

FEN BİLİMLERİ ENSTİTÜSÜ
BİLİŞİM SİSTEMLERİ ANABİLİM DALI



ANDROID İŞLETİM SİSTEMİNDE İLK YARDIM PAKETİ

YÜKSEK LİSANS TEZİ

YILDIRIM ÖZYAKIŞIR

tarafından

YÜKSEK LİSANS

derecesi şartını sağlamak için hazırlanmıştır.

Haziran 2016

Program: Bilişim Sistemleri

ANDROID İŞLETİM SİSTEMİNDE İLK YARDIM PAKETİ

YÜKSEK LİSANS TEZİ

YILDIRIM ÖZYAKIŞIR

tarafından

OKAN ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Bilişim Sistemleri Anabilim Dalına

YÜKSEK LİSANS

derecesi şartını sağlamak için sunulmuştur.

Onaylayan Jüri:

Yrd. Doç. Dr. Nurşen TOPÇUBAŞI
Danışman

Prof. Dr. Ahmet Faik KAŞLI
Üye

Prof. Dr. Ahmet Mesut RAZBONYALI
Üye

Haziran 2016

Program: Bilişim Sistemleri

ÖZET

Bu yüksek lisans tez çalışmasında “Temel İlk Yardım” mobil uygulama süreçleri anlatılmaktadır. Uygulama, ilk yardım gerektiren hallerde başvurulması gereken müdahale adımlarını gösterecek ve temel ilk yardım eğitimi sertifikasını alacak kişiler için yol gösterici olacaktır.

Uygulama 4 başlık altında işlenmiş olup ilk yardım, ek bilgi, eğitim süreci ve test kısımlarından oluşmaktadır. İlk yardım bölümünde; temel ilk yardım müdahale konuları, video ve acil yardım numarasının hızlı arama tuşu bulunmaktadır. Eğitim süreci bölümü; temel ilk yardım eğitimi alacak kişiler için eğitim sürecini, sınav adımlarını ve sertifika sürecini anlatmaktadır. Test bölümünde; 8 adet test uygulaması ve sorularım bölümü bulunmaktadır. Kullanıcı, test sorularım kısmında ve ek bilgi bölümünde; yeni müdahale adımları ve yeni test soruları yazabilmektedir. Ek bilgi ve test sorularım ekranında verilerin tutulması için Android - SQLite database yapısından yararlanılmıştır.

Temel ilk yardım uygulaması akıllı telefonlar arasında yaygın olarak kullanılan android işletim sisteminde geliştirilmiştir. Eclipse ortamında java programlama dili kullanılarak yazılmıştır. Uygulama herhangi bir ücret ödmeden Google Play (android uygulama market) “Temel İlk Yardım” ismi ile aratılarak indirilmektedir.

Anahtar Kelimeler: Temel ilk yardım, Temel ilk yardım eğitim süreci, Temel ilk yardım test, İlk yardım soruları

ABSTRACT

In this postgraduate thesis work "Basic First Aid" will be discussed mobile application processes. Also basic first aid training will be a guide for people who want to get the certificate.

The application is processed under four headings first aid, additional info section, training process and test. In the first aid section; there are basic first aid intervention subjects, a speed dial button and videos and emergency numbers. Training process section; describes training process, examination and certification process stages for people who want to get first aid certificate. In the test section; there are 8 test application and questions section. In the additional info section and test questions section; in the additional info section new intervention steps and new test questions can be created by people who will use the application and it can be added. Was utilised of Android SQLite database structure for to keep the data in the additional info and test questions screen.

Basic first aid application is commonly used between smart phones android operating system has been developed. Written using the java programming language in the Eclipse environment. Application be downloaded with searching "Basic Firstid" without paying any fees to the Google Play (Android app store).

Keywords: Adding new knowledge and questions, Basic first aid, Basic first aid testing, First aid questions

TEŞEKKÜR

Tez çalışmam boyunca gösterdiği her türlü destek ve yardımlarından dolayı çok değerli hocam Yrd. Doç. Dr. Nurşen TOPÇUBAŞI'na en içten dileklerle teşekkür ederim.

Üniversite hayatım boyunca bana her konuda yardımcı olan ve her zaman en büyük destekçim olan eşim Sedef Nihal ÖZYAKIŞIR'a tüm kalbimle teşekkürlerimi sunuyorum.

Tez çalışmam boyunca yardımlarını esirgemeyen çalışma arkadaşlarım Alpaslan Mert KÜÇÜKÖZBEK ve Uğur İÇLİ'ye en içten dileklerle teşekkür ederim.

YILDIRIM ÖZYAKIŞIR

Haziran 2016

İÇİNDEKİLER

ÖZET.....	II
ABSTRACT	III
TEŞEKKÜR.....	IV
İÇİNDEKİLER	V
ŞEKİL LİSTESİ	VIII
KISALTMALAR	XII
I. GİRİŞ	1
II. ANDROID İŞLETİM SİSTEMİNDE YAPILAN AKADEMİK ÇALIŞMALAR... 7	
2.1. Mobil İlaç Prospektüs Uygulaması	7
2.2. Sms Ve Android Tabanlı Uygulamalarla Sınıf İçi Etkileşimin Arttırılması.....	8
2.3. Bir Android Uygulama Modeli: İstanbul Toplu Taşıma Bilgi sistemi	10
2.4. Trip Tracker Application On Android	11
2.5. Android İşletim Sistemi İle Çalışan Mobil Cihazlarda Uygulama Geliştirilmesi. 12	
III. TEMEL İLK YARDIM.....	14
3.1. İlk Yardımcı Kimdir?	14
3.1.1. İlk yardımcının öncelikli amaçları nelerdir?	15
3.1.2. İlk yardımcının müdahalede yapması gerekenler.....	16
3.2. Temel İlk Yardım Sertifikasını Kimler Almalıdır?.....	16
3.3. Temel İlk Yardım Serfikası Ve Sınavı.....	17
IV. MOBİL UYGULAMA	18

V. ANDROID MOBİL UYGULAMA YAZILIM GELİŞTİRME SÜRECİ	20
5.1. Eclipse Yazılım Geliştirme Editörü	20
5.2. Android Sanal Cihaz	22
5.2.1. Android sanal cihaz ayarları	23
5.3. Genymotion Sanal Cihaz.....	25
5.3.1. Genymotion sanal cihaz kurulumu	25
VI. TEMEL İLK YARDIM UYGULAMASI SÜRECİ	28
6.1. Ekran Tasarımı	29
6.2. İlk Yardım	33
6.3. Ek Bilgi	42
6.4. Eğitim Süreci.....	61
6.5. Test.....	63
VII. TEMEL İLK YARDIM UYGULAMASININ YAYINLANMASI.....	88
7.1. Google Play Sayfasında Yer Alan Temel İlk Yardım Uygulaması Açıklama Kısmı	91
7.1.1. İlk yardım bölümü	91
7.1.2. Ek bilgi bölümü	91
7.1.3. Eğitim süreci bölümü	92
7.1.4. Test bölümü	92
VIII. SONUÇ	93
KAYNAKLAR	98
EKLER	100
EK A.....	100
EK B.....	101
EK C.....	102

ÖZGEÇMİŞ 103



ŞEKİL LİSTESİ

Şekil 1.1. Google Play basic first aid arama sonucu	2
Şekil 1.2. Google Play temel ilk yardım arama sonucu	3
Şekil 2.1. Uygulama ilk açılış ekranı.....	8
Şekil 2.2. Sistem bileşenleri.....	9
Şekil 2.3. Transistanbul türkçe ara yüz ekranı	11
Şekil 2.4. Harita sekmesi.....	12
Şekil 2.5. Uygulama açılış ekranı	13
Şekil 3.1. Hayat kurtarma zinciri.....	15
Şekil 4.1. Ağustos 2015 mobil işletim sistemi kullanım oranı.....	18
Şekil 4.2. Temmuz 2015 mobil uygulama sayıları.....	19
Şekil 5.1. Eclipse ekranı.....	20
Şekil 5.2. Android sanal cihaz ayarları	23
Şekil 5.3. Android sanal cihaz.....	25
Şekil 5.4. Genymotion program ekranı	27
Şekil 6.1. Temel ilk yardım uygulama ikon.....	29
Şekil 6.2. Temel ilk yardım ekran tasarımı.....	29
Şekil 6.3. Dört tab başlık ekranı.....	30
Şekil 6.4. Ekran tasarım kodu activity_main.xml	31

Şekil 6.5. Ekran tasarımı MainActivity.java kodu	32
Şekil 6.6. İlk yardım ana ekran	33
Şekil 6.7. İlk yardım ekran tasarımı ve ABC değerlendirilmesi butonu xml kodu.....	36
Şekil 6.8. ABC değerlendirilmesi button kodu	37
Şekil 6.9. Koma pozisyonu bilgi ekranı	37
Şekil 6.10. Koma.xml TextView kodu.....	38
Şekil 6.11. Hızlı arama tuşu	38
Şekil 6.12. Acil numara arama kodu	39
Şekil 6.13. AndroidManifest telefon arama yetkisi	39
Şekil 6.14. İlk yardım video butonu.....	39
Şekil 6.15. Koma pozisyonu video oynatma sayfası.....	40
Şekil 6.16. Video oynatma kodu	41
Şekil 6.17. Ek bilgi sekmesi ve ek bilgi butonu	42
Şekil 6.18. Ek bilgi sekmesi xml kodu.....	43
Şekil 6.19. Ek bilgi ekranı ve yeni ek bilgi ekle butonu	44
Şekil 6.20. Ek bilgi java kodu	47
Şekil 6.21. Ek bilgi kaydet ekranı	47
Şekil 6.22. Ek bilgi kaydet java kodu	50
Şekil 6.23. Belirti adı girilmesi gerektiğini belirten uyarı mesajı	51
Şekil 6.24. Belirti adı boş bırakılmaması java kodu	52

Şekil 6.25. Ek bilgi oku, düzelt ve sil ekranı	52
Şekil 6.26. Ek bilgi oku, düzelt ve sil java kodu.....	56
Şekil 6.27. Ek bilgi sil uyarı mesajı	57
Şekil 6.28. Ek bilgi sil uyarı mesajı java kodu.....	58
Şekil 6.29. Veritabanı java kodu	60
Şekil 6.30. Veritabanıyardimci java kodu	61
Şekil 6.31. Eğitim süreci ekranı	62
Şekil 6.32. Eğitim süreci xml kod.....	63
Şekil 6.33. Test ekranı.....	64
Şekil 6.34. Test soru ve cevap ekranı.....	65
Şekil 6.35. Test soru seçeneği xml kodu.....	66
Şekil 6.36. Cevap java kodu.....	67
Şekil 6.37. “60 dk süreniz dolmuştur. Test süreniz bitmiştir.” mesajı.....	67
Şekil 6.38. Test zamanlayıcı java kodu.....	69
Şekil 6.39. Sorularım ekranı ve yeni soru ekle butonu	70
Şekil 6.40. Sorularım java kodu	73
Şekil 6.41. Soru kaydet ekranı	74
Şekil 6.42. Soru kaydet java kodu.....	76
Şekil 6.43. Soru girmemiz gerektiğini belirten uyarı mesajı.....	77
Şekil 6.44. Soru kaydet - soru veri kontrol java kodu.....	78

Şekil 6.45. Soru oku, düzelt ve sil ekranı.....	78
Şekil 6.46. Soru oku, düzelt ve sil java kodu	82
Şekil 6.47. Soru sil uyarı mesajı.....	82
Şekil 6.48. Soru sil uyarı mesajı java kodu	83
Şekil 6.49. Veritabanisoru java kodu	86
Şekil 6.50. Veritabanisoruyardimci java kodu	87
Şekil 7.1. Google Play Geliştirici Konsolu - Mağaza girişi.....	90
Şekil 7.2. Google Play Temel İlk Yardım uygulaması	90
Şekil 8.1. Google Play Temel İlk Yardım uygulaması	94
Şekil 8.2. Temel ilk yardım uygulaması kullanıcı oyları	94
Şekil 8.3. Temel ilk yardım uygulaması kullanıcı yorumları.....	95
Şekil 8.4. Temel ilk yardım uygulaması android sürümü yüklenme oranları	96
Şekil 8.5. Temel ilk yardım uygulaması cep telefonu marka model yüklenme oranları	97

KISALTMALAR

ABC: Airway Breathing Circulation (havayolu nefeslendirme dolaşım)

GSM: Global System for Mobile Communications (Mobil İletişim İçin Küresel Sistem)

SMS: Short Message Service (Kısa Mesaj Hizmeti)

SDK: Android Software Development Kit (Android Yazılım Geliştirme Kiti)

İBB: İstanbul Büyükşehir Belediyesi

ADT: Android Development Tools (Android uygulama geliştirme araçları Eclipse eklentisi)

JDK: Java Development Kit (Java Geliştirme Kiti)

PHP: Hypertext Preprocessor (Üstünyazı Önışlemcisi)

RSS: Rich Site Summary (Zengin Site Özeti)

XML: Extensible Markup Language (Genişletilebilir İşaretleme Dili)

AVD: Android Virtual Device (Android Sanal Aygıt)

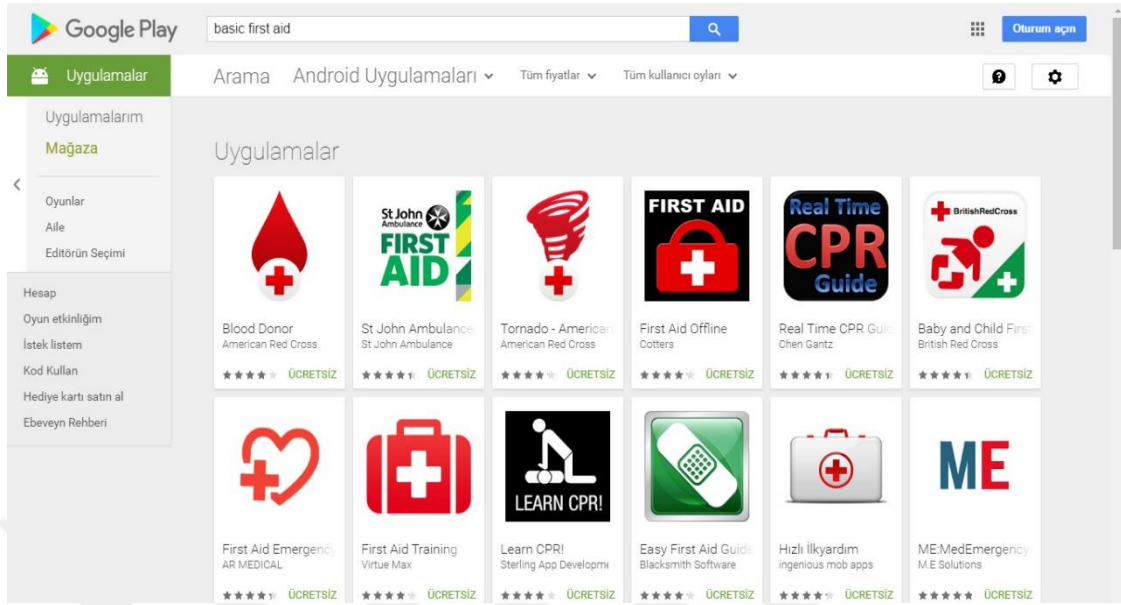
3GP: 3GPP file format (üçüncü nesil mobil telefonlar için geliştirilmiş çoklu ortam container formatıdır)

I. GİRİŞ

İnsanların teknolojiye olan ihtiyacı ve ilgisi gün geçtikçe artmaktadır. Hayatımızın her alanında akıllı cihazlar yer almaktadır. Günümüzde, gerek işlem hızı ve gerekse kullanım kolaylığından dolayı bir çok işlemler artık akıllı cep telefonları ile yapılmaktadır. Öyle ki, akıllı telefonlar sayesinde insanlar bilgisayarlarda yapabilecekleri çoğu işlemleri zaman ve mekandan bağımsız olarak her zaman yanında kolay taşıyabildikleri cihazlarda yapmaktadırlar. Teknolojinin gelişmesiyle telefonlar sadece iletişim aracı olarak kullanımdan çıkıp, günlük yaşantıya destek olacak şekilde hayatımıza girmiştir.

Günümüzde bilgiye en hızlı ve en kolay ulaşılan yöntemlerden biri olarak akıllı telefonlar kullanıldığı düşünüldüğünde bu tezde 1.4 milyar aktif android telefon kullanıcısı (ÜSTÜNŞOY, 2015) göz önüne alınarak android işletim sistemi kullanan kişiler için hayati müdahale adımlarını anlatan temel ilk yardım uygulaması yapılmıştır.

Google Play (Android market) sayfasında ingilizce ve türkçe içerikli temel ilk yardım uygulamaları yer almaktadır. Google Play sayfasında “basic first aid” yazılarak arama yapıldığında Şekil 1.1’de yer alan uygulamalar gelmektedir.

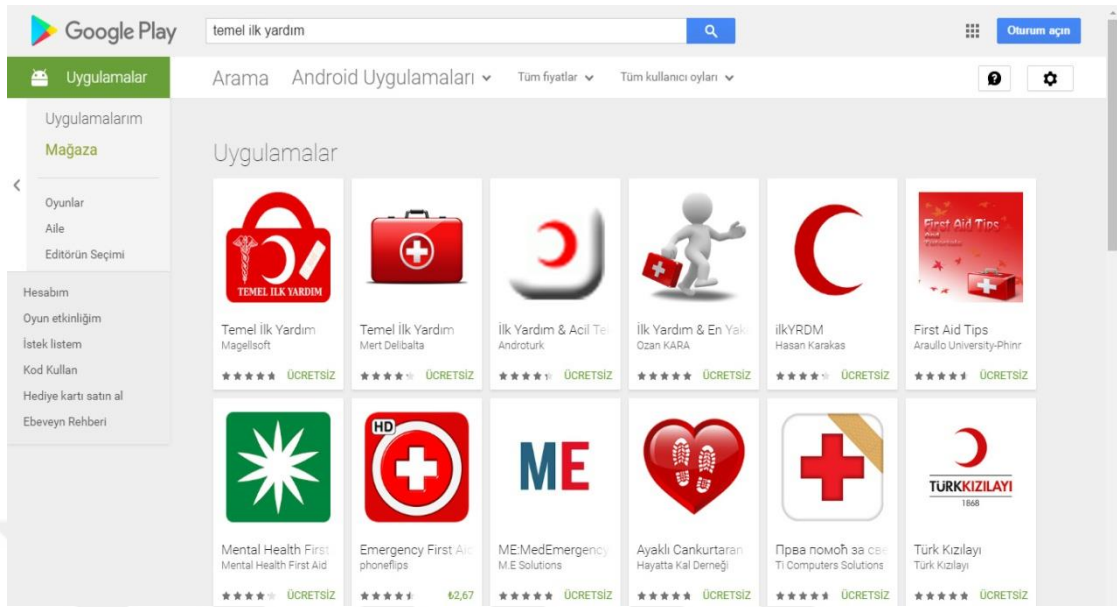


Şekil 1.1. Google Play basic first aid arama sonucu

Uygulamaların ortak özellikleri incelendiğinde;

- Müdahale adımları,
- Slayt adımları ile hazırlanmış sesli müdahale adımları,
- Video,
- Doğal afet,
- Test,
- Hastane lokasyon bilgilerini barındıran uygulamalar yer almaktadır.

Google Play sayfasında “temel ilk yardım” yazılarak arama yapıldığında Şekil 1.2’de yer alan uygulamalar gelmektedir.



Şekil 1.2. Google Play temel ilk yardım arama sonucu

Uygulamaların ortak özellikleri incelendiğinde;

- Müdahale adımları ve resimler,
- Acil telefon numaraları,
- Ülkelerin acil telefon numaraları,
- Köpek ve kedi müdahale adımlarını barındıran uygulamalar yer almaktadır.

Bu tezde oluşturulan temel ilk yardım uygulamasında;

- İlk yardım
 - Müdahale adımları
 - Acil telefon numaraları
 - İlk yardım videoları
- Ek bilgi
- Eğitim süreci

- Test
 - 8 adet test
 - Sorularım

kısımlarından oluşmaktadır.

Türkçe yazılmış ilk yardım uygulamalarında, bu tez kapsamında geliştirilen temel ilk yardım uygulamasını öne çıkaran;

- Önemli müdahale adımlarını anlatan videolar,
- Acil telefon numaraların doğrudan aranması,
- Ek bilgi bölümünde kullanıcılar tarafından yeni müdahale adımları yazılabilmesi,
- İlk yardım sertifikası alma adımlarının ayrıntılı anlatılması,
- Uygulamada yer alan 8 adet test uygulaması ve sorularım bölümü butonuna basılarak kullanıcılar tarafından yeni sorular ve cevaplar yazılabilmesi uygulaması ön plana çıkartmaktadır.

Test – sorularım ve ek bilgi bölümünde oluşturulan bilgiler uygulamanın içinde yer alan veritabanına yazılarak sadece uygulamayı indiren kullanıcının telefonunda saklanmaktadır.

Uygulama Google Play sayfasına 20 Aralık 2015 tarihinde yüklenmiştir. “Temel İlk Yardım” olarak aratıldığında 1. sırada yer almaktadır.

Bu tez’deki amaç; insanlar için arı sokması, burkulmadan başlayarak herhangi bir kaza ya da yaşamı tehlikeye düşüren bir durumda sağlık ekiplerinin olay yerine ulaşmaya dek insan hayatının kurtarılması veya kötüye gitmesini engellemek için

geliştirilen temel ilk yardım uygulamasının yazılmasıdır. Türkçe kaynaklı detaylı olarak hazırlanmış ve günümüzde bilgiye en hızlı erişim sağlayan akıllı telefonların işletim sistemlerinden biri olan android işletim sisteminde çalışan bir uygulamadır.

Android işletim sisteminde çalışan çeşitli alanlarda geliştirilen tıbbi, seyahat, toplu ulaşım vb. gibi bir çok mobil uygulamalar geliştirilmiştir.

2. bölümde android işletim sisteminde yapılan uygulamalar ve bunlara ilişkin akademik çalışmalar mobil ilaç prospektüs uygulaması, sms ve android tabanlı uygulamalarla sınıf içi etkileşimin artırılması, bir android uygulama modeli: istanbul toplu taşıma bilgi sistemi, trip tracker application on android, android işletim sistemi ile çalışan mobil cihazlarda uygulama geliştirilmesi ele alınmaktadır.

3. bölümde temel ilk yardım uygulaması hazırlanmasına yönelik olarak temel ilk yardım nedir? İlk yardımcı kimdir? İlk yardımın öncelikli amaçları nelerdir? Temel ilk yardım sertifikasını kimler almalıdır? Temel ilk yardım sertifikası ve sınavı hakkında bilgiler yer almaktadır.

4. bölümde akıllı telefonların insan hayatındaki yeri ve önemi incelenmiştir. Mobil işletim sistemi olarak Android, iOS, Windows Phone ve BlackBerry OS mobil işletim sistemi kullanım oranı ve mobil uygulama hakkında bilgiler yer almaktadır.

5. bölümde android işletim sisteminde çalışacak temel ilk yardım uygulamasının geliştirilmesi için Eclipse ortamının hazırlanması ve sanal cihaz kurulumu anlatılmaktadır.

