

**T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

MOBİL ZEMİN BİLGİ SİSTEMİ GELİŞTİRİLMESİ

FURKAN ÖZOĞLU

**YÜKSEK LİSANS TEZİ
HARİTA MÜHENDİSLİĞİ ANABİLİM DALI
UZAKTAN ALGILAMA VE CBS PROGRAMI**

**DANIŞMAN
PROF. DR. TÜRKAY GÖKGÖZ**

İSTANBUL, 2017

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MOBİL ZEMİN BİLGİ SİSTEMİ GELİŞTİRİLMESİ

Furkan ÖZOĞLU tarafından hazırlanan tez çalışması 29.11.2017 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Harita Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Prof. Dr. Trkay GKGZ
Yıldız Teknik Üniversitesi

Jri yeleri

Prof. Dr. Trkay GKGZ
Yıldız Teknik Üniversitesi

Prof. Dr. Necla ULUĖTEKİN
İstanbul Teknik Üniversitesi

Yrd. DoĖ. Dr. Sinan ÇETİNKAYA
Yıldız Teknik Üniversitesi

ÖNSÖZ

Bu tezin konusunun belirlenmesinde ve hazırlanmasında değerli fikir ve önerileri ile bana destek olan hocam Prof. Dr. Türkay GÖKGÖZ'e çok değerli yardım ve ilgilerinden dolayı teşekkürlerimi sunarım.

Uygulamamın geliştirilmesi sürecinde bana sonsuz katkıda bulunan İstanbul Büyükşehir Belediyesi Coğrafi Bilgi Sistemi Müdürlüğü Mobil Yazılım Geliştirme Şefliği'ne ve Müdürlerime teşekkür ederim.

Uygulamamda kullanmak üzere veri teminini sağlayan İstanbul Büyükşehir Belediyesi Deprem ve Zemin İnceleme Müdürlüğü'ne ve verilerin hazırlanmasında yardımcı olan Emin Yahya MENTEŞE'ye teşekkürü bir borç bilirim.

Fikirleriyle, çalışmalarıyla hem üniversite hayatıma hem de iş hayatıma sayısız katkıda bulunan en yakın arkadaşım Salih YALÇIN'a ve beni bugüne getiren, karşılıksız desteklerini esirgemeyen aileme en içten teşekkürlerimi sunarım.

Aralık, 2017

Furkan ÖZOĞLU

İÇİNDEKİLER

	Sayfa
KISALTMA LİSTESİ.....	vi
ŞEKİL LİSTESİ.....	vii
ÇİZELGE LİSTESİ	ix
ÖZET	x
ABSTRACT.....	xii
BÖLÜM 1	
GİRİŞ.....	1
1.1 Literatür Özeti	1
1.2 Tezin Amacı	4
1.3 Hipotez	5
BÖLÜM 2	
TEMEL KAVRAMLAR VE STANDARTLAR	6
2.1 Jeolojik Temel Kavramlar ve Tanımlar	6
2.2 Sondaj Veri Standartları	8
2.2.1 Dünyadaki Sondaj Veri Standartları	9
2.2.2 Ülkemizdeki Sondaj Veri Standartları	11
2.2.3 Sondaj Veri Standartlarının Değerlendirilmesi	13
BÖLÜM 3	
AKILLI MOBİL CİHAZLAR VE PLATFORMLARI	14
3.1 Akıllı Mobil Cihazlar	14
3.2 Mobil İşletim Sistemleri	16
3.2.1 Android	17
3.2.2 iOS	17

BÖLÜM 4

iOS	19
4.1 iOS'un Tarihçesi	19
4.2 iOS Platformunda Uygulama Geliştirme	20
4.3 Swift Programlama Dili	21
4.3.1 Xcode	23
4.3.1.1 Storyboard.....	24
4.4 iOS Platformunda Uygulama Geliştirme Kütüphaneleri	25
4.4.1 Yerel (Lokal) Kütüphaneler	26
4.4.2 Harici (3. Parti) Kütüphaneler	27
4.4.2.1 CocoaPods.....	28
4.4.2.2 Harita Tabanlı Kütüphaneler	29

BÖLÜM 5

UYGULAMA	33
5.1 Temel Özelliklerin Belirlenmesi	34
5.1.1 Projenin Oluşturulması	34
5.1.2 Kütüphanelerin Eklenmesi	34
5.2 Tasarımın Hazırlanması	38
5.2.1 Mobil Cihazlarda Uygulama Tasarımı	38
5.2.2 İZemin Mobil Uygulama Tasarımı	39
5.3 Uygulamanın Geliştirilmesi	39
5.3.1 Veri Tabanı Modülü	40
5.3.1.1 Uygulama Alanı	40
5.3.1.2 Sondaj Verilerin Düzenlenmesi	40
5.3.1.3 Verilerin Uygulama ile Entegrasyonu.....	41
5.3.2 Harita Modülü	44

