well as a Direct Test Mode and the lower layer of the Host Controller Interface. Within the host are three protocols: Logical Link Control and Adaptation Protocol, Attribute Protocol, and the Security Manager Protocol also within the host are the Generic Attribute Profile, the Generic Access Profile, and modes.

5.2.1 Controller

The controller is composed of both analog and digital parts of the radio frequency components as well as hardware to support the transmission and reception of packets. The controller interfaces with the outside world through an antenna and to the host through the Host Controller Interface.

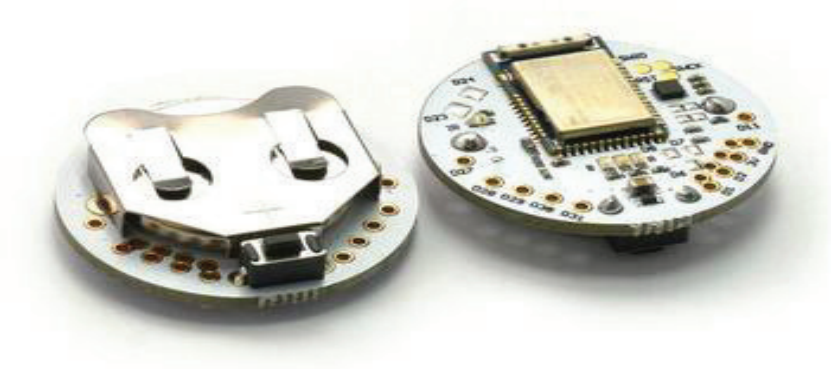
5.2.2 Physical Layer

The Physical Layer is the bit that does the hard work of transmitting and receiving bits using the 2.4GHz radio. Radio waves can carry information by varying the amplitude, frequency, or phase of the wave within a given frequency band. In Bluetooth low energy, the frequency of the radio waves are varied to allow either a zero or a one to be exposed, using a modulation scheme called Gaussian Frequency Shift Keying (GFSK). There is a pulse of energy that expands out over a wider range of frequencies, if the frequency is shifted abruptly to one side or the other at the moment the frequency changes. Filter is used to break the energy expanding too far into higher or lower frequencies. In the case of Gaussian Frequency Shift Keying, the filter used for Bluetooth low energy is not as tight as the filter used for Bluetooth classic. This means that the low energy radio signal spreads out a little more than the classic radio signal.

5.2.3 Direct Test Mode

Direct Test Mode is a novel approach to the testing of the Physical Layer. In most wireless standards, there is no standard way to get a device to perform standard Physical Layer tests. This leads to the problem of many different companies building their own proprietary methods to test only their Physical Layers. This increases the costs for the whole industry and increases the barriers for an end-product manufacturer to change from one silicon supplier to another quickly.

5.2.4 Link Layer



The Link Layer is probably the single most complex part of the Bluetooth low energy architecture. It is responsible for advertising, scanning, and creating and maintaining connections. It is also responsible for ensuring that packets are structured in just the right way, with the correctly calculated check values and encryption sequences. To do this, three basic concepts are defined: channels, packets, and procedures. There are two types of Link Layer channels: advertising channels and data channels. Advertising channels are used by devices that are not in a connection sending data. The data channels are only used once a connection has been established and data needs to flow. There are 37 data channels, and

they are used through an adaptive frequency-hopping engine to ensure robustness. BLE devices contain button cell batteries such as CR2032 that has 230mAh energy capacity at 3V. Generally, it provides usage eighty percent of energy. We use puck.js that primary magnetometer, IR transmitter, NFC tag, thermometer. It has simple design.

Technical Properties

- nRF52832 SoC - 64MHz ARM Cortex M4, 64kB RAM, 512kB Flash
- 8 x 0.1" GPIO (capable of PWM, SPI, I2C, UART, Analog Input)
- 9 x SMD GPIO (capable of PWM, SPI, I2C, UART)
- ABS plastic rear case with lanyard mount
- Silicone cover with tactile button
- MAG3110 Magnetometer
- IR Transmitter
- Built in thermometer, light and battery level sensors
- Red, Green and Blue LEDs
- NFC tag programmable from JavaScript
- Pin capable of capacitive sensing
- Weight: 14g in plastic case, 20g in packaging
- Dimensions of cardboard box: 64mm x 62mm x 16mm
- Dimensions of plastic case: 36mm dia, 12.5mm thick
- Dimensions of bare PCB: 29mm dia, 9mm thick

Puck.js has software platform to programming easily. Software platform are prepared with JavaScript that is a script developed to bring web pages into interactive mode. The abbreviation is JS. It is not server based; user based which provide an opportunity to easily programming the BLE devices. We use their rssi values to pinpoint. Application detects the closest RSSI value and send message the room number with rssi.

CHAPTER 6

RESULTS

6.1 Testing at Laboratory

Beacons are replaced to inside of the doors of chemistry laboratories and tested via android mobile application. We have tested at seven rooms which five of those are laboratories and the one of those is hallway and the other one is toilet. The reason of these replacement is to constitute similar plan before testing at hospital. Test results have showed that beacons do not replace the nearby the doors. We should transpose to inside of the room.

Second test was tried that beacons are located at the inside of the rooms. It was observed that the project was running smoothly. Android application scanned the Bluetooth Low Energy devices and listed from the nearest to farthest. Send the information of the nearest Bluetooth Low Energy device as text message. During second test, we recognized an error which is contact list. We save a phone number to the application to make phone call and we present change number option, however it does not save the new number. It was an error to fix. We used SharedPreferences feature to do contact list. SharedPreferences has present edit and save the contact list. After this correction, we continue to test the system.