6. bölümde temel ilk yardım uygulama süreci anlatılmaktadır. Uygulamayı geliştirmeden önce Temel İlk yardımcı sertifikası ve Temel İlk yardımcı kimliği

alınmıştır. Uygulamanın logo tasarımı, ekran tasarımı, içeriği ve kodları anlatılmaktadır.

7. bölümde Google Play Geliştirici Konsolu üye adımları ve Temel ilk yardım mobil uygulamasının yayınlanması anlatılmaktadır.

8. sonuç bölümünde Temel ilk yardım uygulamasını indiren kullanıcıların sayıları, oyları, yorumları ve android işletim sisteminde kurulan cihazların bilgileri yer almaktadır.



II. ANDROID İŞLETİM SİSTEMİNDE YAPILAN AKADEMİK ÇALIŞMALAR

İkinci bölümde android işletim sisteminde geliştirilen akademik çalışmalar anlatılmaktadır. Yurt içi ve yurt dışı akademik tez çalışmalarında geliştirilen;

- Mobil ilaç prospektüs uygulaması,
- Sms ve android tabanlı uygulamalarla sınıf içi etkileşimin artırılması,
- Bir android uygulama modeli: istanbul toplu taşıma bilgi sistemi,
- Trip tracker application on android,
- Android işletim sistemi ile çalışan mobil cihazlarda uygulama geliştirilmesi,

akademik tezleri incelenmiştir.

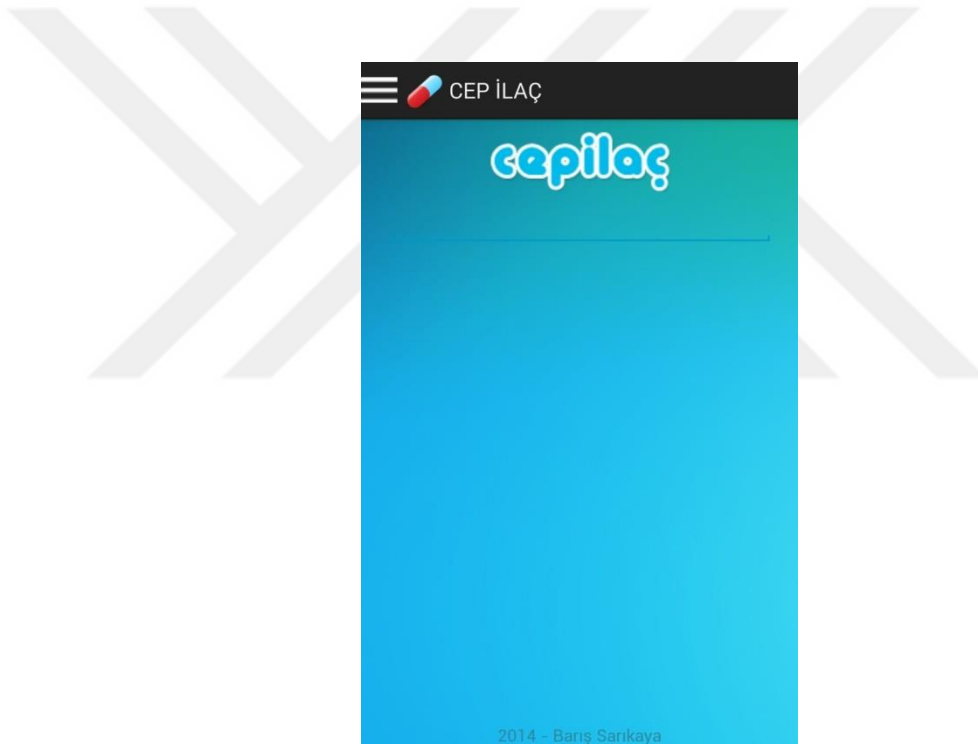
2.1. Mobil İlaç Prospektüs Uygulaması

İstanbul Teknik Üniversitesi - Bilişim Enstitüsü - Bilgisayar Bilimleri Anabilim Dalı - Bilgisayar Bilimleri Programı - Yüksek Lisans Tezi - Barış SARIKAYA'nın Aralık 2013 tarihinde sunmuş olduğu Mobil ilaç prospektüs uygulaması tezi ve Google Play sayfasındaki Cep İlaç uygulaması indirilerek incelenmiştir (SARIKAYA, Mobil ilaç prospektüs uygulaması, 2013)

Mobil ilaç prospektüs uygulama tezinin amacı; *“İlaçların içerisinde bulunan yazılı prospektüs bilgilerinin okunamaması ve özellikle evlerde ilaçların kutularının ve prospektüslerinin zamanla kaybolmasıdır”* (SARIKAYA, Mobil ilaç prospektüs uygulaması, 2013). İlaç kutularının ve yazılı prospektüs bilgilerinin kaybolması

sonucu günlük hayatta insanların kolay bir şekilde okuyabileceğini ilaç bilgilerinin yer aldığı bir uygulama geliştirilmiştir.

Mobil ilaç prospektüs uygulaması Google Play (Android market) sayfasında Cep İlaç ismi ile ücretsiz olarak yer almaktadır. Uygulama en son 8 Eylül 2015 tarihinde güncellenmiştir. Gereken Android sürümü ve boyut cihaza göre değişir olarak yer almaktadır (SARIKAYA, Cep ilaç, 2015). Şekil 2.1’de cep ilaç uygulama açılış ekranı gösterilmektedir.



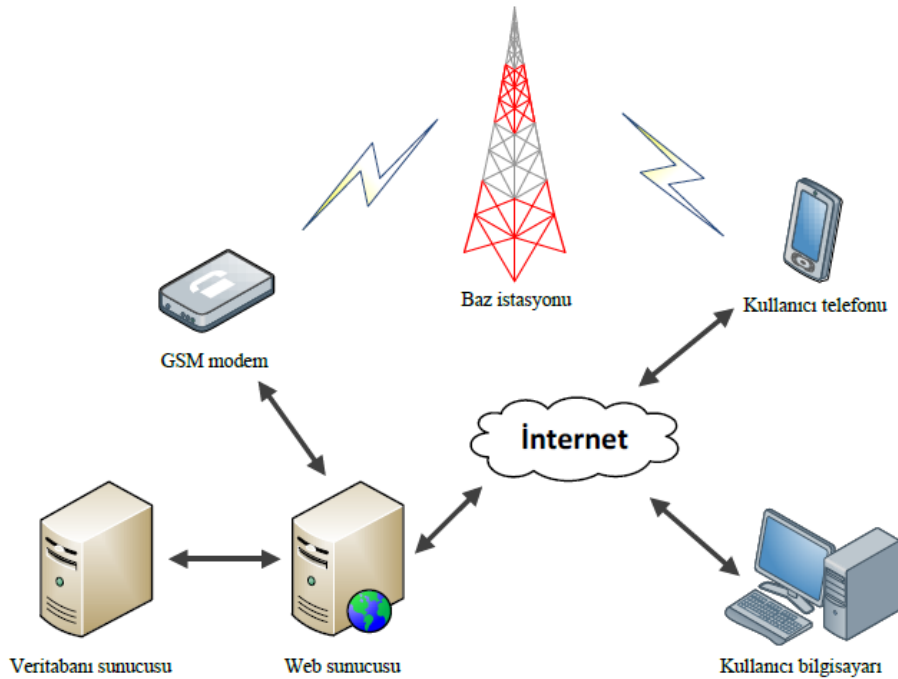
Şekil 2.1. Uygulama ilk açılış ekranı

2.2. Sms Ve Android Tabanlı Uygulamalarla Sınıf İçi Etkileşimin Arttırılması

Bilecik Üniversitesi - Fen Bilimleri Enstitüsü - Bilgisayar Mühendisliği Anabilim Dalı
- Yüksek Lisans Tezi - Yusuf MUŞTU ‘nun 2012 yılında sunmuş olduğu Sms ve

Android tabanlı uygulamalarla sınıf içi etkileşimin artırılması tezi incelenmiştir (MUŞTU, 2012).

Bu tez çalışmasında sınıf içi etkileşimini artırması için öğrenci ve eğitimci'nin sms, web arabirimi ve android uygulması kullanılarak hızlı ve verimli bir biçimde iletişim kurabilmelerini sağlayan bir sistem geliştirilmiştir. Üç sistemin de birbiri ile entegrasyonu yapılmış ve kullanımının kolay olması sağlanmıştır. Uygulamada kullanılan veriler, ortak kullanılan veri tabanı sunucusunda tutulmuştur. Uygulama Google play sayfasında yer almadığından dolayı test edilememiştir. *“Sistem genel olarak öğrenci bilgilerinin bulunduğu bir veritabanı, kullanıcı etkileşimini kontrol edip kullanıcıları yönlendiren servislerin bulunduğu uygulama sunucusu, GSM üzerinden haberleşmenin sağlandığı bir GSM modemi ve kullanıcılardan oluşmaktadır”* (MUŞTU, 2012). Şekil 2.2’de sistem bileşenlerinin haberleşme işlemi gösterilmektedir (MUŞTU, 2012).



Şekil 2.2. Sistem bileşenleri

2.3. Bir Android Uygulama Modeli: İstanbul Toplu Taşıma Bilgi sistemi

İstanbul Üniversitesi - Fen Bilimleri Enstitüsü - Enformatik Anabilim Dalı - Yüksek Lisans Tezi - Mustafa ÖZCAN 'nın Mart 2013 tarihinde sunmuş olduğu Bir Android Uygulama Modeli: İstanbul Toplu Taşıma Bilgi Sistemi tezi incelenmiştir (ÖZCAN, 2013).

Bu tezin amacı; insanların çoğu zaman yanlarında taşıdığı akıllı mobil cihazlardan toplu ulaşım bilgilerine tek kaynaktan erişebileceği bir uygulama geliştirilmiştir. İstanbul'da çok farklı toplu ulaşım alternatifleri olması güncel ve doğru bilgilere daha kolay ve daha hızlı erişim sağlanması düşünülmüştür. "Transistanbul" adlı İstanbul Toplu Taşıma Bilgi Sistemi mobil uygulaması geliştirilmiştir (ÖZCAN, 2013).

Yazılım geliştirme metodolojisi olarak Scrum'ın kullanıldığı belirtilmektedir. Scrum, yazılım mühendisliğinde uygulama geliştirme aşamasında gözlemci, geliştirmeci ve tekrara dayalı olarak tasarlanmış bir yazılım geliştirme süreçleri yönetimidir (Scrum_Nedir?, tarih yok). Scrum süreci planlama aşamasını kolaylaştırmak için şeffaflık, denetleme ve uyarılama'dan oluşan 3 ilkeye önem vermiştir (Sutherland, 2013). Android işletim sisteminde çalışacak uygulama Eclipse ortamında Java programlama dili kullanılarak yazılmıştır. Türkçe ve İngilizce dillerini desteklemektedir.

Uygulama Google Play Store'da yayınlanmak üzere İstanbul Büyükşehir Belediyesi (İBB)'ne teslim edilmiş olduğu belirtilmektedir. Ancak Google play sayfasında Transistanbul isim ile arama yapıldığında uygulamanın bulunmadığı görülmektedir.

Transistanbul uygulamasının türkçe arayüzü Şekil 2.3'de gösterilmektedir (ÖZCAN, 2013).



Şekil 2.3. Transistanbul türkçe ara yüz ekranı

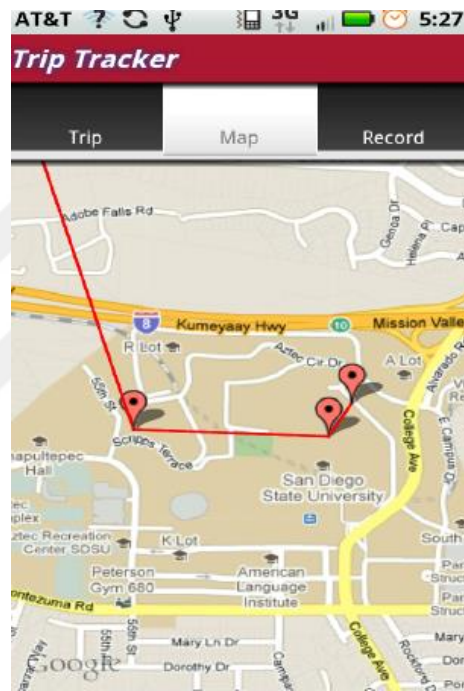
2.4. Trip Tracker Application On Android

San Diego State Üniversitesi – Bilgisayar Bilimleri – Yüksek Lisans Tezi Siddartha Sreenivasa Reddy'nin Yaz 2011 yılında sunmuş olduğu Trip tracker application on android tezi incelenmiştir (Reddy, 2011).

Seyahat, insanlar için monoton hayatını geride bıraktığı, maceracı ruhunu harekete geçiren, günlük streslerden arındıran ve her zaman en iyi vakitlerin geçirildiği bir yöntem olmuştur. Bugünlerde akıllı telefonlar bir çok insanın hayatının ayrılmaz bir parçasıdır. Seyahat takip programı seyahat edenler için coğrafi lokasyona ait etiket konumlarını multimedia özelliği ile elde etmesini sağlar. Bu uygulama; ziyaret yerleri hakkında seyahat notları ve görüntüleri birleştirir ve lokasyon ile ilgili tüm özelliklerin bir harita üzerinden görüntülenmesini sağlar. Bu uygulama tüm bilgilerin kullanıcı

için zengin içeriği ile tek bir yerde toplanması ve kolay erişimli olmasını sağlamıştır (Reddy, 2011).

Android SDK ve Eclipse IDE geliştirme ortamı kullanılarak geliştirilmiştir. Uygulamanın hazırlanması sırasında kullanıcı arayüzü, konum ve harita, görüntü ekleme ve veri depolama özellikleri kullanılmıştır. Harita sekmesi Şekil 2.4'de gösterilmektedir (Reddy, 2011).



Şekil 2.4. Harita sekmesi

2.5. Android İşletim Sistemi İle Çalışan Mobil Cihazlarda Uygulama Geliştirilmesi

Ege Üniversitesi - Fen Bilimleri Enstitüsü - Bilgisayar Mühendisliği Anabilim Dalı – Yüksek Lisans Tezi - Ebru İĞCİ'nin 11.12.2014 tarihinde sunmuş olduğu Android işletim sistemi ile çalışan mobil cihazlarda uygulama geliştirilmesi tezi incelenmiştir (İĞCİ, 2014).

Bu tezin amacı; “MySQL veritabanından okunan veriyi RSS yayınına dönüştüren bir php yazılımı ile, bu veriye RSS aracılığıyla erişip, ayrıştırarak anlamlı veriye dönüştüren ve kullanıcıya sunan mobil bir Android uygulaması geliştirilmiştir” (İĞCİ, 2014).

Ege Mezun uygulama Google play (Android market) sayfasında ücretsiz olarak yer almaktadır. Uygulama en son 12 Ocak 2015 tarihinde güncellenmiştir. 2.2 android sürüm ve üzerinde çalışmaktadır. Uygulama boyutu 2,4 MB olduğu görülmektedir (ÜNİVERSİTESİ, 2015).

Uygulamanın düzgün ve sorunsuz çalışması için Android 4.4. işletim sistemi versiyonun yüklü olması istenmektedir. Ege Üniversitesi mezun olanların bilgilerinin görüntülenebildiği; fakülte, bölüm ve yıl bazlı çalışan bir uygulamadır. Seçilen fakülteye göre bölümler, yıllar ve son olarak mezunlar listelenmektedir. İstenilen kişinin seçimi yapılarak kişinin bilgileri ve fotoğrafı görüntülenebilmektedir. Şekil 2.5’de uygulama ilk açılış ekranı gösterilmektedir.



Şekil 2.5. Uygulama açılış ekranı

III. TEMEL İLK YARDIM

Hayat, insanları çeşitli yaralanma ve hastalıklarla karşılaştırır. İnsanların tıbbi kuruluşlardan uzak bulunduğu zaman ve mekanlarda, basit yada hayatı tehlikeye atan durumlara karşılaştığında müdahale etmek için temel ilk yardım bilgisi gereklidir. Temel ilk yardım insanlar için hayat kurtarıcı bilgi ve yöntemler bütünüdür. Kızılayın web sitesinden alınan tanımı; *“Herhangi bir kaza ya da yaşamı tehlikeye düşüren bir durumda, sağlık görevlilerinin tıbbi yardımı sağlanıncaya kadar, hayatın kurtarılması ya da durumun daha kötüye gitmesini önleyebilmek amacıyla olay yerinde, tıbbi araç gereç aranmaksızın mevcut araç ve gereçlerle yapılan ilaçsız uygulamalardır”* (Temel İlk Yardım Bilgileri, 2015).

Kaza ve yaralanma sonucu kalıcı fiziksel hasar ve ölümlerin bir çoğu olay yerinde ve zamanında yapılamayan veya yanlış yapılan müdahale nedeniyle gerçekleşmektedir. Her bireyin başına hayati kazalar veya yaralanmalar gelebileceğinden, herkesin temel ilk yardım eğitimi alması gereklidir.

3.1. İlk Yardımcı Kimdir?

İlk yardımcı; hayat kurtarmak amacıyla olay yerinde yapılan ilaçsız müdahalelerde bulunan eğitim ve sertifika almış kişidir. Kızılayın web sitesinden alınan tanımı; *“İlk yardımın tanımında belirtilen amaç doğrultusunda, hasta/yaralıya tıbbi araç ve gereç aranmaksızın mevcut araç ve gereçlerle, sağlık ekibi gelinceye kadar, ilaçsız uygulamaları yapan, konuyla ilgili eğitim sonunda ilk yardımcı sertifikası almış kişiyi tanımlar”* (Temel İlk Yardım Bilgileri, 2015).

İlk yardım ile acil yardım tanımları karıştırılmaktadır. Acil yardım; sağlık çalışanları tarafından tıbbi araç ve gereçler ile yapılan müdahale iken, ilk yardım eğitim almış herkez tarafından yapılan ilaçsız müdahaledir.

3.1.1. İlk yardımcının öncelikli amaçları nelerdir?

- Hayati tehlikeyi ortadan kaldırmak,
- Yaşamsal fonksiyonların sürdürülmesini sağlamak,
- Hasta/yaralının durumunun kötüleşmesini engellemek,
- İyileştirmeyi kolaylaştırmak (Hayat Kurtarma Zinciri, 2015).

Hayat kurtarma zinciri Şekil 3.1’de gösterilmektedir (Hayat Kurtarma Zinciri).



Şekil 3.1. Hayat kurtarma zinciri

3.1.2. İlk yardımcının müdahalede yapması gerekenler

- Hasta/yaralının durumunu değerlendirmek (ABC) ve öncelikli müdahale edilecekleri belirlemek,
- Hasta/yaralının korku ve endişelerini gidermek,
- Hasta/yaralıya müdahalede yardımcı olacak kişileri organize etmek,
- Hasta/yaralının durumunun ağırlaşmasını engellemek için gerekli müdahalede bulunmak,
- Kanama, kırık, çıkık ve burkulma vb. durumlarda yerinde müdahale etmek,
- Hasta/yaralıyı sıcak tutmak,
- Hasta/yaralının yarasını görmesine izin vermemek,
- Hasta/yaralıları hareket ettirmeden müdahale yapmak,
- Hasta/yaralıların en uygun yöntemlerle en yakın sağlık kuruluşuna sevkini sağlamak (112). Ancak, ağır hasta/yaralı bir kişi hayati tehlikede olmadığı sürece asla yerinden kıpırdatılmamalıdır (Hayat Kurtarma Zinciri, 2015).

3.2. Temel İlk Yardım Sertifikasını Kimler Almalıdır?

En az ilkokul mezunu olmak şartı ile herkes temel ilk yardım eğitimini ve sertifikasını alabilmektedir. Resmî Gazete’de yayınlanan 29 Temmuz 2015 ÇARŞAMBA 29429 sayılı yönetmelikte ilkyardımcı bulundurması zorunlu işyerleri ve kişi sayısı belirlenmiştir.

- Az tehlikeli işyerlerinde, her 20 çalışan için 1 ilkyardımcı,
- Tehlikeli işyerlerinde, her 15 çalışana kadar 1 ilkyardımcı,

- Çok tehlikeli işyerlerinde, her 10 çalışana kadar 1 ilkyardımcı, bulundurması zorunludur (İlkyardım Yönetmeliği, 2015).

3.3. Temel İlk Yardım Sertifikası Ve Sınavı

T.C. Sağlık Bakanlığı onaylı İlkyardım Eğitim Merkezleri tarafından verilen eğitim ve T.C. Sağlık Bakanlığı tarafından yapılan yazılı ve uygulamalı sınavlarından geçildiği takdirde temel ilk yardımcı sertifikası ve temel ilk yardımcı kimliği almaya hak kazanılır.

İlk yardım eğitim merkezleri tarafından en az 2 gün sürecek toplam 16 saat'ten oluşan eğitim verilmektedir. İlk yardım eğitiminden sonra en fazla 1 hafta içinde sınav yapılmaktadır. T.C. Sağlık Bakanlığı tarafından yapılan yazılı sınavından 100 üzerinden en az 85 puan alındığı takdirde aynı gün yapılan uygulamalı sınava girilebilir. Bu sınavdan da 100 üzerinden en az 85 puan alındığında temel ilk yardımcı sertifikası ve temel ilk yardımcı kimliği almaya hak kazanılır.

İlk yardım eğitimi alan kişilerin sınavlara, 2 defa girme hakkı bulunmaktadır. Temel ilk yardımcı sertifikası ve temel ilk yardımcı kimliği; eğitimi veren ilk yardım eğitim merkezleri tarafından T.C. Sağlık Bakanlığına onaylatılarak kişiye teslim edilir.

Temel ilk yardım eğitimi için belli bir ücret alınır. Temel ilkyardımcı sertifikasının geçerlilik süresi 3 yıldır. 3 yılın sonunda sadece güncelleme eğitimi alarak sertifika süresi uzatılmaktadır.

IV. MOBİL UYGULAMA

İnsanların teknolojiye olan ihtiyacı ve ilgisi gün geçtikçe artmaktadır. Mobil iletişim günlük yaşantıya o kadar girdi ki, bazı insanlar cep telefonları olmadan kendilerini eksik hissetmektedirler. Başlangıçta cep telefonlarının yapabildikleri sesli arama ve mesaj göndermekti. Teknolojinin gelişmesi, internetin cebe girmesi ile her tür işlemler akıllı telefonlarla yapılabilir duruma gelindi.

Akıllı telefonlar ve tabletler sadece iletişim kurmaya yarayan cihazlar değildir, ayrıca öğrenmeye, para kazanmaya ve eğlenceli vakit geçirmeye yarayan günümüz insanların vazgeçilmez araçlarıdır. Bunun için mobil uygulamaların geliştirilmesi gerekmektedir. Mobil uygulama; akıllı cihazlar için geliştirilen çevrimiçi veya çevrimdışı çalışan bir yazılımdır.

Günümüzde bir çok mobil işletim sistemi bulunmaktadır. Mobil işletim sistemi olarak en çok Android, iOS, Windows Phone ve BlackBerry OS kullanılmaktadır. Akıllı telefon işletim sistemi kullanım oranları yıllara göre 2012, 2013, 2014 ve Ağustos 2015 tarihi baz alınarak mobil işletim sistemi kullanım oranı Şekil 4.1'de gösterilmektedir (Smartphone OS Market Share, 2015 Q2, 2015).