BÖLÜM 6

SONUÇ VE ÖNERİLER	54
-------------------------	----

KAYNAKLAR.....	56
----------------	----

EK-A

SONDAJ SINIFI SWIFT KODLARI.....	59
----------------------------------	----

EK-B

BULUT VERİ TABANINDAN UYGULAMAYA SONDAJ VERİLERİNİ YÜKLEYEN SINGLETON METOTU	60
---	----

ÖZGEÇMİŞ	62
----------------	----

KISALTMA LİSTESİ

ASCII	American Standard Code for Information Interchange
CBS	Coğrafi Bilgi Sistemleri
CEO	Chief Executive Officer
CSV	Comma Separated Values
ÇŞB	Çevre ve Şehircilik Bakanlığı
DOM	Dilim Orta Meridyeni
ESRI	Enviromental Systems Research Institute
GIS	Geographic Information System
GNSS	Global Navigation Satellite System
IBAction	Interface Builder Action
IBOutlet	Interface Builder Outlet
IDC	International Data Corporation
IDE	Integrated development environment
iOS	iPhone Operating System
IUGS	International Union of Geological Sciences
İBB	İstanbul Büyükşehir Belediyesi
JPEG	Joint Photographic Experts Group
JSON	JavaScript Object Notation
MacOS	Macintosh Operating System
MVC	Model View Controller
NoSQL	Not Only SQL database
OGC	Open Geospatial Consortium
OHA	Open Handset Alliance
OSM	Open Street Map
SDK	Service Development Kit
SVG	Scalable Vector Graphics
VTYS	Veri Tabanı Yönetim Sistemi
UML	Unified Modeling Language
UI	User Interface
USGIN	US Geoscience Information Network
UX	User eXperience
ZEBIS	Zemin Bilgi Sistemi
WGS84	The World Geodetic System 84

ŞEKİL LİSTESİ

	Sayfa
Şekil 2. 1	Stratigrafinin diğer bilimlerle ilişkisi..... 8
Şekil 2. 2	INSPIRE sondaj sınıfının UML diyagramı 10
Şekil 2. 3	GeoSciML sondaj sınıfının UML diyagramı..... 11
Şekil 2. 4	Ulusal veri standardına göre sondaj kuyusu sınıfının UML diyagramı 12
Şekil 3. 1	iPhone 1. Jenerasyon 14
Şekil 3. 2	HTC TC Dream 15
Şekil 3. 3	Akıllı telefon şirketlerinin pazar payları 15
Şekil 3. 4	Akıllı telefonlarda kullanılan işletim sisteminin pazar payları 16
Şekil 3. 5	OHA konsorsiyumu içerisinde yer alan firmalardan bazıları..... 17
Şekil 3. 6	2008 – 2017 yılları arasındaki App store’da yayınlanan uygulama sayısı 18
Şekil 4. 1	iOS işletim sistemi ile çalışan Apple cihazları..... 19
Şekil 4. 2	iOS versiyonlarının dağılım grafiği 20
Şekil 4. 3	iOS platformunda uygulama geliştirmek ve yayınlamak için gerekliler..... 21
Şekil 4. 4	Swift programlama dilinin açık kaynak kodları 22
Şekil 4. 5	Swift programlama dilinde örnek sınıf yapısı..... 23
Şekil 4. 6	xCode yazılımının kullanıcı arayüzü 24
Şekil 4. 7	Storyboard’daki ekranlardan birine eklenen bileşenler 25
Şekil 4. 8	Storyboard’daki ekranlar ve ekranlar arasındaki bağlantılar..... 26
Şekil 4. 9	CocoaPods’un bilgisayara kurulumu..... 29
Şekil 4. 10	CocoaPods pod dosyasının düzenlenmesi 29
Şekil 4. 11	Pod dosyasındaki kütüphanelerin projeye yüklenmesi 29
Şekil 4. 12	Google Maps iOS SDK’dan örnek ekran görüntüsü 30
Şekil 4. 13	Mapbox iOS SDK’dan örnek ekran görüntüsü 31
Şekil 4. 14	ArcGIS Runtime iOS SDK’dan örnek ekran görüntüsü 32
Şekil 5. 1	Çalışmanın çerçevesi 33
Şekil 5. 2	Veri-Arayüz-Kontrolcü Mimarisi (MVC) 36
Şekil 5. 3	iZemin’de kullanılan kütüphanelerin arayüz sınıfları ile ilişkileri..... 36
Şekil 5. 4	OneSignal bildirim takip ekranı 37
Şekil 5. 5	Firestore analitik takip ekranı 38
Şekil 5. 6	Uygulama geliştirme süreci..... 40
Şekil 5. 7	Uygulama alanı..... 40
Şekil 5. 8	Firestore veri tabanını görüntüleme ve düzenleme ekranı 42
Şekil 5. 9	Firestore veri tabanında oluşturulan veri şeması 43
Şekil 5. 10	Apple MapKit Kütüphanesi 2B harita, uydu görüntüsü 44