6.2 Testing at hospital

Our system is tested at Surgeon Oncology Service of the Tepecik Training and Research Hospital. Seven of the beacon are located at the Surgeon Endoscopy Service as shown in Figure 6.1.

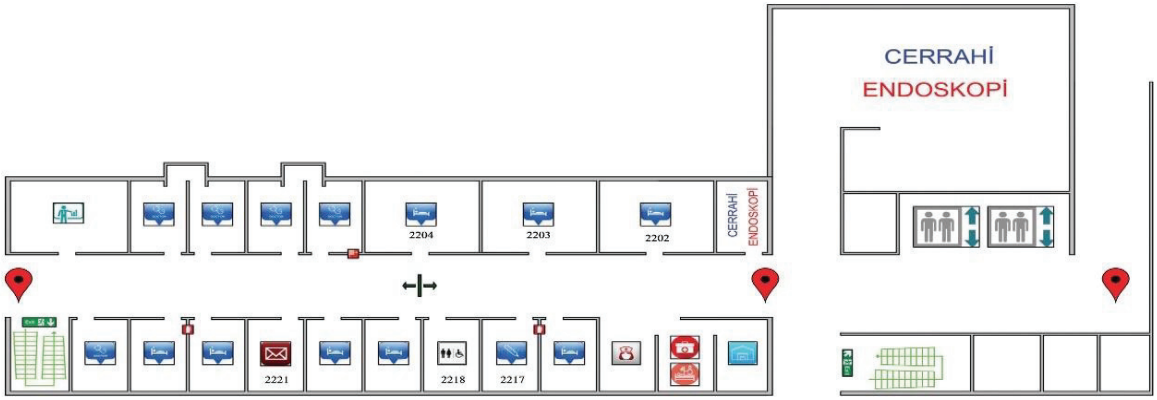


Figure 6.1: Plan of the Tepecik Training and Research Hospital

Measurements are taken at distance=30cm, same floor, downstairs, yard, computed tomography room and out of the hallway and their signal strength are compared with each other. The cursors show testing point at the same floor in Figure 6.1. When examined the signal strengths of the events at 30 cm and the same floor; there is difference between the mean of the 30 cm and the same floor. Therefore, values were found to be statistically significantly lower. (p:0,017; p<0,05)

Table.6.1. Statistical Data Result 1
(Source:SPSS)

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	Distance Rssi	-66,03	40	1,641	,259
	Same	-72,98	40	17,182	2,717

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Dev.	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	30cm Same	6,950	17,640	2,789	1,308	12,592	2,492	39	,017

p:0,017 < p:0,05 showed that signal strengths are changed at different distances. The difference in signal strengths on the same floor and 30 cm distance indicates that the use of signal strength is effective. Application can detect the right signal and send it to the phone number.

Table 6.2. Statistical Data Results 2
(Source: SPSS)

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 2	Distance Rssi	-66,03	40	1,641	,259
	Downstairs Rssi	-82,28	40	15,913	2,516

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Dev.	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 2	30cm - Downstairs	16,250	15,858	2,507	11,178	21,322	6,481	39	,000

The Table 6.2 showed that the comparison between 30 cm distance and downstairs examines. We located the one beacon to the downstairs of the Surgeon Oncology Services. Other beacon devices are located at the upstairs. Walls do not pass the received signal strength indicate that application only detects the beacon at the hallway of the same floor (p:0,000; p<0,05).

Table 6.3: Statistical Data Results 3
(Source:SPSS)

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 3	Distance Rssi	-66,03	40	1,641	,259
	Out of Hallway Rssi	-90,10	40	7,759	1,227

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Dev.	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 3	30cm – Out	24,075	8,257	1,306	21,434	26,716	18,441	39	,000

In these tables, we compared the out of the hallway of the same floor and the examination of 30 cm distance. Rssi values decrease when we have moved away from the beacon located area that the difference of the Means prove that the signal strength is farther out of the hallway.

Table 6.4. Statistical Data Results 4
(Source:SPSS)

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 4	Distance Rssi	-66,03	40	1,641	,259
	CT Rssi	-76,35	40	16,254	2,570

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Dev.	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 4	30cm - CT	10,325	16,066	2,540	5,187	15,463	4,065	39	,000

We have tested beacon devices at Computed Tomography Room. Tomography room at the ground floor. We have located the beacon devices to the five points of this floor and CT room. Measurements are showed that lead coated room does not transmit the bluetooth low energy. Application did not perceive the beacon devices at the out of Computed Tomography Room when I am at the Computed Tomography Room. Application only detects the other devices during the doors open but sent the closest devices.

Table 6.5: Statistical Data Results 5
(Source: SPSS)

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 5	Distance Rssi	-66,03	40	1,641	,259
	Yard Rssi	-87,53	40	12,019	1,900

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Dev.	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 5	30cm - Yard	21,500	12,125	1,917	17,622	25,378	11,214	39	,000

Six beacon devices were placed far from each other in yard. Application detected the closest beacon device all time. The Table 6.5 shows that there is a significant difference between 30 cm measurements and yard measurements.

CHAPTER 7

CONCLUSION

Pagers and payphones are still used for emergency situations inside the hospitals. Code Blue team must interfere in five minutes, but the systems they use do not provide first aid on time. Therefore, the main objective of this research has been to prevent the mortality rate inside the hospital via interference on time. The project contains mobile application and beacon devices. Mobile application provides easily phone call, detects the location of patient and sends a text message the room number where the patient is located and starts a heart-rate rhythm metronome. There is a button on main page which is started the phone call. When phone call is finished, heart-rate rhythm metronome starts to play. The application is available for every android and ios operating systems. It was installed and tested on different devices. The system has been tested at Tepecik Training and Research Hospital. Test results are successful as we planned. We want to create easy system to use, configure and develop.