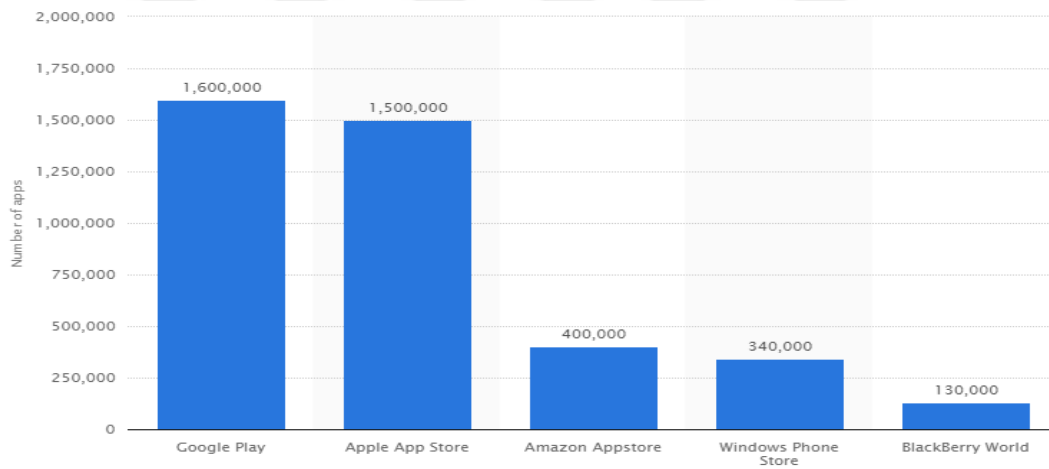
Period	Android	iOS	Windows Phone	BlackBerry OS	Others
2015Q2	82.8%	13.9%	2.6%	0.3%	0.4%
2014Q2	84.8%	11.6%	2.5%	0.5%	0.7%
2013Q2	79.8%	12.9%	3.4%	2.8%	1.2%
2012Q2	69.3%	16.6%	3.1%	4.9%	6.1%

Source: IDC, Aug 2015

Şekil 4.1. Ağustos 2015 mobil işletim sistemi kullanım oranı

Google firması tarafından geliştirilen Android işletim sistemine ait mobil uygulamalar Google play mağazasından indirilmektedir. Apple firması tarafından geliştirilen iOS işletim sistemine ait mobil uygulamalar itunes - Apple Store mağazasından indirilmektedir. Microsoft firması tarafından geliştirilen Windows Phone işletim sistemine ait mobil uygulamalar Windows Phone mağazasından indirilmektedir. Blackberry firması tarafından geliştirilen BlackBerry OS işletim sistemine ait mobil uygulamalar BlackBerry App World mağazasından indirilmektedir.

Temmuz 2015 tarihinde uygulama mağazalarının sayfalarında bulunan mobil uygulama programlarının sayısı Şekil 4.2'de gösterilmektedir (Number of apps available in leading app stores as of July 2015, 2015).



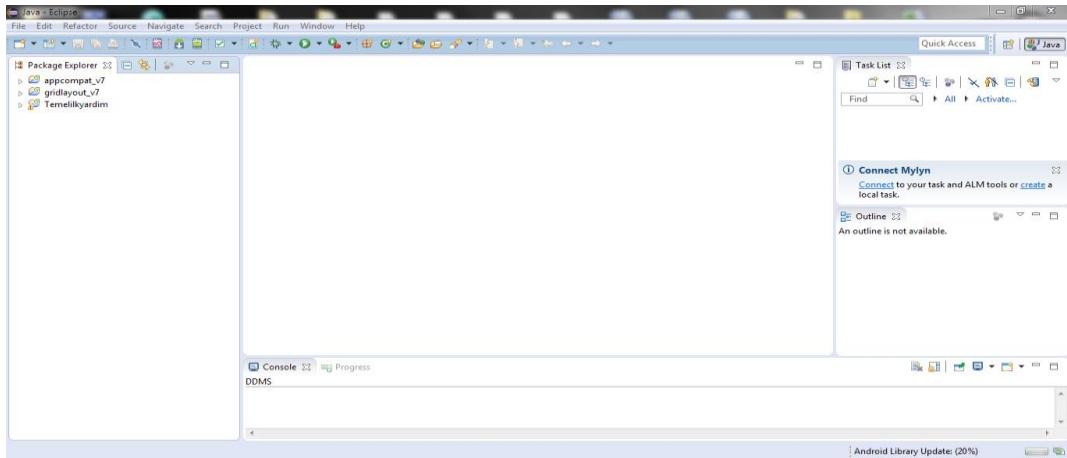
Şekil 4.2. Temmuz 2015 mobil uygulama sayıları

V. ANDROID MOBİL UYGULAMA YAZILIM GELİŞTİRME SÜRECİ

Temel ilk yardım uygulamasının yazılımı Eclipse ortamında Google'ın Android eklentisi kullanılarak java yazılım dili ile yazılmıştır. Uygulamanın kısmi test aşamasında sanal telefon cihazları kullanılmıştır. Android işletim sisteminde çalışacak uygulamaların geliştirilmesi için eclipse ortamında gerekli kurulum adımları ve uygulamaların test işlemi sırasında kullanılan sanal cihaz programlarının kurulum adımları incelenmektedir.

5.1. Eclipse Yazılım Geliştirme Editörü

Eclipse, açık kaynak kodlu ve proje odaklı geliştirmeye açık bir platformdur. Uygulama yazılım sürecinde Windows 7 professional 64 bit service pack 1 işletim sistemi kullanılmıştır. Android işletim sisteminde çalışacak uygulamaların geliştirilmesi için öncelikle Şekil 5.1'de gösterilen Eclipse ortamının hazır hale getirilmesi gerekmektedir.



Şekil 5.1. Eclipse ekranı

- Bilgisayara JDK (Java Development Kit) kurulumunun yapılması gerekmektedir.

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

adresine girerek işletim sistemine uygun JDK dosyası indirilmektedir.

Windows x64 bit işletim sistemi için windows x64, windows 32 bit işletim sistemi için windows x86 versiyonu indirilip, kurulumu yapılmaktadır.

- Eclipse IDE for Java Developers indirilmesi gerekmektedir. Eclipse sitesinde yer alan <http://www.eclipse.org/downloads/> adresinden işletim sistemine uygun 32bit veya 64bit sürümü indirilmektedir. Eclipse IDE'yi indirdikten sonra zip içerisinden çıkartılır ve Eclipse kurulumuna ihtiyaç duymadığından dolayı klasör içindeki Eclipse.exe uzantılı dosya çalıştırılır.
- Android projelerinin daha rahat geliştirilmesi için Android SDK (Software Development Kit) kullanılır.

<http://developer.android.com/sdk/index.html#Other> adresinden installer_r24.

exe dosyasının kurulması gerekmektedir. Kurulum yeri olarak Eclipse klasörünün içinde android_sdk klasörü oluşturulup bu klasörün içine SDK yüklenir. Ancak istenirse bilgisayarın başka bir disk alanına da kurulum yapılabilir.

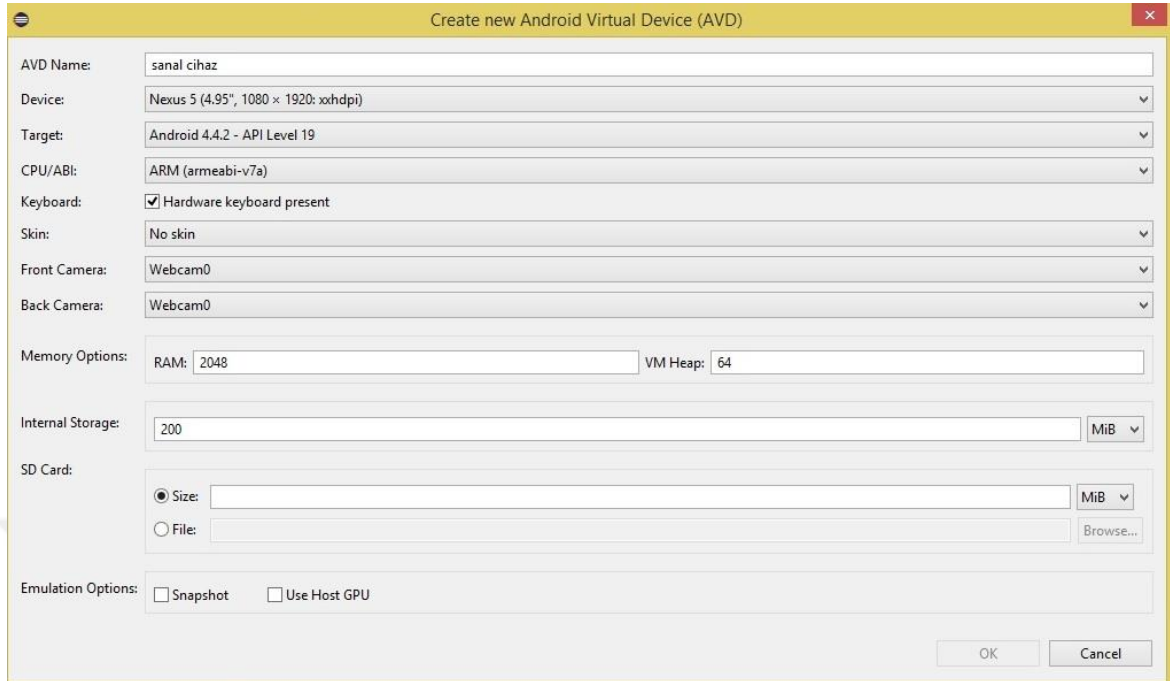
- Eclipse'i daha verimli kullanmak ve Eclipse üzerinde Android uygulamalarını geliştirebilmek için ADT (Android Development Tools) pluginin kurulması gerekmektedir. Eclipse'i açarak Help - Install New Software seçeneği seçilir ve Add butonuna basılarak açılan pencerede name kısmına ADT Plugin, Location sekmesine <https://dl-ssl.google.com/android/eclipse/> yazılıp ok butonuna tıklanır. Developer Tools'u seçip yükleme işlemi yapılır.

- Kurulum bittikten sonra Eclipse yeniden başlatılır. Eclipse açıldığında Open SDK Manager seçeneği gelmektedir. Ancak bu pencere kapatıldığı durumda Eclipse menüleri üzerinden kurulum yapmak gerekmektedir. Windows - Preferences - Android kısmı tıklanır ve buradaki SDK Location bölümünde Android SDK'nın kurulu olduğu klasörün seçilmesi gerekmektedir. Daha sonra Window - Android SDK Manager seçeneğini seçilir. Öncelik olarak tools ayarları, extras dosyasındaki Android Support Library'i ve çalışmanın yapılacağı android versiyonlarının API'leri seçilip kurulumu yapılır.

5.2. Android Sanal Cihaz

Yapılmış olan android uygulamalarını bilgisayarda test etmek için Android sanal cihazı AVD (Android Virtual Device)'nin kurulması gerekmektedir. Yapılan uygulamaların farklı cihazlarda nasıl çalışacağını test edilmesi için birden fazla sanal cihaz oluşturulur.

Eclipse programı açılır. Windows - Android Virtual Device Manager seçeneği seçilir. Açılan pencerede sağ üst köşede yer alan " Create " basarak yeni sanal cihaz ayarları ekranı açılır. Şekil 5.2'de android sanal cihaz ayarları ekranı gösterilmektedir.



Şekil 5.2. Android sanal cihaz ayarları

5.2.1. Android sanal cihaz ayarları

AVD Name: Sanal aygıtın ismi yazılır.

Device: Sanal aygıt cihaz modeli seçilir.

Target: Android sürümü seçilir.

CPU/ABI: Telefonlar için ARM (armeabi) ve Intel Atom iki seçenek bulunmaktadır.

Telefonlar için genelde ARM (armeabi) kullanılmaktadır.

Keyboard: Eğer programda yazı yazılacak ise keyboard seçeneği seçilerek bilgisayarın klavyesi kullanılabilir. Keyboard seçeneği seçilmemesi durumunda yazı işlemini yapmak için emülatörün sanal klavyesinin kullanılması gerekmektedir.

Skin: Skin with dynamic hardware controls seçeneğinin seçilmesi durumunda sanal cihaz uygulamanın yanında tuş kısmı gelmektedir. No skin yapılması durumunda sadece sanal cihaz ekranı açılacaktır.

Front Camera: Uygulamada ön kamera işlemi test edilmek isteniyor ise bu seçenek kullanılmalıdır. Uygulama bilgisayarın kamerasını kullanmaktadır. Kamera kullanılmak istenmiyor ise None seçeneği seçilir.

Back Camera: Arka kamera ile ilgili uygulamada bu seçenek kullanılır. Arka kamera kullanılmak istenmiyor ise None seçeneği seçilir.

Memory options: İki kısım bulunmaktadır. Birinci kısım RAM kısmı; cihazın sanal belleği belirlenir. İkinci kısım VM Heap yığın kapasitesi değeri yer almaktadır genelde varsayılan değer olarak bırakılabilir.

Internal storage: Dahili depolama alanıdır.

SD Card: SD kartın kapasitesi belirlenir.

Snapshot: Emülatörün ImageCopy'si alınır. Emülatörün bir sonraki açılışta daha hızlı açılmasını sağlar.

Use Host GPU: Bilgisayarın işlemcisini kullanır ve emülatörün performansını artırır.

Sanal cihazın ayarları yapıldıktan sonra ok butonuna basılır. Sanal cihaz çalıştırmak istendiğinde sanal cihaz seçilerek start denir. Ekran Launch Options penceresi gelmektedir. Scale display to real size seçeneği seçilir ve screen size (in) seçeneğinden sanal cihazın ekran boyutu ayarlanır. Launch butonuna basılarak sanal cihazın Şekil 5.3'deki gibi açılması sağlanır.



Şekil 5.3. Android sanal cihaz

Ancak AVD (Android Virtual Device) sanal cihaz hızlandırma ayarları yapılmasına rağmen ilk açılış sürecinde çok yavaş açılmakta ve bilgisayar ram miktarını çok fazla kullanmaktadır.

5.3. Genymotion Sanal Cihaz

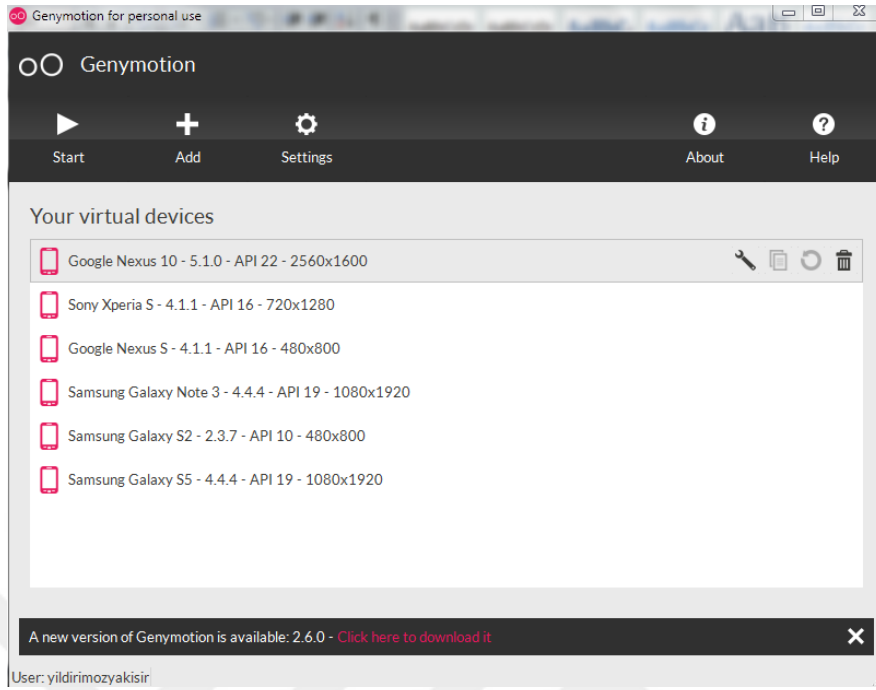
Genymotion sanal cihaz programı üçüncü parti bir programdır. AVD sanal cihazın yavaşlığı ve sorunlar çıkartmasından dolayı kullanımı kolay ve daha hızlı çalışan, güncel telefonların test edilebildiği Genymotion sanal cihaz kullanılmaktadır. Genymotion emülatör programı Oracle ürünü olan VirtualBox sanal makina üzerinden çalışmaktadır.

5.3.1. Genymotion sanal cihaz kurulumu

- Genymotion'un resmi sitesi olan www.genymotion.com'a girilir ve Sign Up bölümünden üyelik açılarak giriş işlemi yapılır. Ücretli ve ücretsiz sürümleri

bulunmaktadır. Ücretiz sürüm indirilir. Genymotion kurulumu ve Oracle virtual box kurulumu yapılır.

- Genymotion programını açılır ve ekranda "You have no virtual device available yet. Do you want to add a new one?" sorusu gelmektedir. Yes denildiğinde Genymotion web sitesine üye olunan kullanıcı adı ve şifresi girilmelidir.
- Genymotion programı üzerinden giriş işlemi yapıldıktan sonra Genymotion ve Eclipse programlarında ayar yapılması gerekmektedir. Genymotion Settings kısmında: ADB tool connection setting – Use custom Android SDK tools ayarında bilgisayarda kurulu olan Android SDK klasörü gösterilmektedir.
- Eclipse tarafında Genymotion programı ayarı yapılması gerekmektedir. Eclipse programı açılır Help - Install New Software sekmesi seçilir ve gelen ekranda sağ taraftaki Add butonuna basılır ve gelen ekranda "Name" kısmına Genymotion yazılır ve "Location" kısmına <http://plugins.genymotion.com/eclipse> adresi yazılır. En son adım olarak ok butonu basılarak kurulum işlemi yapılmaktadır.
- Eclipse programında Windows - Preferences kısmı tıklanır. Açılan ekranda Genymobile altında bulunan Genymotion sekmesi seçilir. Genymotion Directory kısmında Genymotion programının kurulu olduğu klasör seçilir.
- Genymotion programı üzerinden Add seçeneği tıklanır gelen ekranda istenilen android sürümü ve cihaz modeli seçilerek sanal cihaz kurulumu yapılır. Tüm kurulum işlemleri tamamlanmıştır. Şekil 5.4'de kurulan genymotion sanal cihazlar gösterilmektedir. Sanal cihaz seçilip start butonuna basılarak sanal cihaz çalıştırılır.



Şekil 5.4. Genymotion program ekranı

VI. TEMEL İLK YARDIM UYGULAMASI SÜRECİ

Bu tezde yapılan çalışma, insanlar için herhangi bir kaza ya da yaşamı tehlikeye düşüren bir durumda sağlık ekiplerinin olay yerine ulaşmaya dek insan hayatının kurtarılması veya kötüye gitmesini engellemek için başvurulacak bir uygulamadır. Kişiler hasta/yaralılarına yardımcı olması gerektiğinde, yanlış bilgiler ile panik olması sonucu hastayı riske sokacak yanlış müdahaleler yapabilirler. İlk yardım uygulaması, insanlar için kullanımı kolay doğru bilgilerin bulunduğu gerçek hayattan esinlenilerek geliştirilen bir uygulamadır.

Uygulamayı geliştirmeden önce uygulamada yer alacak bilgilerin eğitimini almak ve doğru bilgilerin yazılması için Temel İlk Yardım eğitimi Beyaz Gemi İlk Yardım Eğitim Merkezi tarafından ve T.C. Sağlık Bakanlığı tarafından yapılan Test sınavı ve Uygulama sınavı geçilerek Temel İlk Yardımcı sertifikası ve Temel İlk Yardımcı kimliği alınmıştır. Bahsi geçen sertifika “EK B” ve kimlik “EK C” bölümünde yer almaktadır.

Temel ilk yardım uygulaması ilk yardım, ek bilgi, eğitim süreci ve test bölümünden oluşmaktadır. Android işletim sistemli telefon cihazlarının ekran boyutları birbirinden farklı olması ve uygulamanın dikey/yatay olarak uyumlu çalışabilmesi için layout, layout-land, layout-large, layout-small, layout-xlarge ekran boyut özelliğinden yararlanılmıştır. Uygulamada yer alan resimlerinde farklı boyutlardaki cihazlarda dikey/yatay olarak uyumlu çalışabilmesi için drawable-hdpi, drawable-ldpi, drawable-mdpi, drawable-xhdpi, drawable-xxhdpi görüntü çözünürlük özelliğinden yararlanılmıştır. Uygulamada kullanılan resimler ücretli ve ücretsiz resimlerin yer

aldığı <https://pixabay.com/> sayfasından ücretsiz resimler alınmıştır. Uygulama ikon tasarımı kırmızı çanta üzerinde yara bandı, kızıl ay simgesi, tıp sembolü ve büyük harfler ile “TEMEL İLK YARDIM” yazısı yer almaktadır. Temel ilk yardım uygulama ikon tasarımı Şekil 6.1’de yer almaktadır. Uygulama ikon tasarımı için photoshop programından yararlanılmıştır.

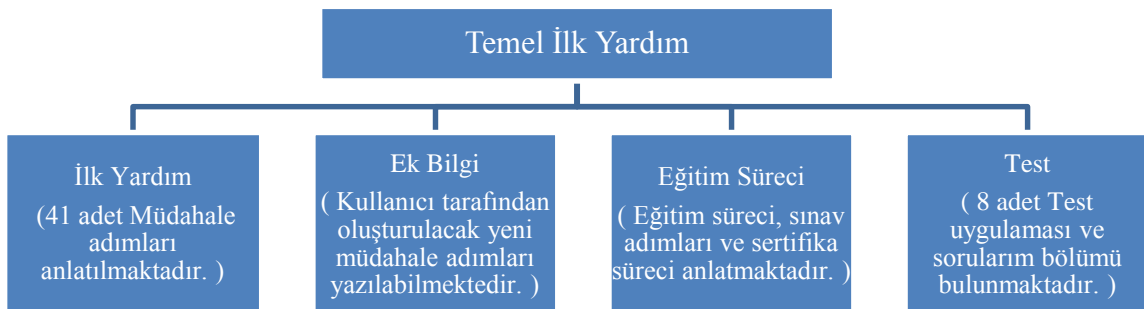


Şekil 6.1. Temel ilk yardım uygulama ikon

İlk yardım bölümünde yer alan videolar Türk Kızılayı <https://www.youtube.com/user/TurkKizilayiOrg/videos> resmi youtube sayfasından Türk kızıl ayı’ndan onay alınarak uygulamaya eklenmiştir. Bahsi geçen onay izni “EK A” bölümünde yer almaktadır.

6.1. Ekran Tasarımı

Temel ilk yardım uygulamasının ekran tasarımı için TabHost özelliğinden yararlanılmıştır. Ekran tasarımı Şekil 6.2’de gösterilmektedir.



Şekil 6.2. Temel ilk yardım ekran tasarımı

Şekil 6.3’de dört tab başlık ekranı yer almaktadır.