Şekil 5. 11	Konuma git butonu ve cihaz konumunu	46
Şekil 5. 12	Konum izni akış diagramı	46
Şekil 5. 13	Sondaj noktalarının liste ve harita üzerindeki gösterimi	47
Şekil 5. 14	Formasyon ve litoloji bilgilerinin gösterilmesi	48
Şekil 5. 15	Haritada herhangi bir noktadan veya koordinat bilgilerinden arama	49
Şekil 5. 16	ArcGIS buffer aracı ile tampon bölgeler (mavi: düzlem, kırmızı: elipsoidal) 52	
Şekil 5. 17	iZemin uygulamasından 5 km’lik örnek tampon bölge.....	52
Şekil 5. 18	ArcGIS Buffer aracının düzlem mesafe seçeneği ile (pembe), ArcGIS Buff..	53
Şekil 5. 19	Kuzey, Doğu, Güney ve Batı sınırlarında tampon böl. arasındaki farklı.....	53



ÇİZELGE LİSTESİ

	Sayfa
Çizelge 2. 1	İstanbul Birliği'nin kaya-stratigrafi birimleri..... 9
Çizelge 4. 1	iOS Platformunda Sık Kullanılan Kütüphaneler 28
Çizelge 5. 1	iPhone 7 donanım özellikleri 34
Çizelge 5. 2	iZemin uygulamasının geliştirilmesinde kullanılan kütüphaneler..... 35
Çizelge 5. 3	Uygulamada kullanılan öznitelikler ve veri tipleri 41

MOBİL ZEMİN BİLGİ SİSTEMİ GELİŞTİRİLMESİ

Furkan ÖZOĞLU

Harita Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Prof. Dr. Türkay GÖKGÖZ

Yer altı ve yer üstü saha çalışmalarında yaygın olarak kullanılan sondaj verilerinin Coğrafi Bilgi Sistemi araçlarıyla yönetimi doğrultusunda bugüne kadar çeşitli masaüstü ve web tabanlı programlar geliştirilmiştir. Bu programlar yardımıyla; veri girişinden, çeşitli sorgulama ve analizlerle elde edilen sonuçların çeşitli şekillerde sunumuna kadar, bir dizi işlem gerçekleştirilebilmektedir. Son zamanlarda, mobil cihaz teknolojisinin gelişmesi ve kullanımının giderek yaygınlaşmasıyla birlikte, sondaj bilgi sistemi niteliğinde çeşitli mobil uygulamalar da geliştirilmiştir. Masaüstü, web ve mobil ortamda geliştirilen program ve uygulamalar farklı özelliklere sahiptir. Ayrıca, tüm dünyada veya ülkemizde geçerli ortak veri standardı olduğu da söylenemez.

Bu çalışmada, sondaj noktalarının ve bilgilerinin görülebileceği, mekânsal sorgulamaların yapılabileceği, iOS platformunda harita tabanlı bir mobil uygulama geliştirilmiştir. xCode geliştirme ortamının en güncel versiyonu kullanılarak, Swift programlama dili ile geliştirilmiş olan bu uygulama kullanıcı dostu bir arayüze sahiptir. Yerel (cihaz içinde) bir veri tabanı yerine bulut veri tabanı kullanılarak, verilerin mobil cihaz haricinde merkezi olarak tutulması, yönetilmesi ve güncelliğinin korunması sağlanmıştır. Kullanılan veri tabanı sistemi; ilişkisel değil, NoSQL (Not Only SQL) bir veri tabanı ile çalışmaktadır. Sonuç olarak, hem sözel hem de harita üzerinde işaretlenen ya da koordinatları girilen bir noktaya dayalı mekânsal aramalar yapılabilmiş olup, sistemde kayıtlı tüm sondaj bilgileri küçük ekranlı akıllı cihazlarda bile etkili bir şekilde gösterilebilmiştir. Uygulamaya eklenen analitik aracı ve bildirim servisi sayesinde, kullanıcı deneyimi de takip edilebilmiştir.