There are some limitations in this project. Mobile application has been developed on Android Studio for Android phones. It has been written on Xcode for iPhones. There was a lot of time lost due to the frequent updates and the errors associated with these updates. There is a huge resource shortage especially in iOS app development so that developing application takes long time. In application, selection of the closest beacon devices is the most challenging part. During the beacon device scan, the application cache is full, and the application is giving a restart failure. After we saved the new phone number, when we turned off the app and restarted it, we noticed that it called the old number. The most important error is that the application has detected the intelligent car keys, smart televisions with bluetooth low energy feature during the beacon device scan. In beacon devices, programming is easy, however the cost is expensive. The price of beacon devices we bought from abroad for the trial is costly if the number of rooms of hospitals is considered.

It would be better to develop an application on a single platform for both Android and iOS operating systems. The problem of registering numbers can be solved

by adding SharedPreferences. The problem of stopping the application has been solved by adding clear
cache code to the application. The detection of an intelligent car key or device is still a problem and a solution has not been reached.

The system can become a commercial product and can be installed to have technologically hospitals. In case of the code blue, the code blue team can perform the intervention in less five minutes.

On the other hand, the future of the system will be clear. The project will be updated with a new version. New version will contain maps of the hospitals. Beacon devices will be produced by our research group to reduce the costs.

REFERENCES

Allan, Alasdair. 2013. *Learning iOS Programming: From Xcode to App Store*: " O'Reilly Media, Inc.".

Bellomo, Rinaldo, Donna Goldsmith, Shigehiko Uchino, Jonathan Buckmaster, Graeme K Hart, Helen Opdam, William Silvester, Laurie Doolan, and Geoffey Gutteridge. 2003. "A prospective before-and-after trial of a medical emergency team." *Medical Journal of Australia* 179 (6):283-288.

Bradley, Steven M, Erin E Gabriel, Tom P Aufderheide, Roxy Barnes, Jim Christenson, Daniel P Davis, Ian G Stiell, and Graham Nichol. 2010. "Survival Increases with CPR by Emergency Medical Services before defibrillation of out-of-hospital ventricular fibrillation or ventricular tachycardia: observations from the Resuscitation Outcomes Consortium." *Resuscitation* 81 (2):155-162.

Buist, Michael D, Gaye E Moore, Stephen A Bernard, Bruce P Waxman, Jeremy N Anderson, and Tuan V Nguyen. 2002. "Effects of a medical emergency team on reduction of incidence of and mortality from unexpected cardiac arrests in hospital: preliminary study." *Bmj* 324 (7334):387-390.

Cooper, Jonas A, Joel D Cooper, and Joshua M Cooper. 2006. "Cardiopulmonary resuscitation: history, current practice, and future direction." *Circulation* 114 (25):2839-2849.

Cummins, Richard O, Douglas Chamberlain, Mary Fran Hazinski, Vinay Nadkarni, Walter Kloeck, Efraim Kramer, Lance Becker, Colin Robertson, Rudi Koster, and Arno Zaritsky. 1997. "Recommended guidelines for reviewing, reporting, and conducting research on in-hospital resuscitation: the in-hospital “Utstein style”." *Annals of emergency medicine* 29 (5):650-679.

Çete, Y. 2000. "Kardiyopulmoner resüsitasyonda son gelişmeler." *Acil Tıp Dergisi*:1-13.

Demirci, Mehmet Ali. 2016. "Hastanemizde mavi kod çağrılarının nedenleri ve doğruluğu."

Edelson, Dana P, Benjamin S Abella, Jo Kramer-Johansen, Lars Wik, Helge Myklebust, Anne M Barry, Raina M Merchant, Terry L Vanden Hoek, Petter A Steen, and Lance B Becker. 2006. "Effects of compression depth and pre-shock pauses predict defibrillation failure during cardiac arrest." *Resuscitation* 71 (2):137-145.

Eftestøl, Trygve, Lars Wik, Kjetil Sunde, and Petter Andreas Steen. 2004. "Effects of cardiopulmonary resuscitation on predictors of ventricular fibrillation defibrillation success during out-of-hospital cardiac arrest." *Circulation* 110 (1):10-15.

Eisenberg, Mickey S, Michael K Copass, Alfred P Hallstrom, Barbara Blake, Lawrence Bergner, Floyd A Short, and Leonard A Cobb. 1980. "Treatment of out-of-hospital cardiac arrests with rapid defibrillation by emergency medical technicians." *New England Journal of Medicine* 302 (25):1379-1383.

Eisenberg, Mickey S, and Terry J Mengert. 2001. "Cardiac resuscitation." *New England Journal of Medicine* 344 (17):1304-1313.

Eisenberg, Mickey S, and Bruce M Psaty. 2009. "Defining and improving survival rates from cardiac arrest in US communities." *JAMA* 301 (8):860-862.

Field, John M, Mary Fran Hazinski, Michael R Sayre, Leon Chameides, Stephen M Schexnayder, Robin Hemphill, Ricardo A Samson, John Kattwinkel, Robert A Berg, and Farhan Bhanji. 2010. "Part 1: executive summary." *Circulation* 122 (18 suppl 3):S640-S656.

Gargenta, Marko. 2011. *Learning android*: " O'Reilly Media, Inc.".