Şekil 6.3. Dört tab başlık ekranı

Dört adet tab İlk yardım, Ek bilgi, Eğitim süreci ve Test başlıkları activity_main.xml dosyasında tasarlanmıştır ve MainActivity.java dosyası ile eşleştirilmiştir. MainActivity.java dosyasında activity_main.xml TabHost ekranında yer alan tab butonlarına basıldığında yönlendirilecek Tab1, Tab2, Tab3, Tab4 ekranların tanımları yapılmıştır. Eclipse ortamında yeni bir java dosyası yaratıldıktan sonra ilgili sayfaların açılması için AndroidManifest.xml dosyasında java sayfasının tanımı yapılmaktadır. Şekil 6.4’de ekran tasarım activity_main.xml kodu yer almaktadır.

```
<LinearLayout 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=".MainActivity"
    android:baselineAligned="false"
    android:background="@drawable/arkaplan">
    <TabHost
        android:id="@android:id/tabhost"
        android:layout_width="0dip"
        android:layout_height="match_parent"
        android:layout_weight="1" >
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical" >
```

```
<TabWidget
  android:id="@android:id/tabs"
  android:layout_width="match_parent"
  android:layout_height="wrap_content" >
</TabWidget>

<FrameLayout
  android:id="@android:id/tabcontent"
  android:layout_width="match_parent"
  android:layout_height="match_parent" >

  <LinearLayout
    android:id="@+id/tab1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
  </LinearLayout>

  <LinearLayout
    android:id="@+id/tab4"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
  </LinearLayout>

  <LinearLayout
    android:id="@+id/tab2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
  </LinearLayout>

  <LinearLayout
    android:id="@+id/tab3"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
  </LinearLayout>
</FrameLayout>
</LinearLayout>
</TabHost>
</LinearLayout>
```

Şekil 6.4. Ekran tasarım kodu activity_main.xml

Şekil 6.5’de ekran tasarımı MainActivity.java kodu yer almaktadır.

```

package com.magellsoft.temelilkyardim;
import android.app.TabActivity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.TabHost;
import android.widget.TabHost.TabSpec;

public class MainActivity extends TabActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //Tabhost oluştur.
        TabHost tabhost = (TabHost)findViewById(android.R.id.tabhost);

        //Sekme adı ve tabhost
        TabSpec tab1 = tabhost.newTabSpec("1.tab");
        TabSpec tab4 = tabhost.newTabSpec("4.tab");
        TabSpec tab2 = tabhost.newTabSpec("2.tab");
        TabSpec tab3 = tabhost.newTabSpec("3.tab");

        tab1.setIndicator("İlk Yardım");
        tab1.setContent(new Intent(this,Tab1.class));
        tab4.setIndicator("Ek Bilgi");
        tab4.setContent(new Intent(this,Tab4.class));
        tab2.setIndicator("Eğitim Süreci");
        tab2.setContent(new Intent(this,Tab2.class));
        tab3.setIndicator("Test");
        tab3.setContent(new Intent(this,Tab3.class));

        //Görüntülenecek tabhost sekmeleri
        tabhost.addTab(tab1);tabhost.addTab(tab4); tabhost.addTab(tab2);
        tabhost.addTab(tab3);

    }
}

```

Şekil 6.5. Ekran tasarımı MainActivity.java kodu

6.2. İlk Yardım

Ekran tasarımı olarak; resim, uygulama hakkında ve kırk bir adet ilk yardım bilgilerinin yer aldığı buton bulunmaktadır. İlk yardım ana ekran Şekil 6.6'da gösterilmektedir.



Şekil 6.6. İlk yardım ana ekran

İlk yardım müdahale butonları:

- Abc değerlendirilmesi
- Bilinç bozukluğu
- Boğulma
- Burkulma
- Burun kanaması
- Burna yabancı cisim kaçması
- Çıkık

- Donukta ilk yardım
- Göğüs ağrısı
- Göze yabancı cisim kaçması
- Havale
- Hayvan ısırması
- İkinci değerlendirme
- Kafatası ve omurga yaralanması
- Kalp masajı ve solunum – Bebek
- Kalp masajı ve solunum – Çocuk
- Kalp masajı ve solunum – Yetişkin
- Kanama - Dış kanama
- Kanama - İç kanama
- Kırık
- Koma pozisyonu
- Kulağa yabancı cisim kaçması
- Kulak kanaması
- Sara krizi
- Sıcak çarpması
- Solunum - Kısmi tıkanma
- Solunum - Tam tıkanma
- Şok
- Taşıma teknikleri
- Temel yaşam desteği – Bebek

- Temel yaşam desteği – Çocuk
- Temel yaşam desteği – Yetişkin
- Turnike uygulaması
- Uzuv kopması
- Yanık - Elektrik ile oluşan
- Yanık - Isı ile oluşan
- Yanık - Kimyasal ile oluşan
- Yaralanma ilk yardım
- Zehirlenme - Cilt yoluyla
- Zehirlenme - Sindirim yoluyla
- Zehirlenme - Solunum yoluyla

İlk yardım ekranında LinearLayout yapısında ScrollView özelliği ile aşağı/yukarı yapılarak diğer ilk yardım bilgilerine ulaşım sağlanmış ve menüler buton olarak tasarlanmıştır. Şekil 6.7’de ilk yardım ekran tasarımı ve ABC değerlendirilmesi butonu xml kodu yer almaktadır.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <ScrollView
        android:id="@+id/scrollView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical" >
```



```

<ImageView
    android:id="@+id/imageView1"
    android:layout_width="wrap_content"
    android:layout_height="200dp"
    android:src="@drawable/tab1resim" />
    <Button
        android:id="@+id/Button42"
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:background="@drawable/hakkinda" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal" >
        <Button
            android:id="@+id/Button001"
            android:layout_width="50dp"
            android:layout_height="45dp"
            android:background="@drawable/abcdegerlendirmesi" />
        <Button
            android:id="@+id/Button01"
            android:layout_width="match_parent"
            android:layout_height="45dp"
            android:background="@drawable/buttontikla"
            android:gravity="left/center_vertical"
            android:text="ABC Değerlendirilmesi"/>
    </LinearLayout>

```

Şekil 6.7. İlk yardım ekran tasarımı ve ABC değerlendirilmesi butonu xml kodu

İlk yardım butonları basıldığında buton ile eşleşen sayfa açılmaktadır. Şekil 6.8'de ABC değerlendirilmesi buton kodu yer almaktadır.

```

package com.magellsoft.temelilkyardim;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class Tab1 extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.tab1);
    }
}

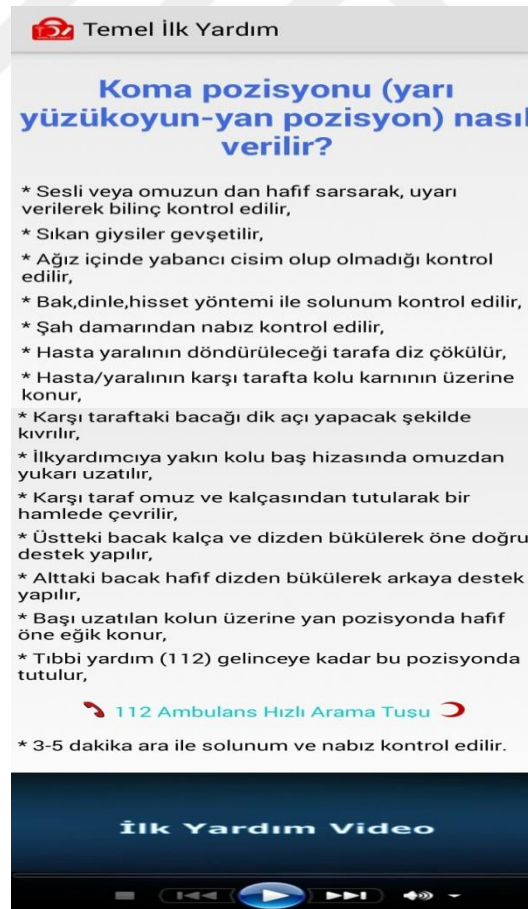
```

// 42 adet Button bulunmaktadır. Button'lara basıldığında yönlendirildiği sayfayı açmaktadır.

```
Button activity1=(Button) findViewById(R.id.Button01);
activity1.setOnClickListener(new OnClickListener () {
    public void onClick(View v){
        startActivity(new Intent(Tab1.this, abc.class));
    }
});
```

Şekil 6.8. ABC değerlendirilmesi buton kodu

İstenilen ilk yardım bilgi sayfası açıldığında müdahale tanımı, müdahale adımları, müdahale adımlarında acil yardım numarasının aranması belirtiliyor ise numaranın hızlı aranmasını sağlayan buton ve çok önemli ilk yardım müdahalelerinde ilk yardım müdahalesini anlatan video'lar yer almaktadır. Şekil 6.9'da koma pozisyonu bilgi ekranı yer almaktadır.



Şekil 6.9. Koma pozisyonu bilgi ekranı

İlk yardım bilgi ekranında LinearLayout yapısında ScrollView özelliği ile aşağı/yukarı yapılarak bilgilerin okunması sağlanmıştır. Koma pozisyonundaki müdahale adımları Şekil 6.10'da yer alan TextView yapısı kullanılarak yazılmıştır.

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

    <ScrollView
        android:id="@+id/scrollView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical" >
            <TextView
                android:id="@+id/TextView01"
                android:layout_width="match_parent"
                android:layout_height="wrap_content" />
            <TextView
                android:id="@+id/TextView02"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:gravity="center"
                android:text="Koma pozisyonu (yarı yzkoyun-yan pozisyon) nasıl verilir?"
                android:textColor="#4169e1"
                android:textSize="25sp"
                android:textStyle="bold" />
        </LinearLayout>
    </ScrollView>
</LinearLayout>
```

Şekil 6.10. Koma.xml TextView kodu

Acil numaraların otomatik aranması için buton yapısından yararlanılmıştır. Hızlı arama tuşu Şekil 6.11'de gösterilmektedir.



Şekil 6.11. Hızlı arama tuşu

Hızlı arama tuşu butonuna basıldığında Şekil 6.12’de yer alan kod çalışmakta ve arama kodunda yazılan numara aranmaktadır.

```
// button1 tıklandığında 112 numarasını arama işlemi yapmaktadır.

    Button button = (Button) findViewById(R.id.button1);
    button.setOnClickListener(new OnClickListener() {

        @Override
        public void onClick(View arg0) {

            Intent callIntent = new Intent(Intent.ACTION_CALL);
            callIntent.setData(Uri.parse("tel:112"));
            startActivity(callIntent);

        }
    });
```

Şekil 6.12. Acil numara arama kodu

Uygulamada tanımlanan acil numaranın aranması için, Şekil 6.13’deki AndroidManifest.xml - permission ayarında CALL_PHONE yetkisi verilmektedir.

```
<uses-permission android:name="android.permission.CALL_PHONE" />
```

Şekil 6.13. AndroidManifest telefon arama yetkisi

Video oynatmak için buton yapısından yararlanılmıştır. Şekil Şekil 6.14’de ilk yardım video butonu gösterilmektedir.



Şekil 6.14. İlk yardım video butonu

Video butonuna basıldığında şekil 6.15’de yer alan video oynatma sayfası açılmaktadır.



Şekil 6.15. Koma pozisyonu video oynatma sayfası

Video oynatmak için Şekil 6.16’da yer alan video oynatma kodu çalıştırmaktadır.

```

package com.magellsoft.temelilkyardim;

import android.app.Activity;
import android.net.Uri;
import android.os.Bundle;
import android.widget.MediaController;
import android.widget.VideoView;

public class komavideo extends Activity {

    private VideoView Videooynat;
    private MediaController videokontrol;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // komavideo layout göster

        setContentView(R.layout.komavideo);
        videokontrol = new MediaController(komavideo.this);

        // komavideo içinde bulunan videonun id ismi yazılır.

        Videooynat = (VideoView) findViewById(R.id.komavideo);
        Videooynat.setMediaController(videokontrol);
    }
}

```

```

        Videooynat.setVideoURI(Uri.parse("android.resource://" + getPackageName()
+ "/" + R.raw.koma));
        Videooynat.start();
    }
}

```

Şekil 6.16. Video oynatma kodu

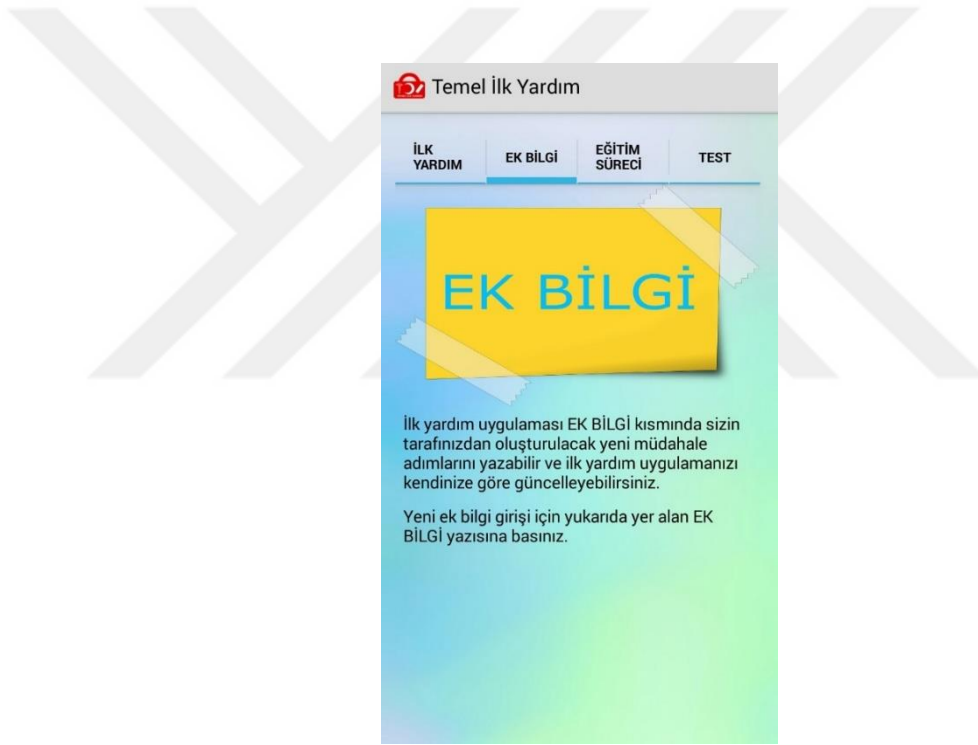
Temel ilk yardım uygulamasında toplam yedi adet video bulunmaktadır. Video'lar Türk Kızılayı resmi Youtube sayfasından Kızılay'dan onay alınarak uygulamaya yüklenmiştir. Eclipse ortamında videoların çalışabilmesi için res klasörünün içerisinde raw adlı bir klasör oluşturulmakta ve videolar raw klasörüne eklenmektedir. Videoların uygulama içine yüklenmesi aşamasında çok fazla alan kaplamaması için 3gp cep telefonları için kullanılan video formatına çevrilerek eklenmiştir. Videoların uygulamanın içine yüklenmesinin sebebi, uygulamayı kullanacak kişinin o anki acil durumda mobil hattında internet paketinin olup olmaması, mobil şebeke ve wireless sinyalinin çekmemesi sonucu videoları izlememe riskini ortadan kaldırmak içindir.

Videolar :

- Yetişkinlerde temel yaşam desteği 8 yaş üstü kişiler,
- 1 - 8 yaş çocuklarda temel yaşam desteği,
- 0 - 1 yaş bebeklerde temel yaşam desteği,
- Komada ilkyardım,
- Hasta yaralıyı araçtan çıkarma manevrası,
- Çocuk yetişkin tam tıkanmada ilkyardım 1 - 8 yaş ve üstü kişilerde,
- Dış kanama ve turnike (Kızılayı, 2015).

6.3. Ek Bilgi

Uygulamayı kullanacak kişi tarafından oluşturulacak yeni ilk yardım müdahale adımlarının yazılabileceği ek bilgi ekranı geliştirilmiştir. Bu sayede uygulamayı kullanacak kişiye temel ilk yardım eğitimi dışında sağlıkla ilgili bilgileri girebileceği ortam yaratılmıştır. Ek bilgi sekmesine tıklandığında Şekil 6.17'deki ekran gelmektedir. Yeni bilgileri görmek ve kaydetmek için Şekil 6.17'deki ek bilgi butonuna basılmaktadır.



Şekil 6.17. Ek bilgi sekmesi ve ek bilgi butonu

Ek bilgi ekran tasarımı xml kodu Şekil 6.18'de yer almaktadır.

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

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

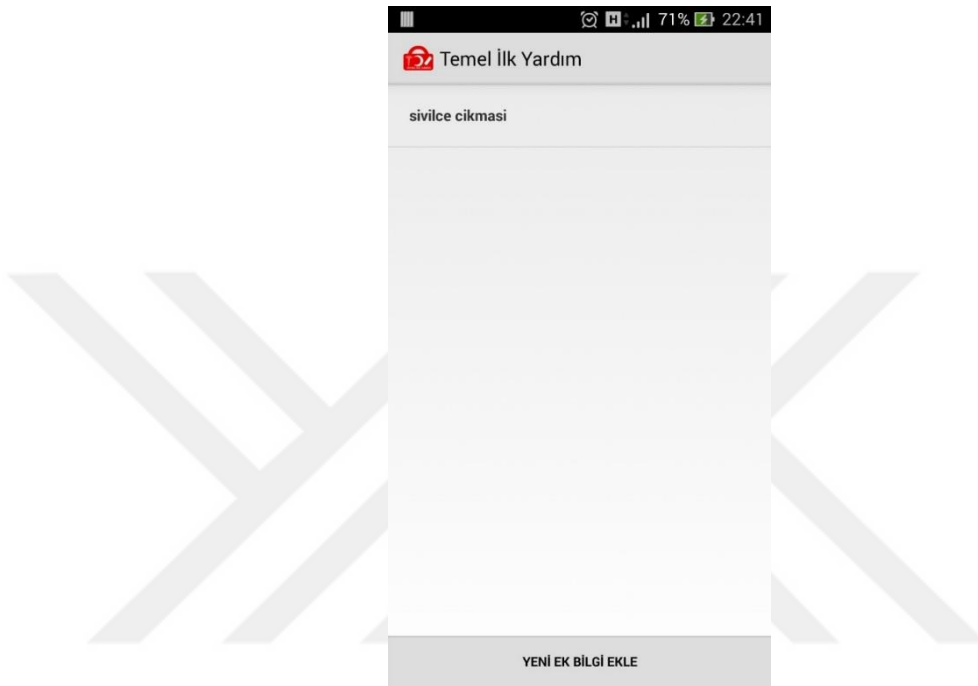
```

<ScrollView
    android:id="@+id/scrollView1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical" >
        <Button
            android:id="@+id/notlarim"
            android:layout_width="wrap_content"
            android:layout_height="200dp"
            android:background="@drawable/tab4resim" />
        <TextView
            android:id="@+id/TextView01"
            android:layout_width="match_parent"
            android:layout_height="5dp" />
        <TextView
            android:id="@+id/TextView02"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="left"
            android:text="İlk yardım uygulaması EK BİLGİ kısmında sizin tarafınızdan
            oluşturulacak yeni müdahale adımlarını yazabilir ve ilk yardım uygulamanızı
            kendinize göre güncelleyebilirsiniz."
            android:textColor="#000000"
            android:textSize="15sp"
            android:layout_marginRight="7dp"
            android:layout_marginLeft="7dp"/>
        <TextView
            android:id="@+id/TextView03"
            android:layout_width="match_parent"
            android:layout_height="10dp" />
        <TextView
            android:id="@+id/TextView04"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="left"
            android:text="Yeni ek bilgi girişi için yukarıda yer alan EK BİLGİ yazısına
            basınız."
            android:textColor="#000000"
            android:textSize="15sp"
            android:layout_marginRight="7dp"
            android:layout_marginLeft="7dp"/>
    </LinearLayout>
</ScrollView>

```

Şekil 6.18. Ek bilgi sekmesi xml kodu

Ek bilgi butonuna basıldığında Şekil 6.19’da kaydedilen ek bilgiler ekranı gelmekte ve yeni ek bilgi yazmak için ekranın en altında yer alan yeni ek bilgi ekle butonu kullanılmaktadır.



Şekil 6.19. Ek bilgi ekranı ve yeni ek bilgi ekle butonu

Ek bilgi butonuna basıldığında Şekil 6.20’de yer alan ek bilgi java kodu çalışmaktadır.

```
package com.magellsoft.temelilkyardim;

import android.app.ListActivity;
import android.content.Intent;
import android.database.Cursor;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.MenuItem.OnMenuItemClickListener;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.CursorAdapter;
import android.widget.ListView;
import android.widget.SimpleCursorAdapter;
```

```

public class yeninotekle extends ListActivity {

    // Değişkenleri Belirle
    public static final String ID = "id";
    private static final String BASLIK = "baslik";
    private ListView notlistegoruntule;
    private CursorAdapter notbaglayici;

    @Override
    public void onCreate(Bundle kayitlinotlar) {
        super.onCreate(kayitlinotlar);
        // ListView göster
        notlistegoruntule = getListView();
        // Liste görünümü
        notlistegoruntule.setOnItemClickListener(notsiralala);

        // Bu ViewTitleNotes tüm başlıkları TextView göster
        String[] baslik = new String[] { BASLIK };
        int[] not = new int[] { R.id.Notlarigoster };

        // Basit imleç adaptörü oluştur
        notbaglayici = new SimpleCursorAdapter(yeninotekle.this, R.layout.notlistele, null,
        baslik, not);
        // Basit İmleç Adaptörü içine Adaptörü ayarlama
        setListAdapter(notbaglayici);
    }

    // not sirala
    OnItemClickListener notsiralala = new OnItemClickListener() {
    public void onItemClick(AdapterView<?> arg0, View arg1, int arg2, long arg3) {

        // notgoruntule activity çalıştır
        Intent notugosterbuton = new Intent(yeninotekle.this, notgoruntule.class);

        // not aktivitesini Görüntülemek için satır kimliğini ilet
        notugosterbuton.putExtra(ID, arg3);
        startActivity(notugosterbuton);
    }
    };

    @Override
    protected void onResume() {
        super.onResume();

        //not ekle kısmına geri gel
        new notal().execute((Object[]) null);
    }
}

```

```

@Override
protected void onStop() {
    Cursor imlec = notbaglayici.getCursor();

    // imleç devre dışı bırak
    if (imlec != null)
        imlec.deactivate();

    notbaglayici.changeCursor(null);
    super.onStop();
}

// Yeni ek bilgi ekle option menu oluştur

@Override
public boolean onCreateOptionsMenu(Menu menu) {

    // Yeni ek bilgi ekle Menu baslik

    menu.add("Yeni Ek Bilgi Ekle")
        .setOnMenuItemClickListener(this.yeninoteklelistesi)

        .setShowAsAction(MenuItem.SHOW_AS_ACTION_IF_ROOM);

    return super.onCreateOptionsMenu(menu);
}

// not ekle için tıkla
OnMenuItemClickListener yeninoteklelistesi = new
OnMenuItemClickListener() {

    public boolean onOptionsItemSelected(MenuItem item) {

        // notkaydet activity aç
        Intent notklebuton = new Intent(yeninotekle.this, notkaydet.class);
        startActivity(notklebuton);

        return false;
    }
};

private class notal extends AsyncTask<Object, Object, Cursor> {
    veritabani dbbaglayici = new veritabani(yeninotekle.this);

    @Override
    protected Cursor doInBackground(Object... params) {

```

```

// dbbaglayici aç
    dbbaglayici.ac();

    return dbbaglayici.tumnotlariliste();
}
@Override
protected void onPostExecute(Cursor islem) {
    notbbaglayici.changeCursor(islem);

// dbbaglayici kapat
    dbbaglayici.kapat();
}
}
}

```

Şekil 6.20. Ek bilgi java kodu

Yeni ek bilgi ekle butonu basıldığında EditText yapısı kullanılarak Şekil 6.21'deki belirti adı, belirtinin tanımı, belirtinin özellikleri ve müdahale işlemleri kısımlarından oluşan yazı yazma ekranı gelmektedir. Gerekli alanlar doldurulduktan sonra ek bilgi kaydet butonuna basılır.