Anahtar Kelimeler: Mobil Uygulama, Swift Programlama Dili, Sondaj, iOS



DEVELOPMENT OF MOBILE BOREHOLE INFORMATION SYSTEM

Furkan ÖZOĞLU

Department of Geomatics Engineering

MSc. Thesis

Advisor: Prof. Dr. Türkay GÖKGÖZ

A variety of desktop and web-based programs have been developed up to date with the use of Geographic Information System tools for borehole data which are widely used in underground and aboveground field studies. With the help of these programs; from the input data to presentation of the results obtained by various queries and analysis, a series of operations can be carried out in various forms. In recent years, various mobile applications have been developed as well as borehole information system with the increasing use and development of mobile device technology. The programs and applications developed in desktop, web and mobile environment have different features. Also, it can not be said that there is a common data standard which is valid all over the world or our country.

In this study, a map-based mobile application on the iOS platform where spatial queries can be made, borehole points and informations could be seen was developed. This application that has a user-friendly interface developed with the Swift programming language, using the latest version of the xCode development environment. Using a cloud database instead of a local (in-device) database was provided in terms of centrally kept, managed and updated data outside the mobile device. NoSQL was used as a alternative of relational databases. As a result, the developed application provides the best usage performance. It is possible to make spatial searches, both semantic and based on a point marked on the map or entered with coordinates, and all the borehole information recorded in the system can be shown effectively even on small screen smart devices.

Keywords: Mobile Application, Swift programming language, Borehole, iOS



YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

1.1 Literatür Özeti

Ülkemizde ve dünyada birçok alanda jeolojik veriler kullanılmaktadır. Kamu kurumları ve özel şirketlerin yaptıkları çalışmaların yanında akademik çalışmalar da yapılmaktadır. Yapılan çalışmalar yüksek maliyetlidir ve elde edilen verilerin saklanması, çeşitli bilgi sistemlerine entegre edilmesi, üzerinde analizler yapılması gerekmektedir.

Sondaj çalışmaları, jeolojik bilgilerin edinilmesine yönelik önemli çalışmalardan biridir. Sondaj; yerkabuğunun tanınması ve araştırılması, doğal kaynakların (su, gaz, petrol, maden, jeotermal enerji kaynakları) aranması ve işletilmesi, büyük yapıların (baraj, gölet, fabrika, güç santralleri, tünel, köprü, yol yapımları, inşaat işlerinde vs.) temel yapılarının araştırılması ve bu gibi yapıların inşaat işlerinde, deprem gibi doğal afetlerin araştırılmasında, jeolojik vb amaçlı veri toplama ve değerlendirme amacıyla, değişik doğrultularda (düşey, yatay, eğik) açılan küçük çaplı (1 – 2 cm’den 100 – 300 cm’ye kadar) kuyulardır. Sondaj kuyuları açıldıktan sonra yapılan analiz çalışmaları sonrasında raporlar hazırlanmaktadır. Bu raporlar koordinatlarıyla birlikte bir veri tabanında tutulmaktadır. Literatüre bakıldığında hazırlanan verileri kullanarak, jeoloji ve zemin bilgilerinin yönetiminin yapıldığı çeşitli platformalarda gerçekleştirilmiş çeşitli çalışmalar ve geliştirilmiş uygulamalar bulunmaktadır.

SedLog; yer altını grafik olarak gösterebilen, Comma Separated Values (CSV) formatında veriler eklenebilen; PDF, Scalable Vector Graphics (SVG) ve Joint Photographic Experts Group (JPEG) formatlarında da çıktı alınabilen bir masaüstü yazılımıdır [1]. SedMob ise SedLog programının mobil uygulamasıdır. Bu uygulamada veriler sunucu yardımı ile CSV

formatında alınmaktadır. Hibrid olduğu için iOS ve Android gibi mobil platformlarda çalışabilmektedir [2].

Güney Kore, Seoul'da yerel on bin sondaj noktasını ve diğer jeolojik verileri göstermek için arama, istatistiksel analiz gibi özellikler barındıran bir web tabanlı CBS uygulaması geliştirilmiştir [3]. Ayrıca, artırılmış gerçeklik teknolojisi ve harita kütüphanesi kullanılarak, yerel kurumların sondaj veri yapısına uygun, iPad cihazlarla uyumlu bir mobil uygulama (BoreholeAR) da geliştirilmiştir [4].

İstanbul için Bir Çekirdek Zemin Bilgi Sistemi Oluşturulması Projesi kapsamında; mevkii ve cadde ismine göre sorgulamalarla harita üzerinde istenen bir yere yaklaşılabilmesi, seçilen bir yere en yakın ya da belli bir mesafedeki sondajların gösterilmesi, seçilen bir sondaja ilişkin bilgilerin sunulması ve seçilen sondajlara ilişkin bilgilerle RockWorks99 programının stratigrafi (stratigraphy) dosya formatında bir ASCII dosyanın meydana getirilmesi gibi işlevleri olan bir masaüstü programı (ZeBiS: **Zemin Bilgi Sistemi**) geliştirilmiştir. Sistemin altlık haritaları sayısal fotogrametrik yöntemle üretilmiş 1:5 000 ölçekli topografik haritalar ve yazılım bileşenleri AutoCAD Map, Access, RockWorks ve AutoCAD Map altında çalışan ve AutoLISP ile yazılmış bir arayüz programıdır [5].