Goldhill, David R, Alistair F McNarry, Vassilis G Hadjianastassiou, and Paris P Tekkis. 2004. "The longer patients are in hospital before Intensive Care admission the higher their mortality." *Intensive care medicine* 30 (10):1908-1913.

Hazinski, Mary Fran, L Chameides, R Hemphill, and RA Samson. 2010. "Highlights of the 2010 American Heart Association guidelines for CPR and ECC." *Dallas, TX: American Heart Association*.

Henderson, Sean O, and Douglas Ballesteros. 2001. "Evaluation of a hospital-wide resuscitation team: does it increase survival for in-hospital cardiopulmonary arrest?" *Resuscitation* 48 (2):111-116.

Heydon, R. 2012. *Bluetooth Low Energy: The Developer's Handbook*: Prentice Hall.

Hillman, Ken M, Peter J Bristow, Tien Chey, Kathy Daffurn, Theresa Jacques, Sandra L Norman, Gillian F Bishop, and Grant Simmons. 2002. "Duration of life-threatening antecedents prior to intensive care admission." *Intensive care medicine* 28 (11):1629-1634.

Hou, Sen-Kuang, Chii-Hwa Chern, Chorng-Kuang How, Lee-Min Wang, Chun-I Huang, and Chen-Hsen Lee. 2007. "Is ward experience in resuscitation effort related to the prognosis of unexpected cardiac arrest?" *Journal of the Chinese Medical Association* 70 (9):385-391.

Iwami, Taku, Graham Nichol, Atsushi Hiraide, Yasuyuki Hayashi, Tatsuya Nishiuchi, Kentaro Kajino, Hiroshi Morita, Hidekazu Yukioka, Hisashi Ikeuchi, and Hisashi Sugimoto. 2009. "Continuous improvements in “chain of survival” increased survival after out-of-hospital cardiac arrests: a large-scale population-based study." *Circulation* 119 (5):728-734.

Keys, Vicky A, and Melissa Selzler. 2011. "Emergency code color standardization: implementation without traumatization." *Journal for Nurses in Professional Development* 27 (3):148-150.

Lown, Bernard, Jose Neuman, Raghavan Amarasingham, and Barouh V Berkovits. 1962. "Comparison of alternating current with direct current electroshock across the closed chest." *American Journal of Cardiology* 10 (2):223-233.

McGrath, Roland B. 1987. "In-house cardiopulmonary resuscitation—after a quarter of a century." *Annals of emergency medicine* 16 (12):1365-1368.

Nolan, Jerry P, Jasmeet Soar, David A Zideman, Dominique Biarent, Leo L Bossaert, Charles Deakin, Rudolph W Koster, Jonathan Wyllie, and Bernd Böttiger. 2010. "European resuscitation council guidelines for resuscitation 2010 section 1. Executive summary." *Resuscitation* 81 (10):1219-1276.

Pearn, John. 1999. "Improving survival:: A multi-portal approach to improving cardiopulmonary resuscitation outcomes." *Resuscitation* 42 (1):3-9.

Pepe, Paul E, Norman S Abramson, and Charles G Brown. 1994. "ACLS-does it really work?" *Annals of emergency medicine* 23 (5):1037-1041.

Rogers, Rick, John Lombardo, Zigurd Mednieks, and Blake Meike. 2009. *Android application development: Programming with the Google SDK*: O'Reilly Media, Inc.

Sener, S, and S Yaylaci. 2010. "Cardiopulmonary resuscitation and emergency cardiovascular care guideline:'Two guidelines and important changes to daily practice." *Turk J Emerg Med* 10:199-208.

Smith, Andrew F, and Jeremy Wood. 1998. "Can some in-hospital cardio-respiratory arrests be prevented? A prospective survey." *Resuscitation* 37 (3):133-137.

Valenzuela, Terence D, Denise J Roe, Shan Cretin, Daniel W Spaite, and Mary P Larsen. 1997. "Estimating effectiveness of cardiac arrest interventions: a logistic regression survival model." *Circulation* 96 (10):3308-3313.

van Alem, Anouk P, Björn T Sanou, and Rudolph W Koster. 2003. "Interruption of cardiopulmonary resuscitation with the use of the automated external defibrillator in out-of-hospital cardiac arrest." *Annals of emergency medicine* 42 (4):449-457.

Waalewijn, Reinier A, Rien de Vos, Jan GP Tijssen, and Rudolph W Koster. 2001. "Survival models for out-of-hospital cardiopulmonary resuscitation from the perspectives of the bystander, the first responder, and the paramedic." *Resuscitation* 51 (2):113-122.

Waalewijn, Reinier A, Jan GP Tijssen, and Rudolph W Koster. 2001. "Bystander initiated actions in out-of-hospital cardiopulmonary resuscitation: results from the Amsterdam Resuscitation Study (ARRESUST)." *Resuscitation* 50 (3):273-279.

Weil, Max Harry, and Michael Fries. 2005. "In-hospital cardiac arrest." *Critical care medicine* 33 (12):2825-2830.

Wu, Olivia, Andrew Briggs, Tom Kemp, Alastair Gray, Kate MacIntyre, Jack Rowley, and Keith Willett. 2012. "Mobile phone use for contacting emergency services in life-threatening circumstances." *Journal of Emergency Medicine* 42 (3):291-298. e3.