Şekil 6.21. Ek bilgi kaydet ekranı

Ek bilgi kaydet java kodu Şekil 6.22’de yer almaktadır.

```

package com.magellsoft.temelilkyardim;

import android.app.Activity;
import android.app.AlertDialog;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.MenuItem.OnMenuItemClickListener;
import android.widget.EditText;

public class notkaydet extends Activity {

    // Değişkenleri Belirle
    private long ID;
    private EditText Belirtiadiedit;
    private EditText Belirtinintanimiedit;
    private EditText Belirtininozellikleriedit;
    private EditText Mudahaleislemleriedit;
    private static final String BASLIK = "baslik";
    private static final String NOT1 = "not1";
    private static final String NOT2 = "not2";
    private static final String NOT3 = "not3";

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.notekle);

        // notekle.xml içinde EditText bul
        Belirtiadiedit = (EditText) findViewById(R.id.Belirtiadiedit);
        Belirtinintanimiedit = (EditText) findViewById(R.id.Belirtinintanimiedit);
        Belirtininozellikleriedit = (EditText) findViewById(R.id.Belirtininozellikleriedit);
        Mudahaleislemleriedit = (EditText) findViewById(R.id.Mudahaleislemleriedit);

        // notgoruntule.java gelen Satır kimliği Al
        Bundle extras = getIntent().getExtras();
        if (extras != null) {
            ID = extras.getLong("id");
            Belirtiadiedit.setText(extras.getString(BASLIK));
            Belirtinintanimiedit.setText(extras.getString(NOT1));
            Belirtininozellikleriedit.setText(extras.getString(NOT2));
            Mudahaleislemleriedit.setText(extras.getString(NOT3));
        }
    }
}

```

```

// Ek bilgi kaydet seçenek menüsü oluştur
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    menu.add("Ek Bilgi Kaydet")
        .setOnMenuItemClickListener(this.bilgikaydet)

    .setShowAsAction(MenuItem.SHOW_AS_ACTION_IF_ROOM);

    return super.onCreateOptionsMenu(menu);
}
// menü için tıklayınız.
OnMenuItemClickListener bilgikaydet = new OnMenuItemClickListener() {
    public boolean onMenuItemClick(MenuItem item) {

        // Kaydetme işlevi
        if (Belirtiadedit.getText().length() != 0) {
            AsyncTask<Object, Object, Object> notkaydetkontrol = new AsyncTask<Object,
            Object, Object>() {
                @Override
                protected Object doInBackground(Object... params) {
                    notkaydet();
                    return null;
                }

                @Override
                protected void onPostExecute(Object result) {
                    // activity kapalı
                    finish();
                }
            };
            // Yukarıdaki işlemi yürütme
            notkaydetkontrol.execute((Object[]) null);
        }

        else {
            // Belirti adı boş bırakılmaması için uyarı mesajı ekrana gelir.
            AlertDialog.Builder uyari = new AlertDialog.Builder(notkaydet.this);
            uyari.setTitle("Lütfen Eksik Alanları Doldurunuz.!");
            uyari.setMessage("Belirti Adı girmeniz gerekmektedir.");
            uyari.setPositiveButton("Tamam", null);
            uyari.show();
        }
        return false;
    }
};

```

```

// nokt kaydet fonksiyonu
private void notkaydet() {
    veritabani dbbaglayici = new veritabani(this);

    if (getIntent().getExtras() == null) {

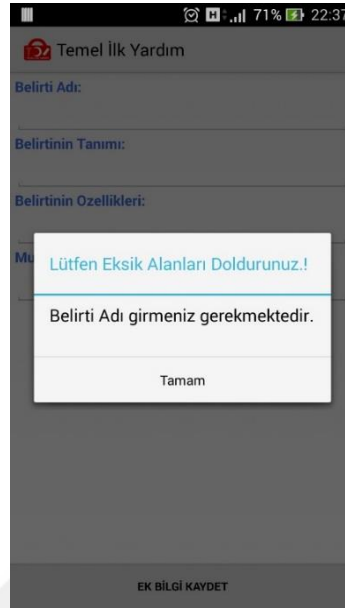
        // Veri dbbaglayici.java içinde not ekle
        dbbaglayici.notekle(Belirtiadiedit.getText().toString(),
            Belirtinintanimiedit.getText().toString(),
            Belirtininozellikleriedit.getText().toString(),
            Mudahaleislemleriedit.getText().toString());
        } else {

        // Veri dbbaglayici.java içinde notlartıgüncelle ekle
        dbbaglayici.notguncelle(ID,
            Belirtiadiedit.getText().toString(),Belirtinintanimiedit.getText().toString(),
            Belirtininozellikleriedit.getText().toString(),Mudahaleislemleriedit.getText().to
            String());
        }
    }
}

```

Şekil 6.22. Ek bilgi kaydet java kodu

Yeni bilgi kaydedilerek ek bilgi ekranında gözükmetedir. Ancak ek bilgi kaydet işleminde belirti adı kısmı boş bırakıldığında herhangi bir yazı yazılmadığı durumda ek bilgi kaydet kontrol işlemi devreye girerek belirti adının girilmesi gerektiğini belirten uyarı mesajı ekrana çıkmaktadır. Şekil 6.23'de uyarı mesajı ekranı yer almaktadır.



Şekil 6.23. Belirti adı girilmesi gerektiğini belirten uyarı mesajı

Belirti adı boş bırakılmaması Şekil 6.24’de yer alan kontrol adımları ile sağlanmıştır.

```

OnMenuItemClickListener bilgikaydet = new
OnMenuItemClickListener() {
    public boolean onOptionsItemSelected(MenuItem item) {

        // Kaydetme işlevi
        if (Belirtiadiedit.getText().length() != 0) {
            AsyncTask<Object, Object, Object> notkaydetkontrol = new
            AsyncTask<Object, Object, Object>() {
                @Override
                protected Object doInBackground(Object... params) {
                    notkaydet();
                    return null;
                }

                @Override
                protected void onPostExecute(Object result) {
                    // activity kapalı
                    finish();
                }
            };

            // Yukarıdaki işlemi yürütme
            notkaydetkontrol.execute((Object[]) null);
        }
        else {
    
```



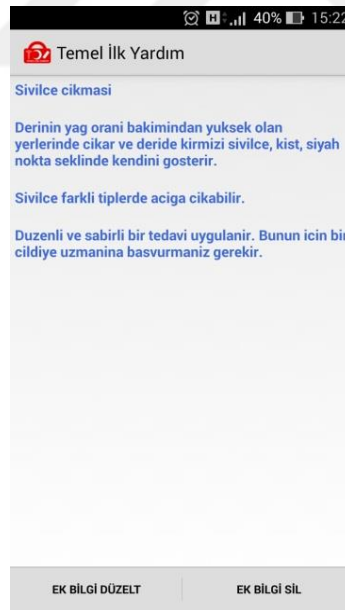
```

// Belirti adı boş bırakılmaması için uyarı mesajı ekrana gelmektedir.
AlertDialog.Builder uyari = new AlertDialog.Builder(notkaydet.this);
uyari.setTitle("Lütfen Eksik Alanları Doldurunuz!");
uyari.setMessage("Belirti Adı girmeniz gerekmektedir.");
uyari.setPositiveButton("Tamam", null);
uyari.show();
    }
    return false;
}
};

```

Şekil 6.24. Belirti adı boş bırakılmaması java kodu

Ek bilgi ekranında kaydedilen bilgilerin okunması, güncellenmesi ve silinmesi işlemi için istenen bilgi metni basılarak Şekil 6.25 ek bilgi oku, düzelt ve sil ekranı gelmektedir. Ekranın en alt kısmında yer alan ek bilgi düzelt ve ek bilgi sil butonları kullanılarak bilginin güncellenmesi veya silinmesi sağlanmaktadır.



Şekil 6.25. Ek bilgi oku, düzelt ve sil ekranı

Ek bilgi düzelt butonu kullanıldığında ek bilgi kaydet ekranı gelmekte ve ek bilgi kaydet butonu basılarak ek bilginin güncellenmesi sağlanmıştır. Ek bilgi oku, düzelt ve sil java kodu Şekil 6.26'da yer almaktadır.

```

package com.magellsoft.temelilkyardim;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.MenuItem.OnMenuItemClickListener;
import android.widget.TextView;

public class notgoruntule extends Activity {

    // Değişkenleri Belirle
    private long ID;
    private TextView Belirtiadi;
    private TextView Belirtinintanimi;
    private TextView Belirtininozellikleri;
    private TextView Mudahaleislemleri;
    private static final String BASLIK = "baslik";
    private static final String NOT1 = "not1";
    private static final String NOT2 = "not2";
    private static final String NOT3 = "not3";

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.notgoruntule);

        // notgoruntule.xml içinde TextView bul
        Belirtiadi = (TextView) findViewById(R.id.Belirtiadi);

        Belirtinintanimi = (TextView) findViewById(R.id.Belirtinintanimi);

        Belirtininozellikleri = (TextView) findViewById(R.id.Belirtininozellikleri);

        Mudahaleislemleri = (TextView) findViewById(R.id.Mudahaleislemleri);
    }
}

```

```

// yeninotekle.java Satır kimliği
    Bundle extras = getIntent().getExtras();
    ID = extras.getLong(yeninotekle.ID);
}
@Override
protected void onResume() {
    super.onResume();
    // notları yükle
    new notuyukle().execute(ID);
}

// notları yükle
private class notuyukle extends AsyncTask<Long, Object, Cursor> {
    // dbbaglayici.java class çağır
    veritabani dbbaglayici = new veritabani(notgoruntule.this);

    @Override
    protected Cursor doInBackground(Long... params) {
        // not arka planda çalıştır
        // dbbaglayici.java class
        dbbaglayici.ac();
        return dbbaglayici.notual(params[0]);
    }

    @Override
    protected void onPostExecute(Cursor sonuc) {
        super.onPostExecute(sonuc);

        sonuc.moveToFirst();
        // Her bir veri ögesi için sütun indeksi Al
        int BaslikIndex = sonuc.getColumnIndex(BASLIK);
        int Not1Index = sonuc.getColumnIndex(NOT1);
        int Not2Index = sonuc.getColumnIndex(NOT2);
        int Not3Index = sonuc.getColumnIndex(NOT3);

        //Metin içinde TextView ayarla
        Belirtiadi.setText(sonuc.getString(BaslikIndex));
        Belirtinintanimi.setText(sonuc.getString(Not1Index));
        Belirtininozellikleri.setText(sonuc.getString(Not2Index));
        Mudahaleislemleri.setText(sonuc.getString(Not3Index));

        sonuc.close();
        dbbaglayici.kapat();
    }
}

```

```

// Ek Bilgi Düzelt ve Ek Bilgi Sil seçenekler menüsü oluşturmak

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    menu.add("Ek Bilgi Düzelt")
        .setOnMenuItemClickListener(this.duzeltbuton)

    .setShowAsAction(MenuItem.SHOW_AS_ACTION_IF_ROOM);

    menu.add("Ek Bilgi Sil")
        .setOnMenuItemClickListener(this.silbuton)

    .setShowAsAction(MenuItem.SHOW_AS_ACTION_IF_ROOM);

    return super.onCreateOptionsMenu(menu);
}

// düzelt menüsü için tıklayınız.
OnMenuItemClickListener duzeltbuton = new OnMenuItemClickListener() {
    public boolean onMenuItemClick(MenuItem item) {

        // yeninotekle.java Row ID ve veri aktarmak
        Intent notekle = new Intent(notgoruntule.this, notkaydet.class);

        notekle.putExtra(yeninotekle.ID, ID);
        notekle.putExtra(BASLIK, Belirtiadi.getText());
        notekle.putExtra(NOT1, Belirtinintanimi.getText());
        notekle.putExtra(NOT2, Belirtininozellikleri.getText());
        notekle.putExtra(NOT3, Mudahaleislemleri.getText());
        startActivity(notekle);

        return false;
    }
};

// sil menüsü için tıklayınız.

OnMenuItemClickListener silbuton = new OnMenuItemClickListener() {
    public boolean onMenuItemClick(MenuItem item) {

        // sil komutunu çağır
        notusilkosulu();
        return false;
    }
};

```

```

private void notusilkosulu() {

    // Silme işlemini onaylamak için ekrana uyarı mesajı gelmektedir.
    AlertDialog.Builder uyari = new AlertDialog.Builder(notgoruntule.this);
    uyari.setTitle("Ek Bilgi Sil");
    uyari.setMessage("Gerçekten bu Ek Bilgiyi silmek istiyor musunuz.?");

    uyari.setPositiveButton("Evet", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int button) {
            final veritabani dbbaglayici = new veritabani(
                notgoruntule.this);
            AsyncTask<Long, Object, Object> notsil = new AsyncTask<Long, Object,
Object>() {

                @Override
                protected Object doInBackground(Long... params) {
                    // sil işlemini gerçekleştir.
                    // dbbaglayici.java
                    dbbaglayici.notsil(params[0]);
                    return null;
                }

                @Override
                protected void onPostExecute(Object sonuc) {
                    // işlemi kapat
                    finish();
                }
            };

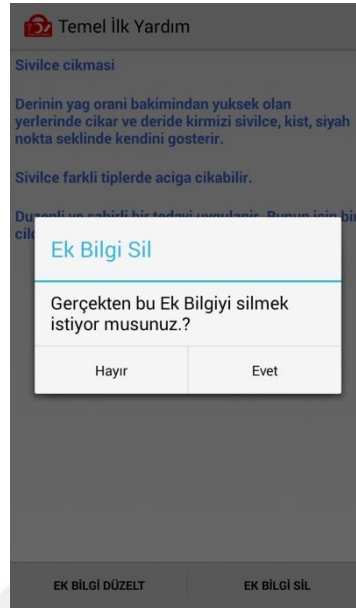
            // Yukarıdaki silme Görev işlemini Yürüt
            notsil.execute(new Long[] { ID });
        }
    });

    //hiçbir işlem yapma
    uyari.setNegativeButton("Hayır", null).show();
}
}

```

Şekil 6.26. Ek bilgi oku, düzelt ve sil java kodu

Ek bilgi sil butonu kullanıldığında Şekil 6.27’de yer alan “Gerçekten bu Ek Bilgiyi silmek istiyor musunuz.?” ek bilgi sil uyarı mesajı gelmektedir. Hayır butonu kullanıldığında herhangi bir işlem yapılmamaktadır. Evet butonu kullanıldığında ek bilgi silme işlemi yapılmaktadır.



Şekil 6.27. Ek bilgi sil uyarı mesajı

Şekil 6.28’de ek bilgi sil uyarı mesajı java kodu yer almaktadır.

```
private void notusilkosulu() {

    // Silme işlemini onaylamak için ekrana uyarı mesajı gelmektedir.
    AlertDialog.Builder uyari = new
    AlertDialog.Builder(notgoruntule.this);
    uyari.setTitle("Ek Bilgi Sil");
    uyari.setMessage("Gerçekten bu Ek Bilgiyi silmek istiyor musunuz?");

    uyari.setPositiveButton("Evet", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int button) {
            final veritabani dbbaglayici = new veritabani(
                notgoruntule.this);

            AsyncTask<Long, Object, Object> notsil = new AsyncTask<Long, Object,
            Object>() {

                @Override
                protected Object doInBackground(Long... params) {
                    // sil işlemini gerçekleştir.
                    // dbbaglayici.java
                    dbbaglayici.notsil(params[0]);
                    return null;
                }
            }
        }
    });
}
```

```

@Override
protected void onPostExecute(Object sonuc) {
    // işlemi kapat
    finish();
}

};
// Yukarıdaki silme Görev işlemini Yürüt
notsil.execute(new Long[] { ID });
});

//hiçbir işlem yapma
uyari.setNegativeButton("Hayır", null).show();
}
}

```

Şekil 6.28. Ek bilgi sil uyarı mesajı java kodu

Ek bilgi ekranında verilerin tutulması için SQLite Database yapısından yararlanılmıştır. SQLite mobil cihazlarda kullanılan SQL veritabanı motorunu uygulayan kendi kendine yeten sunucu kullanmayan sıkıştırılmış bir işlem kütüphanesidir (About SQLite).

Veri tabanı yapısında “veritabanı” ve “veritabanıyardimci” java dosyasından oluşmaktadır. Burada veri tabanı ismi, tablo adı, değişkenlerin tanımı ve database versiyonu “1” yapılarak veri tabanı oluşturulmuştur. Ek bilgi uygulamasında yer alan belirti adı, belirtinin tanımı, belirtinin özellikleri ve müdahale işlemleri’ne yazılan yazıların baslik, not1, not2 ve not3 String olarak veritabanına kaydedilmektedir. Kaydedilen verinin liste ekranında sadece baslik kısmı gözükmektedir (SQLiteDatabase). Şekil 6.29 veritabanı java kodu yer almaktadır.

```

package com.magellsoft.temelilkyardim;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;

public class veritabani {

    // Değişkenleri Belirle
    private static final String DB_ADI = "Notlar";
    private static final String TABLO_ADI = "tablonot";
    private static final String BASLIK = "baslik";
    private static final String ID = "_id";
    private static final String NOT1 = "not1";
    private static final String NOT2 = "not2";
    private static final String NOT3 = "not3";
    private static final int DATABASE_VERSION = 1;
    private SQLiteDatabase veritabani;
    private veritabaniyardimci dbyardimci;

    public veritabani(Context context) {
        dbyardimci = new veritabaniyardimci(context, DB_ADI, null,
            DATABASE_VERSION);
    }

    // Veritabanı fonksiyonu çalıştır
    public void ac() throws SQLException {
        // Veritabanı yazılabilir modda olmasına izin ver
        veritabani = dbyardimci.getWritableDatabase();
    }

    // Veritabanı fonksiyonu kapat
    public void kapat() {
        if (veritabani != null)
            veritabani.close();
    }

    // Veritabanı fonksiyonu yarat
    public void notekle(String baslik, String not1, String not2, String not3) {
        ContentValues yeninot = new ContentValues();
        yeninot.put(BASLIK, baslik);
        yeninot.put(NOT1, not1);
        yeninot.put(NOT2, not2);
        yeninot.put(NOT3, not3);
    }
}

```



```

        ac();
        veritabani.insert(TABLO_ADI, null, yeninot);
        kapat();
    }
    // Veritabanı fonksiyonu güncelle
    public void notguncelle(long id, String baslik, String not1,String not2,String
not3) {
        ContentValues guncellenot = new ContentValues();
        guncellenot.put(BASLIK, baslik);
        guncellenot.put(NOT1, not1);
        guncellenot.put(NOT2, not2);
        guncellenot.put(NOT3, not3);

        ac();
        veritabani.update(TABLO_ADI, guncellenot, ID + "=" + id, null);
        kapat();
    }
    // Veritabanı fonksiyonu sil
    public void notsil(long id) {
        ac();
        veritabani.delete(TABLO_ADI, ID + "=" + id, null);
        kapat();
    }

    // Tüm veriler fonksiyonu Listele
    public Cursor tumnotlarilistele() {
        return veritabani.query(TABLO_ADI, new String[] { ID, BASLIK },
null,
        null, null, null, BASLIK);
    }
    // ID verileri aktar
    public Cursor notual(long id) {
        return veritabani.query(TABLO_ADI, null, ID + "=" + id, null, null,
null, null);
    }
}

```

Şekil 6.29. Veritabanı java kodu

“Veritabanıyardimci” java dosyasında SQLiteOpenHelper sınıfında onCreate() ve onUpgrade() metodları kullanılmıştır. onCreate – metod veritabanı oluşturmak için kullanılmaktadır. onUpgrade – metod veritabanı güncellemek için kullanılmaktadır (SQLiteOpenHelper). Şekil 6.30 veritabanıyardimci java kodu yer almaktadır.

```

package com.magellsoft.temelilkyardim;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteDatabase.CursorFactory;
import android.database.sqlite.SQLiteOpenHelper;

public class veritabaniyardimci extends SQLiteOpenHelper {

    // Değişkenleri Belirle
    private static final String DB_ADI = "Notlar";
    public static final String TABLO_ADI = "tablonot";
    public static final String BASLIK = "baslik";
    public static final String NOT1 = "not1";
    public static final String NOT2 = "not2";
    public static final String NOT3 = "not3";

    public veritabaniyardimci(Context context, String name, CursorFactory factory,
        int version) {
        super(context, DB_ADI, factory, version);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        // database tablo yarat
        String tabloyarat = "CREATE TABLE " + TABLO_ADI+ " (_id integer primary key
        autoincrement," + BASLIK + ", "+ NOT1 + ", "+ NOT2+ ", "+ NOT3+");";
        db.execSQL(tabloyarat);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // Veritabanı güncelle
        db.execSQL("DROP TABLE IF EXISTS " + TABLO_ADI);
        onCreate(db);
    }
}

```

Şekil 6.30. Veritabaniyardimci java kodu

6.4. Eğitim Süreci

Eğitim Süreci bölümü; Temel İlk Yardım eğitimi alacak kişiler için eğitim sürecinin, sınav adımlarının ve sertifika sürecinin kısaca anlatıldığı bölümdür. Sayfa yapısı LinearLayout kullanılarak resim ve yazı kısımlarından oluşmaktadır. ScrollView

özelliği ile aşağı/yukarı yapılarak yazının okunması sağlanmıştır. Şekil 6.31’de eğitim süreci ekranı gösterilmektedir.



Şekil 6.31. Eğitim süreci ekranı

Temel ilk yardımcı sertifikası nasıl alınır? Bilgi ekranı Şekil 6.32’deki TextView yapısı kullanılarak yazılmıştır.