İstanbul'daki Sismik Riskin Değerlendirilmesi Projesi kapsamında, jeolojik ve geoteknik bir dizi analiz çalışmasının yapılabilmesi için bir masaüstü programı (GeoData: **Geological Database**) geliştirilmiştir. GeoData yardımıyla; Access veri tabanında tutulan sondaj verileri, konum ve çeşitli sondaj özniteliklerine göre sorgulanabilmekte ve ayrıca, seçilen sondajların ArcView ortamında haritalarda gösterilebilmesi ve RockWorks ortamında sondajlar arası kesitler çıkarılabilmesi için gerekli dosyalar hazırlanabilmektedir [6].

Geological Survey of Denmark and Greenland (GEUS); Danimarka'nın iklim bakanlığı bünyesinde kurulmuş, uluslararası odaklı, bağımsız bir araştırma ve danışma kurumudur. Danimarka ve Grönland bölgelerinde jeolojik alanda bilimsel araştırma ve çalışmalar yapmaktadır. Bu çalışmalar içerisinde jeolojik haritalar üretmektedirler. Web tabanlı olarak hazırlanan bir platform üzerinden 280.000'den fazla sondaj verileri açık bir şekilde paylaşılmaktadır. Harita üzerinde tıklanan veya aratılan sondajın detayları tablo halinde gösterilmektedir. Ayrıca sondaj raporları ve formasyon bilgileri pdf formatında indirilebilmektedir [7].

German Geological Survey Organization'a (GSO) göre Almanya'da sondaj verileri mevzuta uygun bir şekilde kayıt altına alınmaktadır. GSO bu verilerin işlenmesi, depolanması ve iletilmesinden sorumludurlar. 2009 yılında Almanya'nın sondaj haritasının web ve harita tabanlı olarak oluşturulması için çalışma başlatılmıştır. Bu çalışma sonucunda ücretsiz olarak jeolojik profilleri (stratigrafi, petrografi), yeraltı suyu ölçümleri, sondaj formasyonları gibi jeolojik veriler paylaşılmıştır [8].

Mobil zemin uygulamalarının haricinde harita tabanlı mobil uygulama çalışmaları da yapılmaktadır. BeeSis mobil uygulaması arıların popülasyonunu takip etmek için iOS tabanlı olarak geliştirilmiş bir iPhone uygulamasıdır. Bu çalışma öncelikli olarak gönüllüğü esas almaktadır. Plot bölge olarak Amerikadaki gönüllüler, arıların aktivitelerini bu uygulamada toplayarak arı popülasyonundaki değişimleri ve arıların bal üretmek için ziyaret ettikleri bitkileri takip ederek bilimsel çalışmalara katkıda bulunmayı amaç edinmişlerdir [9].

Amerika Güney Kaliforniya'daki endemik bitkilere dair harita tabanlı bir mobil uygulama geliştirilmiştir. Bu geliştirilen uygulamanın temel amacı insanlara yerli bitkiler hakkında bilgi vermektir. iOS platformunda Objective C programlama dili kullanılmıştır. Ayrıca harita kütüphanesi olarak ESRI'nin ArcGIS Runtime iOS SDK'sı ve veriler üzerinde ekleme, silme, düzenleme işlemleri için ArcGIS Server 10.1 kullanılmıştır [10].

e-SoilMap, hibrit (hem iOS hem de Android platformuna uyumlu) harita tabanlı bir mobil uygulamadır. Altlık harita olarak Open Street Map (OSM) kütüphanesi kullanılmıştır. Bu kütüphanenin tercih edilmesindeki en önemli etken haritanın çevrimdışı olarak kullanılabilmesidir. OSM kütüphanesinde 1:15 000 ölçeğindeki altlık haritanın boyutu 200 mb'lık veri alanına ihtiyaç duymaktadır. Uygulama 3 temel bileşenden oluşmaktadır: (1) harita yönetimi, (2) toprak yönetimi, (3) kullanıcı yönetimi. Bu uygulamanın amacı, tarım ekosistemine yönelik saha araştırmalarındaki toprak bilgisinin etkinliğini artırmak, özellikle teknik konularda yeterli bilgi sahibi olmayan kullanıcıları desteklemek, mobil uygulamadaki verileri okuyabilmesini ve kullanabilmesini sağlamaktır [11].