APPENDIX A

APPLICATION SIMPLE CODES

```
package com.example.ozges.emergencycall1;
import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.media.MediaPlayer;
import android.net.Uri;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.telephony.PhoneStateListener;
import android.telephony.TelephonyManager;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.ImageButton;
import android.content.BroadcastReceiver;
import android.content.Context;
@SuppressWarnings("ALL")
public class MainActivity extends AppCompatActivity  {
    boolean call=false;
    ImageButton ara;
    private static final int request = 100;
    MediaPlayer mp = new MediaPlayer();
    Number number = new Number();
// Make call with button
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ara = (ImageButton)findViewById(R.id.ara);
//Permission part
                makeRequest();
                if ( ContextCompat.checkSelfPermission( this,
Manifest.permission.READ_PHONE_STATE ) !=
PackageManager.PERMISSION_GRANTED ) {

                    ActivityCompat.requestPermissions( this, new
String[] {  Manifest.permission.READ_PHONE_STATE
},PackageManager.PERMISSION_GRANTED
                    );
                }
//Starting metronome
       TelephonyManager telephonyManager =
(TelephonyManager)getSystemService(TELEPHONY_SERVICE);
       telephonyManager.listen(new TeleListener(),
PhoneStateListener.LISTEN_CALL_STATE);

}
```

```
//Make call codes
         public void callButtonOnClick (View v)
{
             if ( ContextCompat.checkSelfPermission( this,
Manifest.permission.CALL_PHONE ) !=
PackageManager.PERMISSION_GRANTED ) {
             ActivityCompat.requestPermissions( this, new String[]
{  Manifest.permission.CALL_PHONE },1
             );
          }
         Intent intent = new Intent(Intent.ACTION_CALL);
             intent.setData( Uri.parse(number.getNumber() ));

 if(mp.isPlaying()) {
     mp.stop();
     mp= new MediaPlayer();
 }
call=true;
       this.startActivity(intent);
    }
    class TeleListener extends PhoneStateListener {
        public void onCallStateChanged(int state, String
incomingNumber) {
            super.onCallStateChanged(state, incomingNumber);
            if(call)
            switch (state) {
                    case TelephonyManager.CALL_STATE_IDLE:
                        mp =
MediaPlayer.create(getApplicationContext(), R.raw.metronome);
                        mp.start();
                        break;
            }
        }
    }
private void makeRequest() {
    if (android.os.Build.VERSION.SDK_INT >=
android.os.Build.VERSION_CODES.GINGERBREAD) {
        if (ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.CALL_PHONE)
                != PackageManager.PERMISSION_GRANTED)
        {
            if
(ActivityCompat.shouldShowRequestPermissionRationale(MainActivity.t
his,Manifest.permission.CALL_PHONE)) {

            } else {

ActivityCompat.requestPermissions(MainActivity.this,new
String[]{Manifest.permission.CALL_PHONE}, request);
            }
        }
    }
}
```

```
    public class InComingBroadcastReceiver extends
BroadcastReceiver{
        @Override
        public void onReceive(Context context,Intent intent){

        }
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate( R.menu.menu_emergy, menu);
        return true;
    }


    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.telChange) {
            Intent i = new Intent(MainActivity.this, Degistir.class
);
            startActivity( i );
            return true;
        }
        if (id == R.id.search_Ble)
        {
            Intent j = new Intent(MainActivity.this, BLE.class);
            startActivity( j );
            return true ;
        }

        return super.onOptionsItemSelected(item);
    }

}
```

activity_main.xml

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.ozges.emergencycall1.MainActivity">
    <ImageButton
        android:id="@+id/ara"
        android:layout_width="120dp"
        android:layout_height="135dp"
        android:layout_alignParentEnd="false"
        android:layout_alignParentRight="false"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:background="#313cb4"
        android:contentDescription="@string/app_name"
        android:onClick="callButtonOnClick"
        android:src="@android:drawable/ic_menu_call"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"

android:text="@string/tepecik_egitim_ve_arastirma_hastanesi"
        android:id="@+id/textView"
        android:textSize="18sp"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="@string/blue_code_app"
        android:id="@+id/textView2"
        android:layout_below="@+id/ara"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="73dp"
        android:textColor="#090808"
        android:textSize="18sp"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="@string/Mavi_Kod"
        android:id="@+id/textView3"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:textSize="18sp"
        android:textColor="#060606"/>
</RelativeLayout>
```

Degistir.java

```
package com.example.ozges.emergencycall1;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
public class Degistir extends AppCompatActivity {
    private EditText pass, newtel;
    private Button bpass, btel;
    private static String sifre = "12345";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_degistir);
        pass = findViewById(R.id.pass);
        newtel = findViewById(R.id.newtel);
        bpass = findViewById(R.id.doğrula);
        btel = findViewById(R.id.yeniNum);
        bpass.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
                if (pass.getText().toString().equals(sifre)) {
                    Toast.makeText(Degistir.this, "Şifre doğru",
Toast.LENGTH_SHORT).show();
                    bpass.setVisibility( View.GONE );
                    pass.setVisibility( View.GONE );

                    btel.setVisibility( View.VISIBLE );
                    newtel.setVisibility( View.VISIBLE );

                    btel.setOnClickListener(new
View.OnClickListener() {
 @Override public void onClick(View v) {

Number.setNumber("tel:" + newtel.getText().toString());
Toast.makeText(Degistir.this, "Numara Değiştirildi",
Toast.LENGTH_SHORT).show();
                                Intent i = new Intent(Degistir.this,
MainActivity.class);
 startActivity(i);
                            }
                    });
                }

            }
        }

     );