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

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

    <ScrollView
        android:id="@+id/scrollView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
```

```

android:orientation="vertical" >

<ImageView
    android:id="@+id/imageView1"
    android:layout_width="wrap_content"
    android:layout_height="200dp"
    android:src="@drawable/tab2resim" />

<TextView
    android:id="@+id/TextView01"
    android:layout_width="match_parent"
    android:layout_height="5dp" />

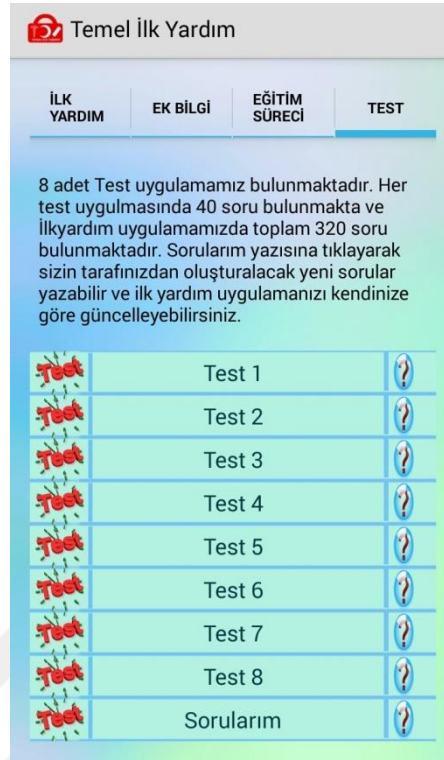
<TextView
    android:id="@+id/TextView11"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="Temel İkyardımcı Sertifikası nasıl alınır?"
    android:textColor="#4169e1"
    android:textSize="20sp"
    android:textStyle="bold"
    android:layout_marginRight="7dp"
    android:layout_marginLeft="7dp"

```

Şekil 6.32. Eğitim süreci xml kodu

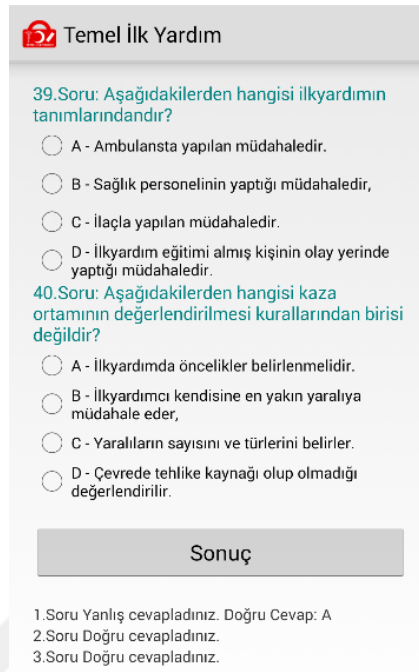
6.5. Test

Test bölümünde sayfa yapısı LinearLayout kullanılarak resim, yazı ve buton kısımlarından oluşmaktadır. ScrollView özelliği ile aşağı/yukarı kullanılarak diğer test butonlarının görülmesi sağlanmaktadır. Şekil 6.33’de test ekranında toplam 8 adet test uygulaması ve sorularım bölümü bulunmaktadır.



Şekil 6.33. Test ekranı

Test butonlarına basıldığında her test uygulamasında farklı soruların yazıldığı 40 soru ekrana gelmektedir ve 60 dk. test süresi başlamaktadır. Test uygulamasında yer alan sonuç butonuna basılarak doğru/yanlış cevaplar görüntülenir ve test süresi zamanı durdurulur. Test soruları TextView özelliği ile yazılmıştır. Soru seçeneği RadioGroup altında her soru şıkkı için Radio butonlarına numara verilerek A, B, C, D şıkları yazılmıştır. Şekil 6.34 test soru ve cevap ekranı yer almaktadır.



Şekil 6.34. Test soru ve cevap ekranı

Şekil 6.35’de test soru seçeneği xml kodu yer almaktadır.

```

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="15sp"
    android:layout_marginRight="7dp"
    android:layout_marginLeft="7dp"
    android:textColor="#008080"
    android:text="1.Soru: Bak-dinle-hisset yöntemi ile yapılan solunum
değerlendirilmesi kaç saniye süre yapılır?" />
<RadioGroup
    android:id="@+id/radioGroup1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >

    <RadioButton
        android:id="@+id/radio1"
        style="android:radioButtonStyle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="#000000"
        android:textSize="13sp"
        android:layout_marginRight="7dp"
        android:layout_marginLeft="7dp"
        android:text="A - 10 saniye" />
  
```

```

<RadioButton
    android:id="@+id/radio2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="#000000"
    android:textSize="13sp"
    android:layout_marginRight="7dp"
    android:layout_marginLeft="7dp"
    android:text="B - 5 saniye" />

<RadioButton
    android:id="@+id/radio3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="#000000"
    android:textSize="13sp"
    android:layout_marginRight="7dp"
    android:layout_marginLeft="7dp"
    android:text="C - 30 saniye" />

<RadioButton
    android:id="@+id/radio4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="#000000"
    android:textSize="13sp"
    android:layout_marginRight="7dp"
    android:layout_marginLeft="7dp"
    android:text="D - 1 saniye" />
</RadioGroup>

```

Şekil 6.35. Test soru seçeneği xml kodu

Şekil 6.36'da cevap ve süre durdurma java kodu yer almaktadır.

```

// sonuc
sonuc.setOnClickListener (new View.OnClickListener() {

    @Override
    public void onClick(View v) {
//sonuc butona tıklandığında zaman sayacını durdur ve cevapları göster
        if (zaman != null) {
            zaman.cancel();
            zaman = null;
        }
    }
}

```

```

int soru1=rg1.getCheckedRadioButtonId();
String metin1="";

switch (soru1){

case R.id.radio1:{metin1=metin1+"1.Soru Doğru cevapladınız.";break;}

case R.id.radio2:{metin1=metin1+"1.Soru Yanlış cevapladınız. Doğru Cevap:
A";break;}

case R.id.radio3:{metin1=metin1+"1.Soru Yanlış cevapladınız. Doğru Cevap:
A";break;}

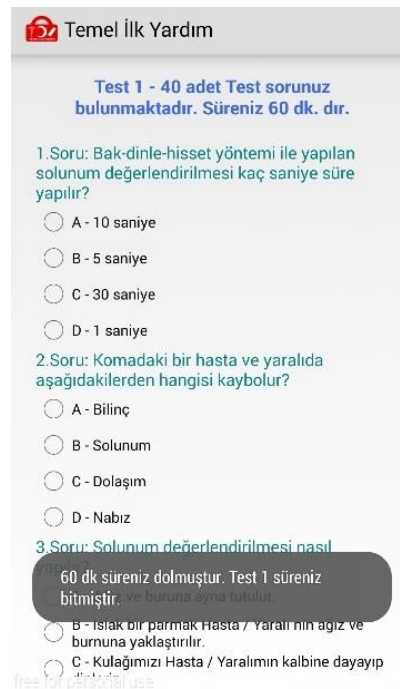
case R.id.radio4:{metin1=metin1+"1.Soru Yanlış cevapladınız. Doğru Cevap:
A";break;}

}

```

Şekil 6.36. Cevap java kodu

Sonuç butonuna basılmadığı durumda test süresi zamanı dolduğu anda ekrana; “60 dk süreniz dolmuştur. Test süreniz bitmiştir.” mesajı Toast özelliği kullanılarak üç kere ekranda görüntülenecektir. Şekil 6.37’de toast mesajı görüntülenmektedir.



Şekil 6.37. “60 dk süreniz dolmuştur. Test süreniz bitmiştir.” mesajı

Şekil 6.38’de 60 dk. zaman ayarlı test zamanlayıcı java kodu yer almaktadır.

```

public class test1 extends Activity {

    Timer zaman;
    TimerTask sayac;

    //TimerTask çalıştırmak
    final Handler toast = new Handler();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.test1);
    }

    @Override
    protected void onResume() {
        super.onResume();

        //arka planda zamanlayıcıyı çalıştırıyoruz.
        startTimer();
    }

    public void startTimer() {
        //Yeni Zamanlayıcı ayarlama
        zaman = new Timer();

        //Zamanlayıcı Görev işi başlat
        zamanlayicisureyiyarla();

        //zamanlayıcı 3600000ms.(60 dk.) ulaştıktan sonra ekrana toast uyarısı ver.
        zaman.schedule(sayac, 3600000); //
    }

    public void zamanlayicisureyiyarla() {

        sayac = new TimerTask() {
            public void run() {

                //Toast çalıştırmak için geçerli zamanlayıcıyı çalıştır.
                toast.post(new Runnable() {
                    public void run() {

                        // Geçerli zaman yazısı
                        Calendar takvim = Calendar.getInstance();
                        SimpleDateFormat zamanformat = new SimpleDateFormat("HH:mm");
                        final String baslazaman = zamanformat.format(takvim.getTime());
                    }
                });
            }
        };
    }
}

```

```
// toast göster
    int toastgoster = Toast.LENGTH_LONG;

    Toast toast = Toast.makeText(getApplicationContext(), baslazaman, toastgoster);
    toast.show();

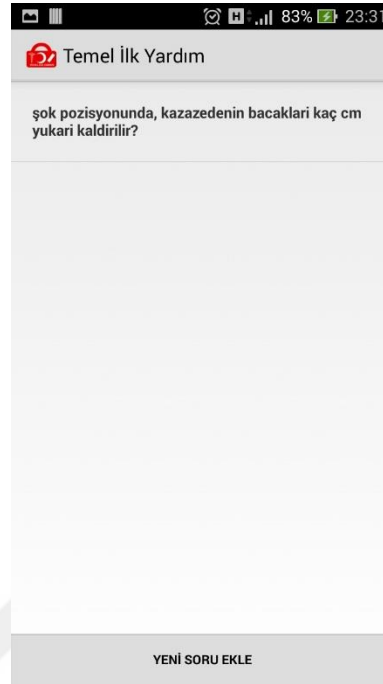
    Toast.makeText(getApplicationContext(), "60 dk süreniz dolmuştur. Test 1 süreniz
    bitmiştir.", Toast.LENGTH_LONG).show();

    Toast.makeText(getApplicationContext(), "60 dk süreniz dolmuştur. Test 1 süreniz
    bitmiştir.", Toast.LENGTH_LONG).show();

    Toast.makeText(getApplicationContext(), "60 dk süreniz dolmuştur. Test 1 süreniz
    bitmiştir.", Toast.LENGTH_LONG).show();
}
});
};
```

Şekil 6.38. Test zamanlayıcı java kodu

Test ekranında yer alan “Sorularım” kısmında uygulamayı kullanacak kişi tarafından ilk yardımla ilgili yeni test soruları yazılabilmektedir. Sorularım butonuna basıldığında Şekil 6.39’da kaydedilen sorular ekranı gelmekte ve yeni soru yazmak için ekranın en altında yer alan yeni soru ekle butonu kullanılmaktadır.



Şekil 6.39. Sorularım ekranı ve yeni soru ekle butonu

Sorularım butonuna basıldığında Şekil 6.40'da yer alan java kodu çalışmaktadır.

```

package com.magellsoft.temelilkyardim;

import android.app.ListActivity;
import android.content.Intent;
import android.database.Cursor;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.MenuItem.OnMenuItemClickListener;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.CursorAdapter;
import android.widget.ListView;
import android.widget.SimpleCursorAdapter;

public class soruekle extends ListActivity {

    // Değişkenleri Belirle
    public static final String ID = "id";
    private static final String SORU = "soru";
    private ListView sorulistegoruntule;

```

```

private CursorAdapter sorubaglayici;

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    // ListView göster
    sorulistegoruntule = getListView();

    // Liste görünümü
    sorulistegoruntule.setOnItemClickListener(sorusirala);

    // Bu ViewTitleNotes tüm başlıkları TextView göster
    String[] soru = new String[] { SORU };
    int[] sorug = new int[] { R.id.Sorularigoster };

    // Basit İmleç Adaptörü oluştur
    sorubaglayici = new SimpleCursorAdapter(soruekle.this,
        R.layout.soruliste, null, soru, sorug);

    // Basit İmleç Adaptörü içine Adaptörü ayarla
    setListAdapter(sorubaglayici);
}

// soru sirala
OnItemClickListener sorusirala = new OnItemClickListener() {
    public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
        long arg3) {

        // sorugoruntule activity çalıştır
        Intent sorugosterbuton = new Intent(soruekle.this, sorugoruntule.class);

        // soru aktivitesini Görüntülemek için SATIR kimliğini ilet
        sorugosterbuton.putExtra(ID, arg3);
        startActivity(sorugosterbuton);
    }
};

@Override
protected void onResume() {
    super.onResume();

    //soru ekle kısmına geri gel

        new sorual().execute((Object[]) null);
}

```

```

@Override
protected void onStop() {
    Cursor imlec = sorubaglayici.getCursor();

    // İmleç devre dışı bırakır
    if (imlec != null)
        imlec.deactivate();

    sorubaglayici.changeCursor(null);
    super.onStop();
}

// Yeni Soru Ekle option menu oluştur

@Override
public boolean onCreateOptionsMenu(Menu menu) {

    // Yeni Soru Ekle Menu baslik

    menu.add("Yeni Soru Ekle")
        .setOnMenuItemClickListener(this.yenisoruueklelistesi)

    .setShowAsAction(MenuItem.SHOW_AS_ACTION_IF_ROOM);

    return super.onCreateOptionsMenu(menu);
}

// soru ekle için tıkla
OnMenuItemClickListener yenisoruueklelistesi = new
OnMenuItemClickListener() {
    public boolean onOptionsItemSelected(MenuItem item) {

        // sorukaydet activity aç
        Intent sorueklebuton = new Intent(soruuekle.this,
sorukaydet.class);
        startActivity(soruueklebuton);

        return false;
    }
};

private class sorual extends AsyncTask<Object, Object, Cursor> {
    veritabanisoru dbbaglayici = new veritabanisoru(soruuekle.this);

    @Override
    protected Cursor doInBackground(Object... params) {

```

```
// dbbaglayici aç
    dbbaglayici.ac();

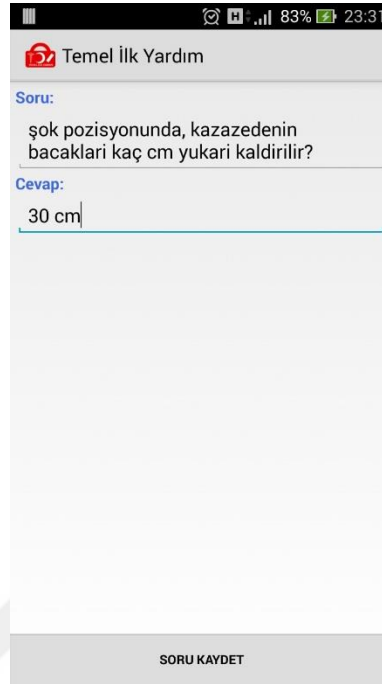
    return dbbaglayici.tumsorularilistele();
}

@Override
protected void onPostExecute(Cursor islem) {
    sorubaglayici.changeCursor(islem);

    // dbbaglayici kapat
    dbbaglayici.kapat();
}
}
```

Şekil 6.40. Sorularım java kodu

Yeni soru ekle butonuna basıldığında EditText yapısı kullanılarak Şekil 6.41’de yer alan soru ve cevap kısımlarından oluşan yazı yazma ekranı gelmektedir. Gerekli alanlar doldurulduktan sonra soru kaydet butonuna basılarak soru kaydedilmekte ve sorularım ekranında gözükmemektedir.



Şekil 6.41. Soru kaydet ekranı

Soru kaydet java kodu Şekil 6.42’de yer almaktadır.

```
package com.magellsoft.temelilkyardim;

import android.app.Activity;
import android.app.AlertDialog;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.MenuItem.OnMenuItemClickListener;
import android.widget.EditText;

public class sorukaydet extends Activity {
    // Değişkenleri Belirle
    private long ID;
    private EditText Soruadiedit;
    private EditText Cevaptanimiedit;
    private static final String SORU = "soru";
    private static final String CEVAP = "cevap";

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.soru);
    }
}
```

```

// soruekle.xml içinde EditText bul
Soruadiedit = (EditText) findViewById(R.id.Soruadiedit);
Cevaptanimiedit = (EditText) findViewById(R.id.Cevaptanimiedit);

// sorugoruntule.java gelen Satır kimliği Al
Bundle extras = getIntent().getExtras();
if (extras != null) {
    ID = extras.getLong("id");
    Soruadiedit.setText(extras.getString(SORU));
    Cevaptanimiedit.setText(extras.getString(CEVAP));
}
}
// Soru kaydet seçenek menüsü oluştur
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    menu.add("Soru Kaydet")
        .setOnMenuItemClickListener(this.sorukaydet)

        .setShowAsAction(MenuItem.SHOW_AS_ACTION_IF_ROOM);

    return super.onCreateOptionsMenu(menu);
}
// menüsü için tıklayınız.
OnMenuItemClickListener sorukaydet = new OnMenuItemClickListener() {
    public boolean onMenuItemClick(MenuItem item) {

        // Kaydetme işlevi
        if (Soruadiedit.getText().length() != 0) {
            AsyncTask<Object, Object, Object> sorukaydetkontrol = new
            AsyncTask<Object, Object, Object>() {

                @Override
                protected Object doInBackground(Object... params) {
                    sorukaydet();
                    return null;
                }
                @Override
                protected void onPostExecute(Object result) {
                    // activity kapalı
                    finish();
                }
            };
            // Yukarıdaki işlemi yürüt
            sorukaydetkontrol.execute((Object[]) null);
        }
        else {

```



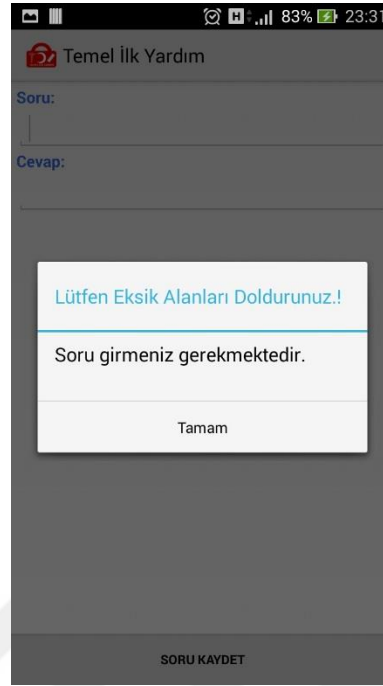
```

// Soru alanı boş bırakılmaması için uyarı ekranı görüntülenir.
AlertDialog.Builder uyari = new AlertDialog.Builder(sorukaydet.this);
    uyari.setTitle("Lütfen Eksik Alanları Doldurunuz!");
    uyari.setMessage("Soru girmeniz gerekmektedir.");
    uyari.setPositiveButton("Tamam", null);
    uyari.show();
        }
        return false;
    }
};
// soru kaydet fonksiyonu
private void sorukaydet() {
    veritabanisoru dbbaglayici = new veritabanisoru(this);
    if (getIntent().getExtras() == null) {
// Veri dbbaglayici.java içinde soru ekle
        dbbaglayici.soruekle(Soruadiedit.getText().toString(),
            Cevaptanimiedit.getText().toString());
    } else {
// Veri dbbaglayici.java içinde sorularıgüncelle ekle
        dbbaglayici.soruguncelle(ID,
            Soruadiedit.getText().toString(),Cevaptanimiedit.getText().toString());
    } }
}

```

Şekil 6.42. Soru kaydet java kodu

Ancak soru kısmı boş bırakıldığında herhangi bir yazı yazılmadığı durumda soru kaydet kontrol işlemi devreye girerek soru girmemiz gerektiğini belirten bir uyarı mesajı ekrana çıkartmaktadır. Şekil 6.43’de uyarı mesajı gösterilmektedir.



Şekil 6.43. Soru girmemiz gerektiğini belirten uyarı mesajı

Soru kısmını boş bırakılmaması Şekil 6.44’de yer alan kontrol adımları ile sağlanmıştır.

```

OnMenuItemClickListener sorukaydet = new OnMenuItemClickListener() {
    public boolean onOptionsItemSelected(MenuItem item) {

        // Kaydetme işlevi

        if (Soruedit.getText().length() != 0) {
            AsyncTask<Object, Object, Object> sorukaydetkontrol = new
            AsyncTask<Object, Object, Object>() {
                @Override
                protected Object doInBackground(Object... params) {
                    sorukaydet();
                    return null;
                }

                @Override
                protected void onPostExecute(Object result) {
                    // activity kapalı

                    finish();
                }
            };
        }
    }
};

```

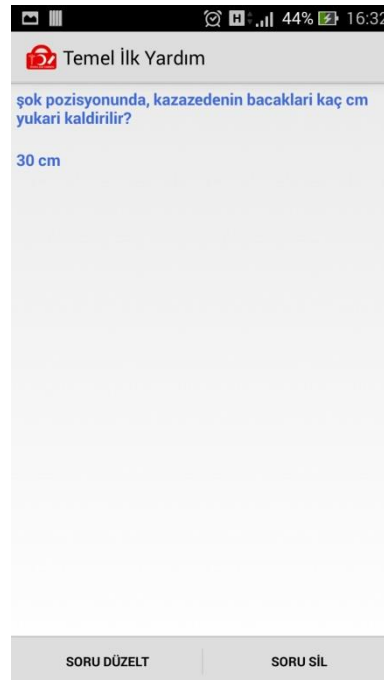
```

// Yukarıdaki işlemi yürüt
        sorukaydetkontrol.execute((Object[]) null);
    }
    else {
        // soru eksiksiz doldurmak için uyarı iletişim mesajı görüntüler
        AlertDialog.Builder uyari = new AlertDialog.Builder(sorukaydet.this);
        uyari.setTitle("Lütfen Eksik Alanları Doldurunuz.!");
        uyari.setMessage("Soru girmeniz gerekmektedir.");
        uyari.setPositiveButton("Tamam", null);
        uyari.show();
    }
        return false;
    }
};

```

Şekil 6.44. Soru kaydet - soru veri kontrol java kodu

Sorularım ekranında kaydedilen bilgilerin okunması, güncellenmesi ve silinmesi işlemi için istenen soru metni basılarak Şekil 6.45 soru oku, düzelt ve sil ekranı gelmektedir. Ekranın en alt kısmında yer alan soru düzelt ve soru sil butonları kullanılarak bilginin güncellenmesi veya silinmesi sağlanmaktadır.



Şekil 6.45. Soru oku, düzelt ve sil ekranı

Soru düzelt butonu kullanıldığında soru kaydet ekranı gelmekte ve soru kaydet butonu basılarak sorunun güncellenmesi sağlanmıştır. Şekil 6.46'da soru oku, düzelt ve sil java kodu yer almaktadır.