Bu çalışma da yukarıda açıklanan konu ve kapsamda bir çalışmadır. Çalışmanın amacı ve hipotezi bu bölümün geri kalan kısmında açıklanmaktadır. Bölüm 2'de sondaj ve formasyon gibi bazı jeolojik temel terimler ile birlikte dünya ve ülkemizdeki sondaj veri

standartları açıklanmaktadır. Akıllı mobil cihazlar ve platformları Bölüm 3'te ele alınmaktadır. Bu çalışmanın da platformu olan iOS, Bölüm 4'te ayrıntılı olarak incelenmektedir. Bölüm 5'te ise bu çalışmada geliştirilen mobil uygulama tüm işlem adımlarıyla birlikte ayrıntılı açıklanmaktadır. Geliştirilen mobil uygulamanın özgün değerleri, mevcutlara göre avantaj ve dezavantajları Bölüm 6'da ortaya konulmaktadır.

1.2 Tezin Amacı

Bu çalışmada, saha çalışmalarında yaygın olarak kullanılan sondaj verilerinin, iOS tabanlı bir mobil uygulamada formasyon bilgilerinin yanı sıra konumsal bilgileriyle birlikte harita üzerinde gösterilmesi amaçlanmıştır.

Bu amaca ulaşmak için aşağıdaki hedefler belirlenmiştir.

- Arayüz tasarımı (UI-User Interface) ve uygulama prototipi (UX-User Experience) hazırlanacaktır.
- Kullanıcı dostu (User-Friendly) arayüze sahip olacaktır.
- Apple tasarım ilkeleri ile uyumlu ve tüm iPhone ekranlarına uyumlu olacaktır.
- iOS platformunda geriye dönük versiyonları destekleyecektir.
- Geliştirme ortamı (IDE) ve Swift programlama dilinin en güncel versiyonu kullanılacaktır.
- Apple'ın Corelocation ve MapKit kütüphaneleri kullanılacaktır.
- Veriler JSON nesneleri olarak, NoSQL bulut veri tabanında tutulacaktır.
- Hem semantik arama hem de geometrik arama yapılabilecektir.
- Tamamen veya kısmen ücretsiz servislerden yararlanılacaktır.
- Analitik (Firebase Analytics) ve bildirim (OneSignal) araçları kullanılarak, kullanıcı deneyimi takip edilecektir.

Geliştirilen uygulamanın genel özellikleri ise aşağıdaki gibi olacaktır;

- İşletim sistemi: iOS 9.3 – 11.0
- Uygulama tasarım aracı: Sketch

- Uygulama geliştirme platformu: xCode 9
- Uygulama geliştirme tasarım aracı: Storyboard
- Uygulama geliştirme dili: Swift 4
- Veri tabanı: Firebase
- Harita altlığı: MapKit
- Konum altlığı: Core Location

1.3 Hipotez

Sondaj bilgi sistemlerine birçok alanda gereksinim duyulmaktadır. Bugüne kadar çeşitli masaüstü, web tabanlı ve mobil sondaj bilgi sistemleri geliştirilmiştir. Mevcut mobil sondaj bilgi sistemlerinin hiçbirinde NoSQL bulut veri tabanı kullanılmamaktadır. Bu bağlamda; tüm kullanıcıların eş zamanlı olarak bağlanabildiği bir NoSQL bulut veri tabanı kullanan bir mobil sondaj bilgi sistemi native uygulama olarak geliştirilebilir. Böylece, lokal veri tabanlarının çeşitli dezavantajları (kullanıcının güncel veriye hemen ulaşamaması gibi) ortadan kaldırılabilir, işletim sisteminin geliştiricilere sağladığı tüm kaynaklarına (donanımsal ve yazılımsal) erişilebilir ve hibrit olarak geliştirilen uygulamalara göre performans açısından daha iyi sonuçlar elde edilebilir.