    }
}
```

activity_degistir.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:orientation="vertical"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.ozges.emergencycall1.Degistir">

    <EditText
        android:id="@+id/pass"
        android:hint="@string/sifreyi_girin"
        android:inputType="numberPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <Button
        android:id="@+id/doğrula"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/sifreyi_dogrula"/>

    <EditText
        android:id="@+id/newtel"
        android:hint="@string/yeni_numarayi_gir"
        android:visibility="gone"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"/>
    <Button
        android:id="@+id/yeniNum"
        android:text="@string/yeni_numarayi_dogrula"
        android:visibility="gone"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</RelativeLayout>
```

BLE.java

```
package com.example.ozges.emergencycall1;

import android.Manifest;
import android.annotation.TargetApi;
import android.app.AlertDialog;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothGatt;
import android.bluetooth.BluetoothGattCallback;
import android.bluetooth.BluetoothGattCharacteristic;
import android.bluetooth.BluetoothGattService;
import android.bluetooth.BluetoothManager;
import android.bluetooth.le.BluetoothLeScanner;
import android.bluetooth.le.ScanCallback;
import android.bluetooth.le.ScanResult;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.support.annotation.NonNull;
import android.support.annotation.RequiresApi;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.telephony.SmsManager;
import android.text.method.ScrollingMovementMethod;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.appindexing.Action;
import com.google.android.gms.appindexing.AppIndex;
import com.google.android.gms.common.api.GoogleApiClient;
import java.util.ArrayList;
import java.util.Collections;
import java.util.HashMap;
import java.util.List;
import static android.net.Uri.parse;



public class BLE extends AppCompatActivity {
    BluetoothManager btManager;
    BluetoothAdapter btAdapter;
    BluetoothLeScanner btScanner;
    Button startScanningButton;
    Button stopScanningButton;
    TextView peripheralTextView;
```

```
Number number = new Number();
Uri phoneNo =  ( Uri.parse( number.getNumber() ));
private final static int REQUEST_ENABLE_BT = 1;
private static final int PERMISSION_REQUEST_COARSE_LOCATION = 1;
Boolean btScanning = false;
int deviceIndex = 0;
ArrayList<BluetoothDevice> devicesDiscovered = new ArrayList<>();
EditText deviceIndexInput;
BluetoothGatt bluetoothGatt;
int rssidegeri = 0;
ArrayList <Integer> rssDegerleri = new ArrayList<Integer>();
ArrayList<ArrayList<Integer>> rssiIsimleriveDegerleri = new
ArrayList<ArrayList<Integer>>();
List<BeaconDevices> beaconDevicess = new
ArrayList<BeaconDevices>();
String closestBeacon;

public final static String ACTION_GATT_CONNECTED =
        "com.example.bluetooth.le.ACTION_GATT_CONNECTED";
public final static String ACTION_GATT_DISCONNECTED =
        "com.example.bluetooth.le.ACTION_GATT_DISCONNECTED";
public final static String ACTION_GATT_SERVICES_DISCOVERED =
        "com.example.bluetooth.le.ACTION_GATT_SERVICES_DISCOVERED";
public final static String ACTION_DATA_AVAILABLE =
        "com.example.bluetooth.le.ACTION_DATA_AVAILABLE";
public final static String EXTRA_DATA =
        "com.example.bluetooth.le.EXTRA_DATA";
@RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
// Stops scanning after 5 seconds.
private Handler mHandler = new Handler();
private static final long SCAN_PERIOD = 50000;
private GoogleApiClient client;
@TargetApi(Build.VERSION_CODES.M)
@RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate( savedInstanceState );
    setContentView( R.layout.activity_ble );
    peripheralTextView = findViewById( R.id.PeripheralTextView );
    peripheralTextView.setMovementMethod( new
ScrollingMovementMethod() );
startScanningButton = findViewById( R.id.StartScanButton );
startScanningButton.setOnClickListener( new View.OnClickListener()
{
    public void onClick(View v) {
        startScanning();
    }
} );
stopScanningButton = findViewById( R.id.StopScanButton );
stopScanningButton.setOnClickListener( new View.OnClickListener() {
    public void onClick(View v) {
        stopScanning();
    }
} );
stopScanningButton.setVisibility( View.INVISIBLE );
```

```
btManager = (BluetoothManager) getSystemService(
Context.BLUETOOTH_SERVICE );
    assert btManager != null;
    btAdapter = btManager.getAdapter();
    btScanner = btAdapter.getBluetoothLeScanner();
    if (btAdapter != null && !btAdapter.isEnabled()) {
        Intent enableIntent = new Intent(
BluetoothAdapter.ACTION_REQUEST_ENABLE );
        startActivityForResult( enableIntent, REQUEST_ENABLE_BT );
    }
    // Make sure we have access coarse location enabled, if not,
prompt the user to enable it
    if (this.checkSelfPermission(
Manifest.permission.ACCESS_COARSE_LOCATION ) !=
PackageManager.PERMISSION_GRANTED) {
        final AlertDialog.Builder builder = new
AlertDialog.Builder( this );
        builder.setTitle( "This app needs location access" );
        builder.setMessage( "Please grant location access so this
app can detect peripherals." );
        builder.setPositiveButton( android.R.string.ok, null );
        builder.setOnDismissListener( new
DialogInterface.OnDismissListener() {
            @TargetApi(Build.VERSION_CODES.M)
            @RequiresApi(api = Build.VERSION_CODES.M)
@Override public void onDismiss(DialogInterface dialog) {
requestPermissions( new
String[]{Manifest.permission.ACCESS_COARSE_LOCATION},
PERMISSION_REQUEST_COARSE_LOCATION );
            }
      } );
     builder.show();
    }
client = new
GoogleApiClient.Builder(this).addApi(AppIndex.API).build();
}

public void sendMessage() {

    try {String textMessage = beaconDevicess.get(0).toString();
        SmsManager smsManager = SmsManager.getDefault();
        ArrayList<String> msgArray =
smsManager.divideMessage(textMessage);

smsManager.sendMultipartTextMessage(String.valueOf(phoneNo),
null,msgArray, null, null);
        Toast.makeText(getApplicationContext(), "Message
Sent",Toast.LENGTH_LONG).show();
    } catch (Exception ex) {

        Toast.makeText(getApplicationContext(), ex.getMessage(),
Toast.LENGTH_LONG).show();
        ex.printStackTrace();
  }
}
```

```
private void stopScanning() {
    Toast.makeText( getApplicationContext(),
beaconDevicess.get(0).toString() , Toast.LENGTH_LONG ).show();
    System.out.println( "stopping scanning" );
    peripheralTextView.append( "Stopped Scanning\n" );
    btScanning = false;
    startScanningButton.setVisibility( View.VISIBLE );
    stopScanningButton.setVisibility( View.INVISIBLE );