```

package com.magellsoft.temelilkyardim;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.MenuItem.OnMenuItemClickListener;
import android.widget.TextView;

public class sorugoruntule extends Activity {

    // Değişkenleri Belirle
    private long ID;
    private TextView Soruadi;
    private TextView Cevaptanimi;
    private static final String SORU = "soru";
    private static final String CEVAP = "cevap";

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.sorugoruntule);

        // sorugoruntule.xml içinde TextView bul
        Soruadi = (TextView) findViewById(R.id.Soruadi);
        Cevaptanimi = (TextView) findViewById(R.id.Cevaptanimi);

        // soruekle.java Satır kimliği
        Bundle extras = getIntent().getExtras();
        ID = extras.getLong(soruekle.ID);
    }

    @Override
    protected void onResume() {
        super.onResume();
    }
}

```

```

        // soru ekle yükle
        new soruyukle().execute(ID);
    }
    // soruları yükle
    private class soruyukle extends AsyncTask<Long, Object, Cursor> {
    // dbbaglayici.java class çağır
        veritabanisoru dbbaglayici = new veritabanisoru(sorugoruntule.this);

        @Override
        protected Cursor doInBackground(Long... params) {
            // soru arka planda çalışma
            // dbbaglayici.java class
            dbbaglayici.ac();
            return dbbaglayici.sorual(params[0]);
        }

        @Override
        protected void onPostExecute(Cursor sonuc) {
            super.onPostExecute(sonuc);

            sonuc.moveToFirst();
            // Her bir veri ögesi için sütun indeksi Al
            int soruIndex = sonuc.getColumnIndex(SORU);
            int CevapIndex = sonuc.getColumnIndex(CEVAP);

            //Metin içinde TextView ayarla
            Soruadi.setText(sonuc.getString(soruIndex));
            Cevaptanimi.setText(sonuc.getString(CevapIndex));

            sonuc.close();
            dbbaglayici.kapat();
        }
    }

    // soruyu Düzelt ve soruyu Sil seçenekler menüsü oluşturmak
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        menu.add("Soru Düzelt")
            .setOnMenuItemClickListener(this.duzeltbuton)

            .setShowAsAction(MenuItem.SHOW_AS_ACTION_IF_ROOM);

        menu.add("Soru Sil")
            .setOnMenuItemClickListener(this.silbuton)

            .setShowAsAction(MenuItem.SHOW_AS_ACTION_IF_ROOM);
    }

```

```

        return super.onCreateOptionsMenu(menu);
    }

    // düzelt menüsü için tıklayınız.
    OnMenuItemClickListener duzeltbuton = new OnMenuItemClickListener() {
        public boolean onOptionsItemSelected(MenuItem item) {

            // soruekle.java Row ID ve veri aktar
            Intent soruyuekle = new Intent(sorugoruntule.this, sorukaydet.class);

            soruyuekle.putExtra(soruekle.ID, ID);
            soruyuekle.putExtra(SORU, Soruadi.getText());
            soruyuekle.putExtra(CEVAP, Cevaptanimi.getText());

            startActivity(soruyuekle);

            return false;
        }
    };

    // sil menüsü için tıklayınız.
    OnMenuItemClickListener silbuton = new OnMenuItemClickListener() {
        public boolean onOptionsItemSelected(MenuItem item) {

            // sil komutunu çağır
            sorusilkosulu();

            return false;
        }
    };

    private void sorusilkosulu() {

        // Silme işlemini onaylamak için uyarı mesajı ekrana gelir.
        AlertDialog.Builder uyari = new AlertDialog.Builder(sorugoruntule.this);
        uyari.setTitle("Soru Sil");
        uyari.setMessage("Gerçekten bu Soruyu silmek istiyor musunuz.?");
        uyari.setPositiveButton("Evet", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int button) {
                final veritabanisoru dbbaglayici = new veritabanisoru(sorugoruntule.this);
                AsyncTask<Long, Object, Object> sorusil = new AsyncTask<Long, Object,
                Object>() {
                    @Override
                    protected Object doInBackground(Long... params) {
                        // sil işlemini gerçekleştir.
                        // dbbaglayici.java

```

```

        dbbaglayici.sorusil(params[0]);
        return null;
    }

    @Override
    protected void onPostExecute(Object sonuc) {
        // işlemi kapat
        finish();
    }
};
// Yukarıdaki silme Görev işlemini Yürüt
sorusil.execute(new Long[] { ID });
}
});
//hiçbir işlem yapma
uyari.setNegativeButton("Hayır", null).show();
}
}

```

Şekil 6.46. Soru oku, düzelt ve sil java kodu

Soru sil butonu kullanıldığında Şekil 6.47’de yer alan “Gerçekten bu Soruyu silmek istiyor musunuz.?” soru sil uyarı mesajı ekranı gelmektedir. Hayır ve Evet butonları bulunmaktadır. Hayır butonu kullanıldığında herhangi bir işlem yapılmamaktadır. Evet butonu kullanıldığında soru silme işlemi yapılmaktadır.



Şekil 6.47. Soru sil uyarı mesajı

Şekil 6.48'de soru sil uyarı mesajı java kodu yer almaktadır.

```

private void sorusilkosulu() {
// Silme işlemini onaylamak için uyarı mesaj kutusu gelir.
    AlertDialog.Builder uyari = new
AlertDialog.Builder(sorugoruntule.this);
    uyari.setTitle("Soru Sil");
    uyari.setMessage("Gerçekten bu Soruyu silmek istiyor musunuz?");

    uyari.setPositiveButton("Evet", new
DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int button) {
final veritabanisoru dbbaglayici = new veritabanisoru(sorugoruntule.this);

            AsyncTask<Long, Object, Object> sorusil = new
AsyncTask<Long, Object, Object>() {
                @Override
                protected Object doInBackground(Long... params) {
// dbbaglayici.java
                    dbbaglayici.sorusil(params[0]);
                    return null;
                }
                @Override
                protected void onPostExecute(Object sonuc) {
// işlemi kapat
                    finish();
                }
            };
// Yukarıdaki silme Görev işlemini Yürüt
            sorusil.execute(new Long[] { ID });
        }
    });

//hiçbir işlem yapma
    uyari.setNegativeButton("Hayır", null).show();
}
}

```

Şekil 6.48. Soru sil uyarı mesajı java kodu

Test sorularım ekranında verilerin tutulması için SQLite Database yapısından yararlanılmıştır. SQLite mobil cihazlarda kullanılan SQL veritabanı motoru uygulayan bir işlem kütüphanesidir.

Veri tabanı yapısında “veritabanisoru” ve “veritabanisoruyardimci” java dosyasından oluşmaktadır. Burada veri tabanı ismi, tablo adı, değişkenlerin tanımı ve database versiyonu “1” yapılarak veri tabanı oluşturulmuştur. Sorularım uygulamasında yer alan soru ve cevap kısımlarına yazılan yazılar String olarak veritabanına kaydedilmektedir. Kaydedilen verinin liste ekranında sadece baslik kısmı gözükmektedir (SQLiteDatabase). Şekil 6.49 veritabanisoru java kodu yer almaktadır.

```
package com.magellsoft.temelilkyardim;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;

public class veritabanisoru {

    // Değişkenleri Belirle
    private static final String DB_ADI = "Sorular";
    private static final String TABLO_ADI = "tablosoru";
    private static final String SORU = "soru";
    private static final String ID = "_id";
    private static final String CEVAP = "cevap";

    private static final int DATABASE_VERSION = 1;
    private SQLiteDatabase veritabani;
    private veritabanisoruyardimci dbyardimci;

    public veritabanisoru(Context context) {
        dbyardimci = new veritabanisoruyardimci(context, DB_ADI, null,
            DATABASE_VERSION);
    }
}
```

```

// Veritabanı fonksiyonu çalıştır
public void ac() throws SQLException {
    // Veritabanı yazılabilir modda olmasına izin ver
    veritabani = dbyardimci.getWritableDatabase();
}

// Veritabanı fonksiyonu kapat
public void kapat() {
    if (veritabani != null)
        veritabani.close();
}

// Veritabanı fonksiyonu yarat
public void soruekle(String soru, String cevap) {
    ContentValues yenisoru = new ContentValues();
    yenisoru.put(SORU, soru);
    yenisoru.put(CEVAP, cevap);

    ac();
    veritabani.insert(TABLO_ADI, null, yenisoru);
    kapat();
}

// Veritabanı fonksiyonu güncelle

public void soruguncelle(long id, String soru, String cevap) {
    ContentValues guncellesoru = new ContentValues();
    guncellesoru.put(SORU, soru);
    guncellesoru.put(CEVAP, cevap);

    ac();
    veritabani.update(TABLO_ADI, guncellesoru, ID + "=" + id, null);
    kapat();
}

// Veritabanı fonksiyonu sil
public void sorusil(long id) {
    ac();
    veritabani.delete(TABLO_ADI, ID + "=" + id, null);
    kapat();
}

// Tüm veriler fonksiyonu Liste
public Cursor tumsorularilistele() {
    return veritabani.query(TABLO_ADI, new String[] { ID, SORU }, null,
        null, null, null, SORU);
}

```

```

// ID verileri aktar
public Cursor sorual(long id) {
    return veritabani.query(TABLO_ADI, null, ID + "=" + id, null, null,
        null, null);
}
}

```

Şekil 6.49. Veritabanisoru java kodu

“Veritabanisoruyardimci” java dosyasında SQLiteOpenHelper sınıfında onCreate() ve onUpgrade() metodları kullanılmıştır. onCreate – metod veritabanı oluşturmak için kullanılmaktadır. onUpgrade – metod veritabanı güncellemek için kullanılmaktadır (SQLiteOpenHelper). Şekil 6.50 veritabanisoruyardimci java kodu yer almaktadır.

```

package com.magellsoft.temelilkyardim;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteDatabase.CursorFactory;
import android.database.sqlite.SQLiteOpenHelper;

public class veritabanisoruyardimci extends SQLiteOpenHelper {

    // Değişkenleri Belirle
    private static final String DB_ADI = "Sorular";
    public static final String TABLO_ADI = "tablosoru";
    public static final String SORU = "soru";
    public static final String CEVAP = "cevap";

    public veritabanisoruyardimci(Context context, String name, CursorFactory
factory, int version) {
        super(context, DB_ADI, factory, version);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        // database tablo yarat
        String tabloyarat = "CREATE TABLE " + TABLO_ADI+ " (_id integer primary key
autoincrement," + SORU + ", "+ CEVAP + ");";
        db.execSQL(tabloyarat);
    }
}

```