TEMEL KAVRAMLAR VE STANDARTLAR

2.1 Jeolojik Temel Kavramlar ve Tanımlar

Zemin Etütü ve Raporları: Zemin incelemesindeki başlıca amaç meydana gelebilecek hasarları önceden tahmin etmek ve mümkün olduğunca önüne geçmek için gerekli tedbirleri almaktır. Özellikle zemini iyileştirmenin gerekli olup olmadığını ve zeminin tipini belirlemektedir. Zeminin temel yapısal etkileşiminin irdelenmesinde kullanmak üzere zemin özelliklerinin ve zeminin fiziksel parametrelerinin belirlendiği rapora zemin etüt raporu denir. Ülkemiz dünyanın en karmaşık jeolojik yapısına sahip ülkelerden biridir. Bu nedenle basit jeoloji özelliğine sahip yapılara göre daha detaylı etüt çalışmaları yapılması gerekmektedir [12], [13]. Bu rapor özellikle ülkemizde 17 Ağustos 1999 Marmara depreminden sonra ciddi anlamda tartışılmaya başlanmıştır. Zemin etüt raporu hazırlanması kanuni zorunluluk haline getirilmiştir. Zemin ve temel etütlerinin, zemin ve yapı etkileşimi göz önüne alınarak, temel tasarımı için gerekli olabilecek her türlü bilgiyi kapsayacak şekilde yapılmış olması, temel ve kazı sistemlerinin doğru olarak projelendirilmesi ve yapım sırasında karşılaşılabilecek çeşitli zemin problemlerine önceden çözümler getirilmiş olması gerekmektedir [13], [14]. Zemin etüt çalışmalarının yapılmaması, eksik veya hatalı yapılması yapı güvenliği açısından sakıncalı sonuçlar doğurabilir. Bu sakıncaları ortadan kaldırmak için, zeminin mühendislik özelliklerinin ve temel ön tasarımının belirlenmesi kapsamında yazılan zemin etüt raporlarının doğru ve standartlara uygun bir şekilde yapılması gerekmektedir [13].

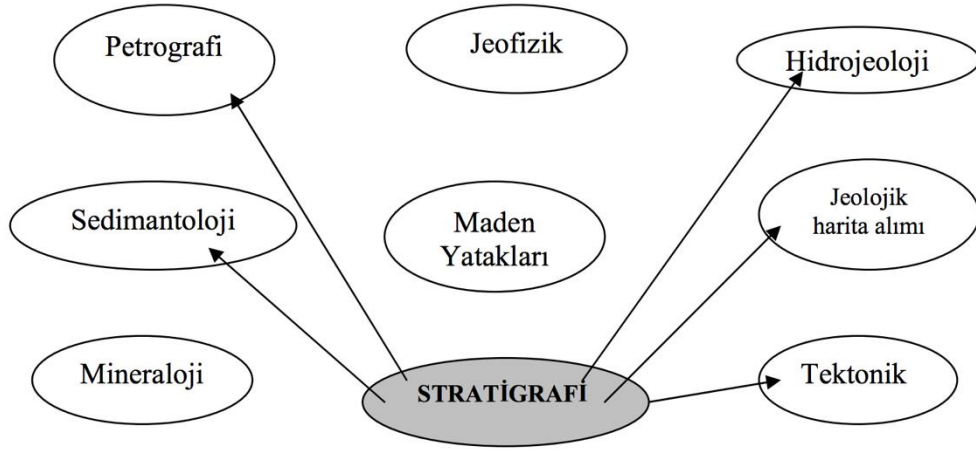
Coğrafi bilgi sistemlerini özelleştirerek zemin çalışmaları için hazırlanan zemin bilgi sistemleri çeşitli analiz işlemlerinde kullanılabilmektedir. Zemin bilgi sistemleri web ve mobil sistemlerle entegrasyonu sağlanarak internet ortamında bilgi paylaşımı sağlanmaktadır.

Sondaj: Eski çağlardan günümüze kadar, insanoğlunun yeraltı doğal zenginliklerin saptanması ve hammadde üretimine yönelik bir aracı olarak kullanılmıştır. Teknolojik gelişmeler ile sondaj işlemleri gelişmiştir. Sondaj, çeşitli amaçlarda, farklı çaplarda ve derinliklerde yer yüzeyinde delik açmaya yarayan yöneme denilmektedir.

Sondajlar aşağıdaki gibi sınıflandırılmaktadır [15] :

- Derinliklerine göre: Sığ, derin, çok derin
- Çaplarına göre: Dar, geniş, çok geniş
- Yerlerine göre: Yer üstünde, yer altında, su üstünde
- Amaçlarına göre
 - Yapılış amaçlarına göre: Arama, geliştirme, üretim
 - Aranılan madde türüne göre: Petrol, su, zemin, jeotermal, vb
- Yöntemlerine göre: Darbeli, döner, kombine

Stratigrafi: Tabakalı kayaların litoloji, fosil içeriği, yaş ve jeofizik temelinde gruplanma esaslarını, böylece gruplanmış paketlerin geometrisini, düşey ve yanal devamlılığını ele alan; bunların arkasındaki gerçekliği ortaya koymaya çalışan bilim dalıdır. Şekil 2.1’de stratigrafinin diğer bilimlerle ilişkisi gösterilmiştir.



Şekil 2. 1 Stratigrafinin diğer bilimlerle ilişkisi [16]

Sondaj Formasyonu: Jeolojik haritaya açık seçik geçirilebilecek, büyük ölçüde homojen litolojilerden (bir veya birkaç kaya türünün düşey/yanal yönde ardalanmasından) ibaret bir birimdir ve haritada gösterilebilmektedir. Bir formasyon adı ancak bir jeolojik makale olarak dergide basıldığında formal (kurallı) olarak hayat bulmaktadır. Bu noktadan sonra, karışıklık ve yanlış yorumlanmanın önüne geçmek için diğer jeologlar bu ismi kullanmalıdırlar [16]. Formasyonların alt bölümleri ise formasyon üyesi olarak adlandırılmaktadır. Çizelge 2.1’de İstanbul Birliği’nin formasyon ve üyelerini içeren kaya stratigrafi birimleri gösterilmektedir.