    AsyncTask.execute( new Runnable() {
        @RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
        @Override
        public void run() {
            btScanner.stopScan( leScanCallback );
        }
    } );
    sendMessage();
    beaconDevicess.clear();
}
@RequiresApi(api = Build.VERSION_CODES.JELLY_BEAN_MR2)
private void disconnectDeviceSelected() {
    peripheralTextView.append( "Disconnecting from device\n" );
    bluetoothGatt.disconnect();
}

@RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
private void startScanning() {
    ActivityCompat.requestPermissions(this,new
String[]{Manifest.permission.SEND_SMS},1);
    System.out.println( "start scanning" );
    btScanning = true;

    deviceIndex = 0;
    devicesDiscovered.clear();
    peripheralTextView.setText( "" );
    peripheralTextView.append( "Started Scanning\n" );
    startScanningButton.setVisibility( View.INVISIBLE );
    stopScanningButton.setVisibility( View.VISIBLE );
    AsyncTask.execute( new Runnable() {
        @RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
        @Override
        public void run() {
            btScanner.startScan( leScanCallback );
        }

    } );

    mHandler.postDelayed( new Runnable() {
        @Override
        public void run() {
            stopScanning();
        }
    }, SCAN_PERIOD );
}
```

```
@RequiresApi(api = Build.VERSION_CODES.JELLY_BEAN_MR2)
private void connectToDeviceSelected() { peripheralTextView.append(
"Trying to connect to device at index: " +
deviceIndexInput.getText() + "\n" );
int deviceSelected = Integer.parseInt(
deviceIndexInput.getText().toString() );
    bluetoothGatt = devicesDiscovered.get( deviceSelected
).connectGatt( this, false, btleGattCallback );
}
// Device scan callback.
private ScanCallback leScanCallback = new ScanCallback() {
    @RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
    @Override
    public void onScanResult(int callbackType, ScanResult result) {

        peripheralTextView.append( "Device Name:  " +
result.getDevice().getName() + " rssi: " + result.getRssi() + "\n"
);
if (result.getDevice().getName() != null){
            beaconDevicess.add(new
BeaconDevices(result.getDevice().getName(), result.getRssi()));
            Collections.sort(beaconDevicess);
}
        final int scrollAmount =
peripheralTextView.getLayout().getLineTop(
peripheralTextView.getLineCount() ) -
peripheralTextView.getHeight();
        // if there is no need to scroll, scrollAmount will be <=0
        if (scrollAmount > 0)
            peripheralTextView.scrollTo( 0, scrollAmount );
    }
};
@RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
@Override
public void onRequestPermissionsResult(int requestCode,
@NonNull String permissions[], @NonNull int[] grantResults) {
switch (requestCode) {
   case PERMISSION_REQUEST_COARSE_LOCATION: {
    if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                System.out.println( "coarse location permission
granted" );
            } else {
  final AlertDialog.Builder builder = new AlertDialog.Builder( this
);
builder.setTitle( "Functionality limited" );
builder.setMessage( "Since location access has not been granted,
this app will not be able to discover beacons when in the
background." );
builder.setPositiveButton( android.R.string.ok, null );
builder.setOnDismissListener( new
DialogInterface.OnDismissListener() {
@Override
public void onDismiss(DialogInterface dialog) {
                }
                 } ); builder.show();
            }
        }
    }
}
```

```
@RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
private final BluetoothGattCallback btleGattCallback = new
BluetoothGattCallback() {

};


@RequiresApi(api = Build.VERSION_CODES.JELLY_BEAN_MR2)
private void displayGattServices(List<BluetoothGattService>
gattServices) {
    if (gattServices == null) return;

    // Loops through available GATT Services.
    for (BluetoothGattService gattService : gattServices) {

        final String uuid = gattService.getUuid().toString();
        System.out.println( "Service discovered: " + uuid );
        BLE.this.runOnUiThread( new Runnable() {
            public void run() {
                peripheralTextView.append( "Service disovered: " +
uuid + "\n" );
            }
        } );

        new ArrayList<HashMap<String, String>>();

        List<BluetoothGattCharacteristic> gattCharacteristics =
                gattService.getCharacteristics();

        // Loops through available Characteristics.
        for (BluetoothGattCharacteristic gattCharacteristic :
                gattCharacteristics) {

            final String charUuid =
gattCharacteristic.getUuid().toString();
            System.out.println( "Characteristic discovered for
service: " + charUuid );
            BLE.this.runOnUiThread( new Runnable() {
                public void run() {
                    peripheralTextView.append( "Characteristic
discovered for service: " + charUuid + "\n" ); }} );

        }
    }
}
@Override
    public void onStart() {
        super.onStart();

        client.connect();
        Action viewAction = Action.newAction(
                // TODO: choose an action type.
                Action.TYPE_VIEW,
                /*
```

```
TODO: Define a title for the content shown.
TODO: If you have web page content that matches this app activity's
content,
    make sure this auto-generated web page URL is correct.
    Otherwise, set the URL to null.

                */
                "Main Page",
                parse("http://host/path"),
                // TODO: Make sure this auto-generated app URL is
correct.
                parse("android-
app://com.example.ozges.emergencycall1/http/host/path")
        );
        AppIndex.AppIndexApi.start(client, viewAction);
    }
    @Override
    public void onStop() {
        super.onStop();

        Action viewAction = Action.newAction(
                //
                //
                //TODO: choose an action type.
                //TODO: Define a title for the content shown.
                //
                Action.TYPE_VIEW,

                "Main Page",
                //
                //
                //TODO: If you have web page content that matches
this app activity's content,
                //make sure this auto-generated web page URL is
correct.
                //Otherwise, set the URL to null.
                //
                //
                parse("http://host/path"),
                /* TODO: Make sure this auto-generated app URL is
correct. */
                parse("android-
app://com.example.ozges.emergencycall1/http/host/path")
        );
        AppIndex.AppIndexApi.end(client, viewAction);
        client.disconnect();
    }
}
```