```
}  
  
@Override  
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
    // Veritabanı sürüm değişikliği silinecek  
    db.execSQL("DROP TABLE IF EXISTS " + TABLO_ADI);  
    onCreate(db);  
}  
}
```

Şekil 6.50. Veritabanisoruyardimci java kodu



VII. TEMEL İLK YARDIM UYGULAMASININ YAYINLANMASI

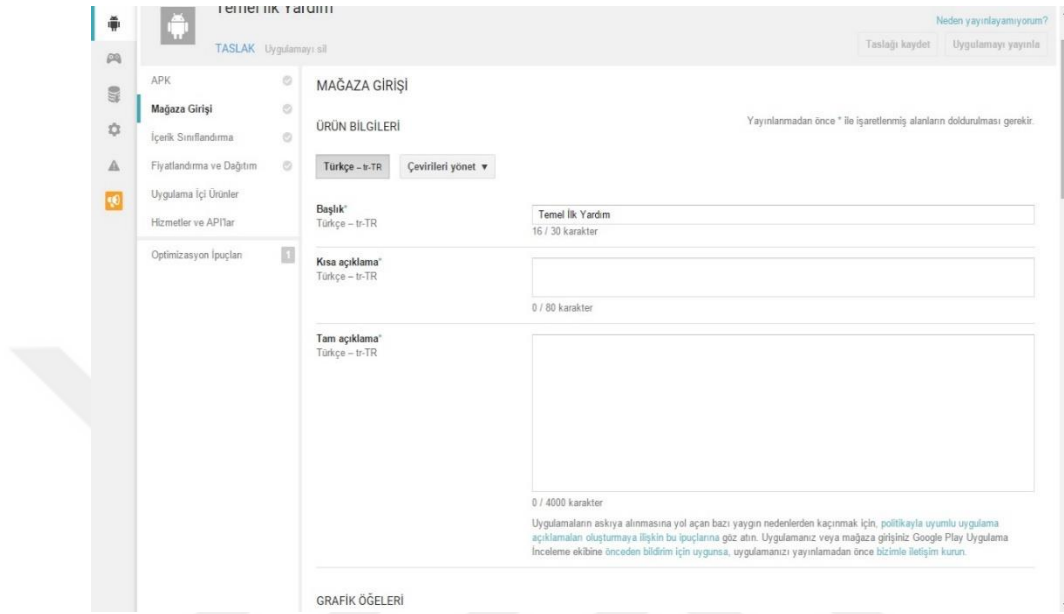
Temel ilk yardım mobil uygulamasının son aşaması android uygulamaların yer aldığı Google play (android market) sayfasında uygulamanın tanıtıldığı ve indirildiği platformda yayınlanmasıdır. Uygulamayı yayınlamadan önce uygulamanın son halinin farklı telefon modellerinde test edilmesi gerekmektedir. Çünkü android mobil işletim sistemine sahip cihazların gerek android sürümleri farklı olması gerek cihaz ekran boyutlarının farklı olması sebebiyle farklı cihazlarda uygulamalar sorun çıkartabilmektedir. Bu sorunları ortadan kaldırmak için android sürümleri ve farklı ekran boyutlarına sahip cihazlar belirlenerek (Motorola MB860, Asus zenfone5, Huawei G610 ve Samsung note 3) Temel ilk yardım uygulaması test edilip herhangi bir sorun ile karşılaşılmadığı görülmüştür. Uygulamada yer alan Müdahale adımları, Videolar, Hızlı arama tuşu ve Test uygulamaları tüm Andorid sürümlerinde çalışmaktadır. Ancak uygulamada yer alan 4 Tab menü yapısı ilk yardım, ek bilgi, eğitim süresi ve test başlıklarının daha düzgün gözükmesi ve kullanıcının yazabileceği “ek bilgi” – “test-sorularım“ uygulamasının çalıştırılabilmesi için Android 4 (API 14) ve üzeri sürümlerin kullanılması gerekmektedir.

Uygulamanın yayınlanması için öncelik olarak uygulamanın çalıştırılabilir paketinin oluşturulması ve apk dosyasının imzalanması gerekmektedir. Eclipse ortamında projeye sağ tıklanıp Android tools-Export Signed Application Package seçilir. Gelen ekran adımları izlenerek projenin adı seçilip, create new keystore seçeneği ile şifreyi, location kısmında oluşacak imza dosyasının nereye kaydedileceği, daha sonraki ekran adımlarında sırasıyla alias: dosyanın adı, password: keystore seçeneği ile belirlenen

şifreyi, confirm: yazılmış olan şifre tekrar girilir, validity(years): android sertifika süresi genel olarak google tarafından 25 yıl yazılması önerilir, first and last name: ad soyad, organization unit:şirketin departman adı, organization:şirket adı, city or locality:şehir, state or province:ilçe, country code:ülke kodu yazılarak en son adımda destination apk file kısmında imzalanacak apk dosyası seçilip finish denir. Apk dosyası imzalanmıştır. Uygulama Google play sayfasına yüklenmeye hazırdır.

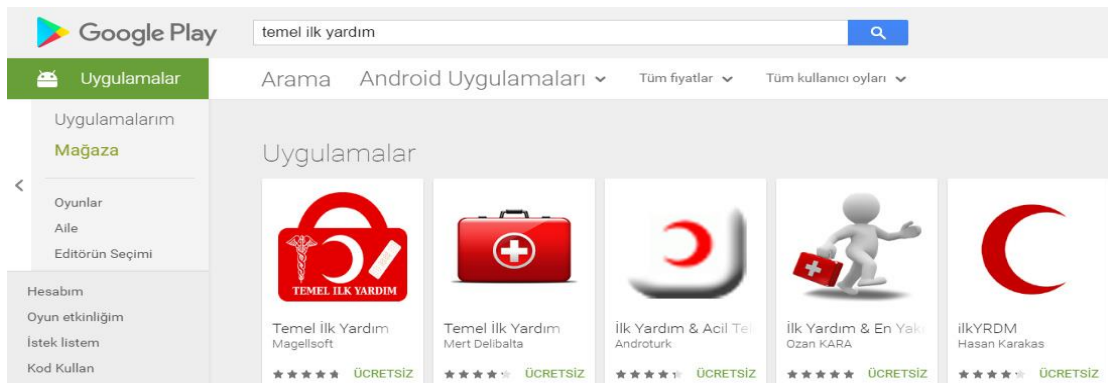
Uygulamanın yayınlanması için, Google Play Geliştirici Konsolu olarak adlandırılan ve 25 amerikan doları ücreti karşılığında üye olunarak sisteme giriş yapılmasına izin verilmektedir. Google Play Geliştirici Konsolu açıldığında gelen ilk ekranda daha önceden yayınlanan uygulama var ise ekranda yer alacaktır yok ise yeni bir uygulama eklemek için “Yeni uygulama ekle” düğmesi kullanılarak apk dosyasının yükleneceği sayfa gelmektedir. APK yüklemek için bilgisayarda bulunan ve önceden oluşturulan imzalı ve paketlenmiş APK dosyası seçilerek yüklenir. Yükleme aşamasında üç seçenek sunulmaktadır. Üretim, Beta Testi ve Alfa Testi. Uygulamanın son hali hazır olduğundan dolayı üretim kısmı seçilerek APK dosyası yüklenir ve sırasıyla diğer adımlar mağaza girişi, içerik sınıflandırma, fiyatlandırma ve dağıtım bilgileri doldurularak uygulamanın google play sayfasında yayınlanması sağlanır. Temel ilk yardım mobil uygulaması 20 Aralık 2015 günü Android Google Play Uygulama Marketi'nde yayınlanmıştır. Temel ilk yardım uygulaması "sağlık ve fitness" kategorisi altında ve herkes tarafından indirilebilen ücretsiz bir uygulamadır. Uygulama geliştirici kısmı belirleyici imza taşıması için uygulamayı Sunan: “Magellsoft” olarak yazılmıştır.

Uygulamanın toplam boyutu: 21M'dır. Şekil 7.1'de Google Play geliştirici konsolu mağaza girişi ekranı gösterilmiştir.



Şekil 7.1. Google Play Geliştirici Konsolu - Mağaza girişi

Google Play sayfası arama motoruna temel ilk yardım yazıp uygulama aratıldığında birinci sırada yer alan Şekil 7.2'deki Temel İlk Yardım geliştirici "Magellsoft" yazan uygulama yer almaktadır.



Şekil 7.2. Google Play Temel İlk Yardım uygulaması

7.1. Google Play Sayfasında Yer Alan Temel İlk Yardım Uygulaması Açıklama Kısmı

Google play sayfasında yer alan açıklama kısmı uygulamayı indirecek kişiler için uygulamanın tanıtıldığı ve uygulamanın özelliklerinden bahsedildiği açıklama alanıdır. Temel ilk yardım uygulaması 4 ana başlıktan oluşmaktadır. İlk yardım, ek bilgi, eğitim süreci, test

7.1.1. İlk yardım bölümü

- Hasta ve yaralıya temel ilk yardım ve müdahale adımları yer almaktadır.
- Temel ilk yardım başlıkları alfabetik sıraya göre dizilmiştir.
- Okumak istediğiniz ilk yardım başlığına tıklayarak müdahalenin içeriği hakkında bilgi alabilir ve müdahale işlemine dair video var ise ilk yardım Video kısmına tıklayarak ilgili video'ları izleyebilirsiniz. Video dosyaları uygulamanın içinde yüklü gelmektedir, çalışma sırasında internet'i kullanmaz. İlk Yardım müdahalelerinde yer alan '112 Ambulans Hızlı Arama Tuşu' ve '110 İtfaiye Hızlı Arama Tuşu' na basarak arama yapabilirsiniz.
- Video'lar Türk Kızılayı Youtube kanalından alınmıştır.
<https://www.youtube.com/user/TurkKizilayiOrg/videos>

7.1.2. Ek bilgi bölümü

- İlk yardım uygulaması Ek bilgi kısmında sizin tarafınızdan oluşturulacak yeni müdahale adımlarını yazabilir ve ilk yardım uygulamanızı kendinize göre güncelleyebilirsiniz.

7.1.3. Eğitim süreci bölümü

- Temel İlk Yardım eğitimi alacak kişiler için Eğitim sürecini, Sınav adımlarını ve Sertifika sürecini öğrenebilirsiniz.

7.1.4. Test bölümü

- 8 adet Test uygulaması bulunmaktadır. Her test uygulamasında 40 soru bulunmakta ve ilk yardım uygulamasında toplam 320 soru bulunmaktadır. Test süresi 60 dk.'dır. Test süreniz dolduğunda ekrana “60 dk süreniz dolmuştur. Test süreniz bitmiştir.” mesajı çıkacaktır. Test uygulamasında Sonuç yazısına tıklayarak cevaplarınız aşağıda gözükecek ve test süreniz kapatılmış olacaktır. Test bölümünde ayrıca Sorularım yazısına tıklayarak sizin tarafınızdan oluşturulacak yeni sorular yazabilir ve ilk yardım uygulamanızı kendinize göre güncelleyebilirsiniz.

Uygulamada yer alan 4 Tab başlığının daha düzgün gözükmesi ve kendi yazacağınız Ek Bilgi - Sorularım uygulamasının çalışabilmesi için Android 4 ve üzeri sürümleri kullanmanız tavsiye edilir. Google Play sayfasında temel ilk yardım uygulamasının tam açıklama kısmında bilgiler yer almaktadır.

Temel ilk yardım uygulaması 17.04.2016 tarihinde 1.1 sürüm olarak güncellenerek yayınlanmıştır. Güncelleme işleminde eski 1.0 sürümünde uygulamada 4.sekmede yer alan NOT sekmesinin ismi EK BİLGİ olarak güncellenmiştir ve 2.sekme alınmıştır.

VIII. SONUÇ

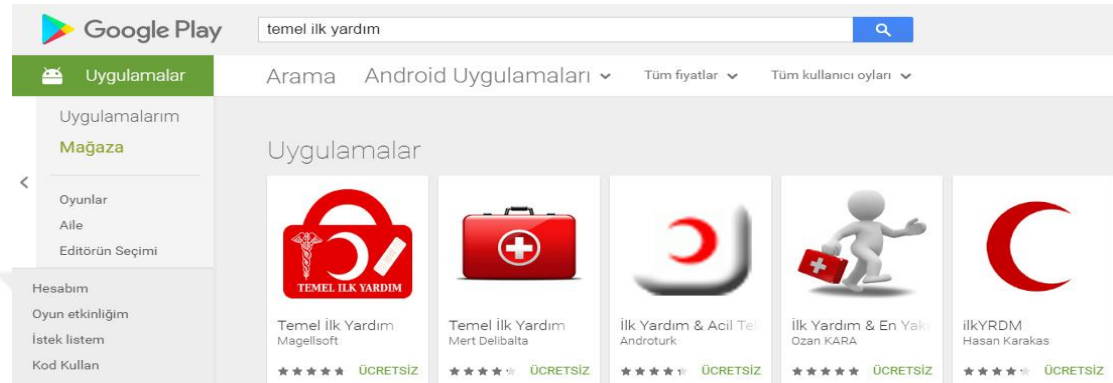
Bu tez'deki amaç; İnsanlar için herhangi bir kaza ya da yaşamı tehlikeye düşüren bir durumda sağlık ekiplerinin olay yerine ulaşınca dek insan hayatının kurtarılması veya kötüye gitmesini engellemek için tez kapsamında türkçe kaynaklı bilgilerin yer aldığı ve android işletim sisteminde çalışan bir uygulama geliştirilmesidir. Uygulama 4 ana başlık ilk yardım, ek bilgi, eğitim süreci ve test başlıkları altında toplanmıştır. Bu çalışmada, dört temel yapı oluşturulmuştur; temel ilk yardım bilgileri, videolar, test.

Bu uygulamayı diğer uygulamalardan ayıran özellikler:

- Acil aramaların doğrudan yapılması,
- Yeni ilk yardım bilgilerinin girilmesi,
- İlk yardım sertifikası alma adımlarının ayrıntılı anlatılması,
- İlk yardım sınavında çıkabilecek test sorularının bulunması,
- Kullanıcının ilave test sorusu ekleme imkanının olması.

Temel ilk yardım mobil uygulaması 20 Aralık 2015 günü Android Google Play Uygulama Marketi'nde 1.0 sürümü yayınlanmıştır. Temel ilk yardım uygulaması 17.04.2016 tarihinde 1.1 sürüm olarak güncellenerek yayınlanmıştır. Güncelleme işleminde eski 1.0 sürümünde uygulamada 4. sekmede yer alan NOT sekmesinin ismi EK BİLGİ olarak güncellenmiştir ve 2. sekmeye alınmıştır. Temel ilk yardım uygulaması "sağlık ve fitness" kategorisi altında ve herkes tarafından indirilebilen ücretsiz bir uygulamadır. Kullanıcıların uygulamayı indirip oy vermesi uygulamanın popüler olmasını ve diğer uygulamaların önünde üst sıralarda yer almasını

sağlamaktadır. Google play sayfası arama motoruna temel ilk yardım yazıp uygulama aratıldığında birinci sırada yer alan Şekil 8.1'deki Temel İlk Yardım geliştirici "Magellsoft" yazan uygulama yer almaktadır.



Şekil 8.1. Google Play Temel İlk Yardım uygulaması

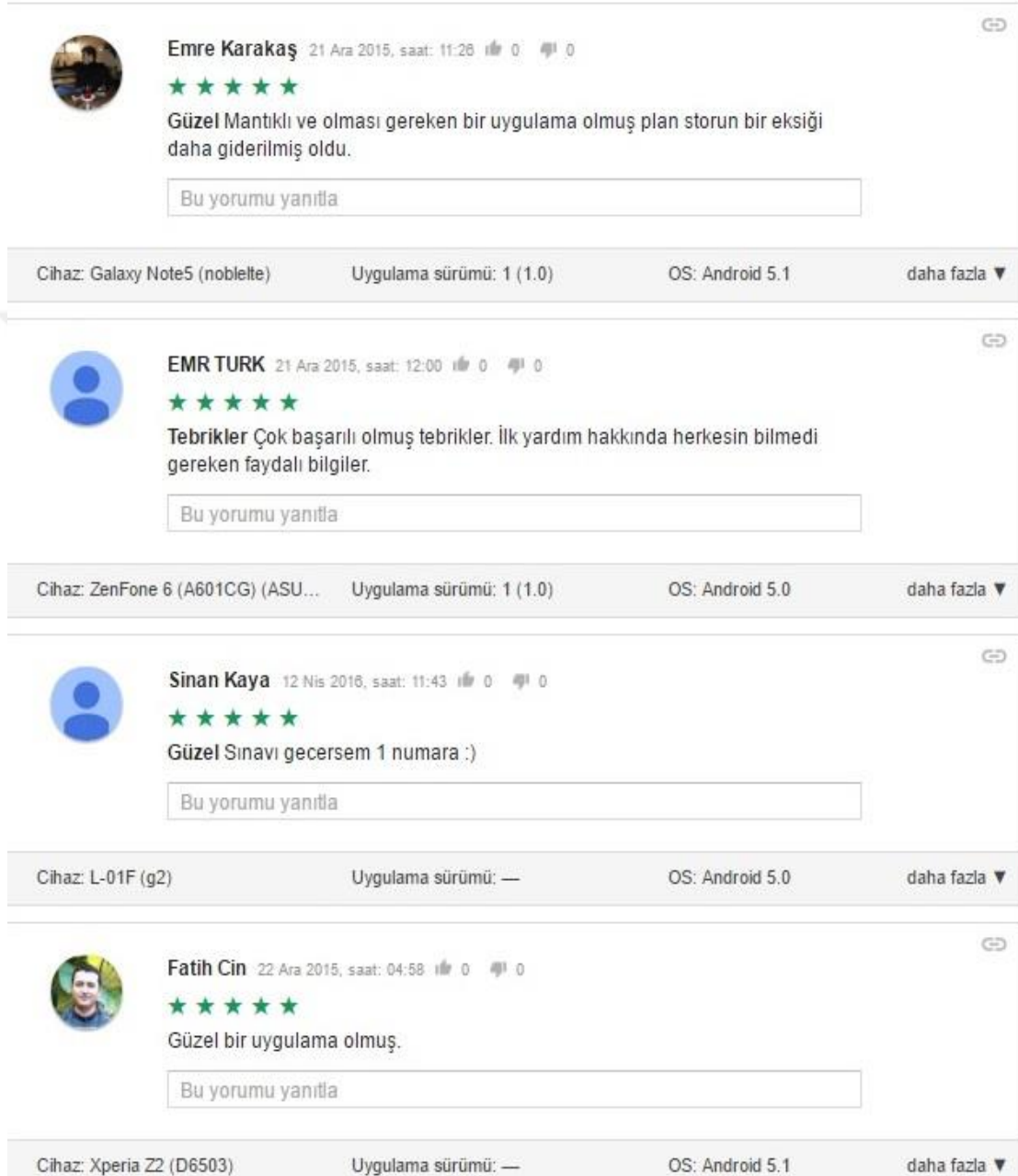
Uygulama 20 Aralık 2015 - 04 Haziran 2016 tarihleri arasında mobil telefon cihazlarında geçerli sayı: 453 ve toplam yükleme sayısı: 827 yükleme yapıldığı görülmektedir. 26 kişi tarafından uygulama ile ilgili oy kullandığı gözükmemektedir. Kullanıcıların kullandıkları oyların tamamı 5 üzerinden 4,81 değeri almıştır. Şekil 8.2'de uygulamanın indirilen sayısı ve oy sayısı gösterilmektedir.



UYGULAMA ADI	FİYAT	GEÇERLİ / TOPLAM YÜKLEMELER	ORT. OY / TOPLAM #	KİLİTLENMELER VE ANR'LER	SON GÜNCELLEME TARİHİ	DURUM
 Temel İlk Yardım 1.1	Ücretsiz	453 / 827	★ 4,81 / 26	1	17 Nis 2016	Yayımlandı

Şekil 8.2. Temel ilk yardım uygulaması kullanıcı oyları

Google Play uygulama marketinde kullanıcılar tarafından oy vermenin yanı sıra uygulama hakkında yorumlar yazılabilmekte ve herkes tarafından yorumlar okunabilmektedir. Uygulamada yazılan yorumların bazıları Şekil 8.3'de gösterilmiştir. Yorumlar sayesinde uygulamayı indirecek diğer kullanıcılar uygulama hakkında fikir

elde etmiş olmaktadır. Temel ilk yardım uygulamasına 12 kullanıcı tarafından yorum yazılmıştır.





Emre Karakaş 21 Ara 2015, saat: 11:28  0  0

★★★★★
Güzel Mantıklı ve olması gereken bir uygulama olmuş plan storun bir eksiği daha giderilmiş oldu.

Bu yorumu yanıtla



Cihaz: Galaxy Note5 (noblette) Uygulama sürümü: 1 (1.0) OS: Android 5.1 daha fazla ▼

EMR TURK 21 Ara 2015, saat: 12:00  0  0

★★★★★
Tebrikler Çok başarılı olmuş tebrikler. İlk yardım hakkında herkesin bilmediği gereken faydalı bilgiler.

Bu yorumu yanıtla



Cihaz: ZenFone 6 (A601CG) (ASU... Uygulama sürümü: 1 (1.0) OS: Android 5.0 daha fazla ▼

Sinan Kaya 12 Nis 2016, saat: 11:43  0  0

★★★★★
Güzel Sınavı geçersen 1 numara :)

Bu yorumu yanıtla

Cihaz: L-01F (g2) Uygulama sürümü: — OS: Android 5.0 daha fazla ▼

Fatih Cin 22 Ara 2015, saat: 04:58  0  0

★★★★★
Güzel bir uygulama olmuş.

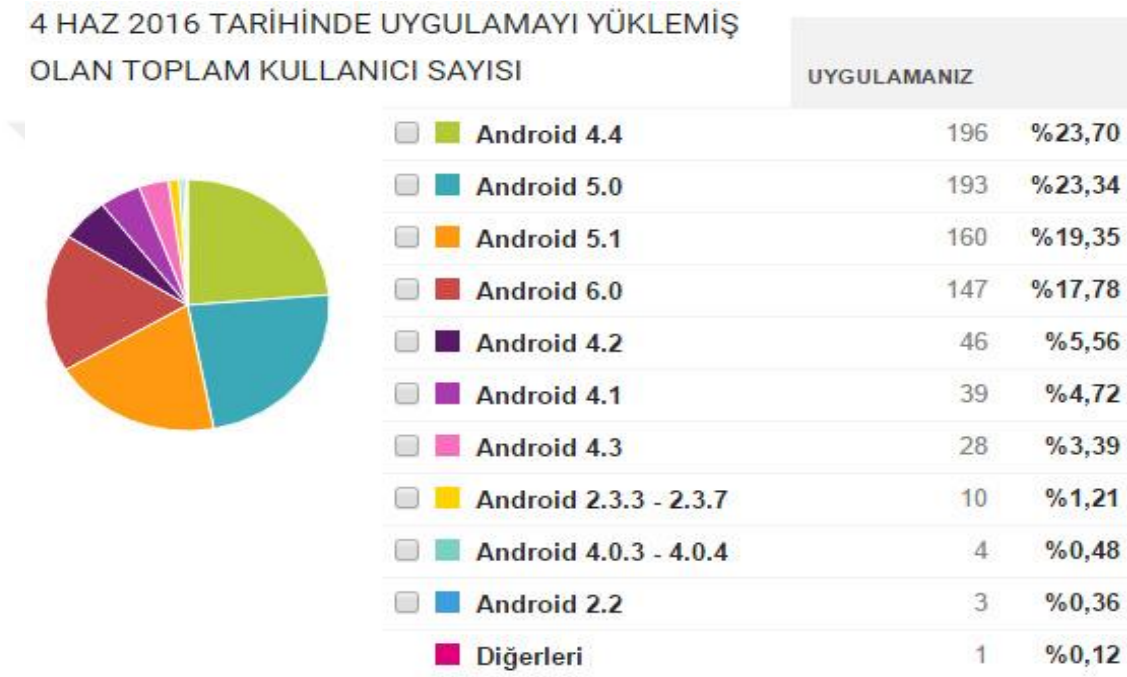
Bu yorumu yanıtla

Cihaz: Xperia Z2 (D6503) Uygulama sürümü: — OS: Android 5.1 daha fazla ▼

Şekil 8.3. Temel ilk yardım uygulaması kullanıcı yorumları

Uygulamayı yükleyen kişilerin olumlu geri dönüşleri ve uygulamanın almış aldığı puan sıralamasıyla uygulamanın faydalı olduğu görünmektedir. Android geliştiricisi

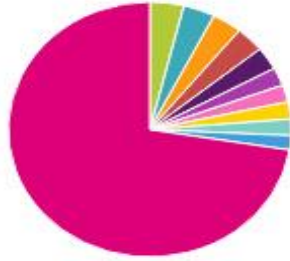
tarafından istatistikler bölümünde uygulamanın yüklü olduğu cihazların cep telefonu marka modeli ve android sürümleri konusunda bilgi veren Şekil 8.4'de gösterilen bir bölüm bulunmaktadır. Uygulamanın indiren toplam cep telefonu android sürümleri kontrol edildiğinde en çok Android 4.4 sürümüne uygulamanın yüklü olduğu görülmektedir.



Şekil 8.4. Temel ilk yardım uygulaması android sürümü yüklenme oranları

Yine aynı sorgu ekranından uygulamayı indiren toplam cep telefonu marka modeli kontrol edildiğinde en çok LGE LG G3 (g3) modeline uygulamanın yüklü olduğu görülmektedir. Şekil 8.5'de cep telefonu marka ve modellerine göre yüklenme oranları gösterilmektedir.

4 HAZ 2016 TARİHİNDE UYGULAMAYI YÜKLEMİŞ
OLAN TOPLAM KULLANICI SAYISI



	UYGULAMANIZ	
<input type="checkbox"/> LGE LG G3 (g3)	32	%3,87
<input type="checkbox"/> Samsung Galaxy Note...	30	%3,63
<input type="checkbox"/> Samsung Galaxy Note...	28	%3,39
<input type="checkbox"/> GENERAL_MOBILE	27	%3,26
<input type="checkbox"/> Samsung Galaxy J7 (j...	24	%2,90
<input type="checkbox"/> Samsung Galaxy S5 (...)	19	%2,30
<input type="checkbox"/> Samsung Galaxy S4 (...)	18	%2,18
<input type="checkbox"/> LGE LG G4 (p1)	18	%2,18
<input type="checkbox"/> Samsung Galaxy S3 (...)	17	%2,06
<input type="checkbox"/> Samsung Galaxy S3 M..	15	%1,81
<input type="checkbox"/> Diğerleri	599	%72,43

Şekil 8.5. Temel ilk yardım uygulaması cep telefonu marka model yüklenme oranları

KAYNAKLAR

About SQLite. (tarih yok). Kasım 15, 2015 tarihinde sqlite: <https://www.sqlite.org/about.html> adresinden alındı

Hayat Kurtarma Zinciri. (2015). Ekim 09, 2015 tarihinde ilkyardim: <http://www.ilkyardim.org.tr/hayat-kurtarma-zinciri.html> adresinden alındı

İÇCİ, E. (2014, Aralık 11). Android işletim sistemi ile çalışan mobil cihazlarda uygulama geliştirilmesi. s. 1-112. <https://tez.yok.gov.tr/UlusalTezMerkezi/> adresinden alındı

(2015). *İlkyardım Yönetmeliği*. Ankara: Sağlık Bakanlığı. Ekim 31, 2015 tarihinde <http://www.resmigazete.gov.tr/eskiler/2015/07/20150729-2.htm> adresinden alındı

Kızılayı, T. (2015, Ocak 15). *Türk Kızılayı*. youtube: <https://www.youtube.com/user/TurkKizilayiOrg/videos> adresinden alındı

MUŞTU, Y. (2012, Ocak 18). Sms ve android tabanlı uygulamalarla sınıf içi etkileşimin artırılması. s. 1-87. <https://tez.yok.gov.tr/UlusalTezMerkezi/> adresinden alındı

Number of apps available in leading app stores as of July 2015. (2015, Temmuz). Şubat 07, 2016 tarihinde statista: <http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/> adresinden alındı

ÖZCAN, M. (2013, Mart 29). Bir android uygulama modeli: İstanbul toplu taşıma bilgi sistemi. s. 1-85. <https://tez.yok.gov.tr/UlusalTezMerkezi/> adresinden alındı

Reddy, S. S. (2011, 10 05). Trip tracker application on android. s. 1-30. <http://sdsu-dspace.calstate.edu/bitstream/handle/10211.10/1303/Sreenivasa.pdf?sequence=1> adresinden alındı

SARIKAYA, B. (2013, Aralık). Mobil ilaç prospektüs uygulaması. s. 1-67. <https://tez.yok.gov.tr/UlusalTezMerkezi/> adresinden alındı

SARIKAYA, B. (2015, Eylül 08). *Cep ilaç*. Ağustos 01, 2015 tarihinde Google Play: <https://play.google.com/store/apps/details?id=biz.sarikaya.hazir&hl=tr> adresinden alındı

Scrum Nedir? (tarih yok). Ocak 09, 2016 tarihinde emsteknoloji: http://emsteknoloji.com/Scrum_Nedir? adresinden alındı

Smartphone OS Market Share, 2015 Q2. (2015, Ağustos). Nisan 17, 2016 tarihinde idc: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp> adresinden alındı

SQLiteDatabase. (tarih yok). Kasım 15, 2015 tarihinde developer.android:

[http://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html#setVersion\(int\)](http://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html#setVersion(int)) adresinden alındı

SQLiteOpenHelper. (tarih yok). Kasım 15, 2015 tarihinde developer.android:

<http://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html> adresinden alındı

Sutherland, K. S. (2013, Temmuz). *The Scrum Guide*. scrumguides:

<http://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-us.pdf> adresinden alındı

Temel İlk Yardım Bilgileri. (2015). Ekim 2015, 2015 tarihinde ilkyardim:

<http://www.ilkyardim.org.tr/temel-ilk-yardim-bilgileri.html> adresinden alındı

ÜNİVERSİTESİ, E. (2015, ocak 12). *Ege Mezun*.

<https://play.google.com/store/apps/details?id=com.egemezun&hl=tr> adresinden alındı

ÜSTÜNŞOY, E. (2015, Eylül 30). *Android 1.4 Milyar Kullanıcıya Ulaştı*. Aralık 09, 2015 tarihinde

shiftdelete: <http://shiftdelete.net/android-14-milyar-kullaniciya-ulasti-64374> adresinden alındı

EKLER

EK A

TÜRK KIZILAYI YOUTUBE KANALINDA YER ALAN İLK YARDIM UYGULAMALARINI İÇEREN TÜM VİDEOLARIN KULLANIM İZİNİ



yıldırım özyakışır <yildirimozyakisir@gmail.com>

İlet:: FW: Youtube 'da bulunan video'larınızın kullanımı hakkında.

4 ileti

Elif Taşkın <elif@kizilay.org.tr>

2 Aralık 2015 12:27

Alıcı: "yildirimozyakisir@gmail.com" <yildirimozyakisir@gmail.com>

Cc: Nagihan Toklu <nagihan.toklu@kizilay.org.tr>, Tayfun Kalsın <tayfun.kalsin@kizilay.org.tr>, Serkan Özkaptan <serkan.ozkaptan@kizilay.org.tr>, Mehtap Baykal <mehtapb@kizilay.org.tr>, Murat Sancar <murat.sancar@kizilay.org.tr>

Sayın Yıldırım Özyakışır,

Aşağıda belirtilen talebiniz değerlendirilmiş olup Türk Kızılayı YouTube kanalında yer alan ilk yardım uygulamalarını içeren tüm videolar ve www.ilkyardim.org.tr web sitesindeki bilgiler kaynakça gösterilerek kullanılabilir. Yapacağınız uygulamanın ücretli olup olmadığı, uygulamanın kimler için kullanılacağı ve uygulamanın nihai son halinin paylaşılmasına ilişkin bilgi verirsiniz seviniriz. İyi çalışmalar dileriz.

Elif Taşkın
İlk Yardım Eğitim Birimi
Türk Kızılayı
Ankara

----- Orijinal İleti -----

Konu: FW: Youtube 'da bulunan video'larınızın kullanımı hakkında.

Gönderen: Info <info@kizilay.org.tr>

Alıcı: Ulusalİlkyardim <_Ulusalİlkyardim@kizilay.org.tr>

CC:

From: yıldırım özyakışır [mailto:yildirimozyakisir@gmail.com]

Sent: Tuesday, December 01, 2015 10:14 AM

To: Info

Subject: Youtube 'da bulunan video'larınızın kullanımı hakkında.

Merhaba,

İsmim Yıldırım Özyakışır, Okan Üniversitesi - Bilişim Sistemleri Yüksek Lisans öğrencisiyim. Tez çalışmamı Android telefonlarında Temel ilk yardım uygulaması yapmaktayım. Türk Kızılayın youtube sayfasında yer alan (<https://www.youtube.com/user/TurkKizilayiOrg/videos>) Abc değerlendirilmesi - Tam tıkanma solunum vb. ilkyardim videolarını Android uygulamamda kullanmak istiyorum. Videoların kullanımı için sizden izin istiyorum. Yardımlarınız için çok teşekkür ederim.

Gereğinin yapılmasını arz ederim.

Yıldırım ÖZYAKIŞIR

EK B**İLK YARDIMCI BELGESİ**

	T.C. İSTANBUL VALİLİĞİ İL SAĞLIK MÜDÜRLÜĞÜ	
Belge No : Belge Geçerlilik Tarihi :	TEMEL İLK YARDIM/2004/1200/3164 26.07.2018	
İLK YARDIMCI BELGESİ		
Sayın YILDIRIM ÖZYAKIŞIR		
<i>İlkyardım Yönetmeliği kapsamında 25.07.2015 - 26.07.2015 tarihleri arasında Beyaz Gemi İlkyardım Eğitim Merkezi tarafından düzenlenen eğitim programını başarı ile bitirerek İLK YARDIMCI olmaya hak kazanmıştır.</i>		
YASEMİN EYÜPOĞLU Eğitimci	Dr. Özgür Yaşar Müdür a. İL Sağlık Müdür Yardımcısı	Yeşim Hekim Mesul Müdür

EK C

İLK YARDIMCI KİMLİĞİ

İLK YARDIMCI KİMLİK KARTI




TC KİMLİK NO
45835433746

ADI-SOYADI
YILDIRIM ÖZYAKIŞIR

BELGE NO
TEMEL İLK YARDIM/2004/1200/3164

BELGE GEÇERLİLİK TARİHİ
26.07.2018

BELGENİN VERİLDİĞİ MERKEZ
BEYAZ GEMİ İ.E.M.



İL SAĞLIK MÜDÜRLÜĞÜ

Bu kart sahibi;

Herhangi bir kaza ya da yaşamı tehlikeye sokan durumda hasta/yaralıya sağlık görevlerinin yardımı sağlanıncaya kadar hayatın kurtarılması ya da durumun daha kötüye gitmesini önlemek amacıyla olay yerinde, tıbbi araç ya da gereç aranmaksızın mevcut araç ve gereçlerle yapılan ilaçsız ilkyardımu uygulamalarını belge geçerlilik tarihleri arasında yapmaya yetkilidir.

Bu kart T.C. Sağlık Bakanlığı İkyardım Yönetmeliği uyarınca T.C. İstanbul Valiliği İl Sağlık Müdürlüğünce verilmiştir.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı: Yıldırım ÖZYAKIŞIR

Uyruğu: Türkiye

Doğum Tarihi ve Yeri: 8 Mart 1990, Kartal / İSTANBUL

Tel: +90 553 214 24 19

Email: yildirimozyakisir@gmail.com

EĞİTİM

Derece	Eğitim Birimi	Eğitim Tarihi
Yüksek Lisans	Okan Üniversitesi / Bilişim Sistemleri	2014-Devam ediyor
Lisans	Anadolu Üniversitesi / İktisat	2009-2013
Ön Lisans	Bilecik Üniversitesi / Bilgisayar Tek. ve Prog.	2007-2009
Lise	Kadıköy Tic. Mes. Lis. / Bilgisayar Prog.	2004-2007

İŞ DENEYİMLERİ

Kurum	Görev	Yıl
Martı Gemi Acentalığı	Bilgi İşlem	2014-Devam ediyor
Amerikan Hastanesi	Sistem Destek Uzman Yardımcısı	2013-2014
Unite BT	Turkcell Teknoloji / Sistem Destek Uzmanı	2010-2013
Probil	Sistem Destek Uzmanı	2009-2010

ANDROID İŞLETİM SİSTEMİNDE İLK YARDIM PAKETİ

YÜKSEK LİSANS TEZİ

YILDIRIM ÖZYAKIŞIR

tarafından

OKAN ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Bilişim Sistemleri Anabilim Dalına

YÜKSEK LİSANS

derecesi şartını sağlamak için sunulmuştur.

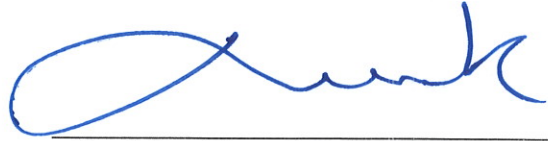
Onaylayan Jüri:



Yrd. Doç. Dr. Nurşen TOPÇUBAŞI
Danışman



Prof. Dr. Ahmet Faik KAŞLI
Üye



Prof. Dr. Ahmet Mesut RAZBONYALI
Üye

Haziran 2016

Program: Bilişim Sistemleri