activity_ble.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
                xmlns:tools="http://schemas.android.com/tools"
                android:layout_width="match_parent"
                android:layout_height="match_parent"

android:paddingBottom="@dimen/activity_vertical_margin"

android:paddingLeft="@dimen/activity_horizontal_margin"

android:paddingRight="@dimen/activity_horizontal_margin"

android:paddingTop="@dimen/activity_vertical_margin"

tools:context="com.example.ozges.emergencycall1.BLE">

    <Button
        android:layout_width="100dp"
        android:layout_height="50dp"
        android:text="@string/scan"
        android:id="@+id/StartScanButton"
        android:layout_marginTop="40dp"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="50dp"
        android:text="@string/stop_scanning"
        android:id="@+id/StopScanButton"
        android:layout_marginTop="40dp"/>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="100dp"
        android:id="@+id/PeripheralTextView"
        android:scrollbars="vertical"
        android:maxLines="30" />

</RelativeLayout>
```

BeaconDevices.java

```
package com.example.ozges.emergencycall1;

public class BeaconDevices implements Comparable<BeaconDevices> {

    private String deviceName;

    public int getRssiDegeri() {
        return rssiDegeri;
    }

    public void setRssiDegeri(int rssiDegeri) {
        this.rssiDegeri = rssiDegeri;
    }

    private int rssiDegeri;

    public String getDeviceName() {
        return deviceName;
    }

    public void setDeviceName(String deviceName) {
        this.deviceName = deviceName;
    }

    public BeaconDevices (String deviceName, int rssiDegeri) {
        this.deviceName = deviceName;
        this.rssiDegeri = rssiDegeri;

    }
    @Override
    public int compareTo(BeaconDevices beaconDevice) {
        return beaconDevice.rssiDegeri - this.rssiDegeri;
    }

    public String toString() { return String.format("%s, %d",
deviceName, rssiDegeri);
    }
}
```

Number.java

```
package com.example.ozges.emergencycall1;



public class Number {
    public static String nullNumber;


    private static String Number="tel:05325138125";

    public static String getNumber() {

        return Number;
    }

    public static void setNumber(String number) {
        Number = number;
    }
}
```

menu_emergy.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="ozge.sevin.keskin">

    <item
        android:id="@+id/telChange"
        android:icon="@drawable/ic_assignment_ind_black_24dp"
        android:orderInCategory="200"
        android:title="@string/aramayi_degistir"
        app:showAsAction="always"/>

    <item
        android:id="@+id/search_Ble"
        android:checkable="true"
        android:enabled="true"
        android:icon="@drawable/bluetooth_icon"
        android:orderInCategory="200"
        android:title="@string/scanble"
        android:visible="true"
        app:showAsAction="ifRoom"/>

</menu>
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
          xmlns:tools="http://schemas.android.com/tools"
          package="com.example.ozges.emergencycall1" >

    <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission
android:name="android.permission.READ_PHONE_STATE"/>
    <uses-permission
android:name="android.permission.BLUETOOTH_ADMIN"/>
    <uses-permission android:name="android.permission.BLUETOOTH"/>
    <uses-permission android:name="android.permission.CALL_PHONE"/>
    <uses-permission android:name="android.permission.SEND_SMS"/>
    <uses-feature  android:name=" android.hardware.bluetooth_le"
android:required="true"/>


    <application

        android:allowBackup="true"
        android:icon="@mipmap/icon"
        android:label="EmergencyCall1"
        android:supportsRtl="true"
        android:theme="@style/AppTheme"
        tools:ignore="GoogleAppIndexingWarning">
        <activity android:name=".MainActivity" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN"
/>
                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".Degistir">

        </activity>
        <activity android:name=".BLE"/>
        <!-- ATTENTION: This was auto-generated to add Google Play
services to your project for
       App Indexing.  See https://g.co/AppIndexing/AndroidStudio
for more information. -->
        <meta-data
            android:name="com.google.android.gms.version"
            android:value="@integer/google_play_services_version"
/>

    </application>



</manifest>
```