Litoloji: Taşların fiziksel, kimyasal ve dokusal özelliklerini inceleyen bilim dalıdır. Ayrıca bir bölgedeki hâkim kayaç ve tabaka yapısına göre isimlendirmede de kullanılmaktadır.

2.2 Sondaj Veri Standartları

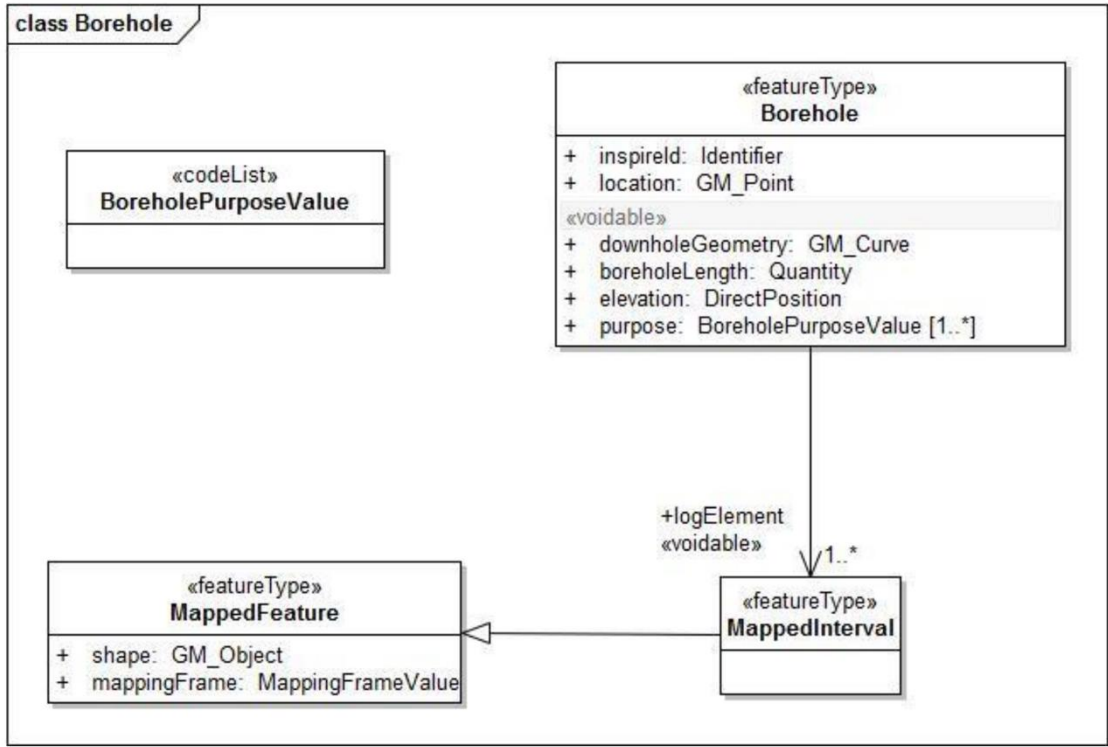
Sondaj verileri, zemin çalışmalarında daimî olarak kullanılmaktadır. Son yıllarda sondaj verileri sayısal ortamda yani veri tabanında tutulmaktadır. Bu veri tabanlarına bilgisayar ve akıllı mobil cihazlar üzerinden bu verilere erişilebilmesi için ihtiyaç duyulmaktadır. Zeminle ilişkili hem geometrik hem de semantik verilerin, belirli standartlarda veri tabanlarında saklanması için dünyada ve ülkemizde çeşitli çalışmalar yapılmaktadır. Bu standartlaştırma sürecinde ihtiyaçlar doğrultusunda sürekli olarak güncellemeler yapılmaktadır.

2.2.1 Dünyadaki Sondaj Veri Standartları

INSPIRE D2.8.II.4: Avrupa Birliği mekânsal altyapı standartlarından biri olan bu standart 2013 yılında hazırlanmıştır. Jeolojik verilerin özellikleri ve teknik kurallardan oluşmaktadır. Şekil 2.2’de sondaj sınıfının Unified Modeling Language (UML) diyagramı verilmiştir. Bu diyagrama göre sondaj sınıfındaki bir sondaj nesnesinin özellikleri şunlardır: id, konum, geometri, uzunluk, yükseklik ve amaç.

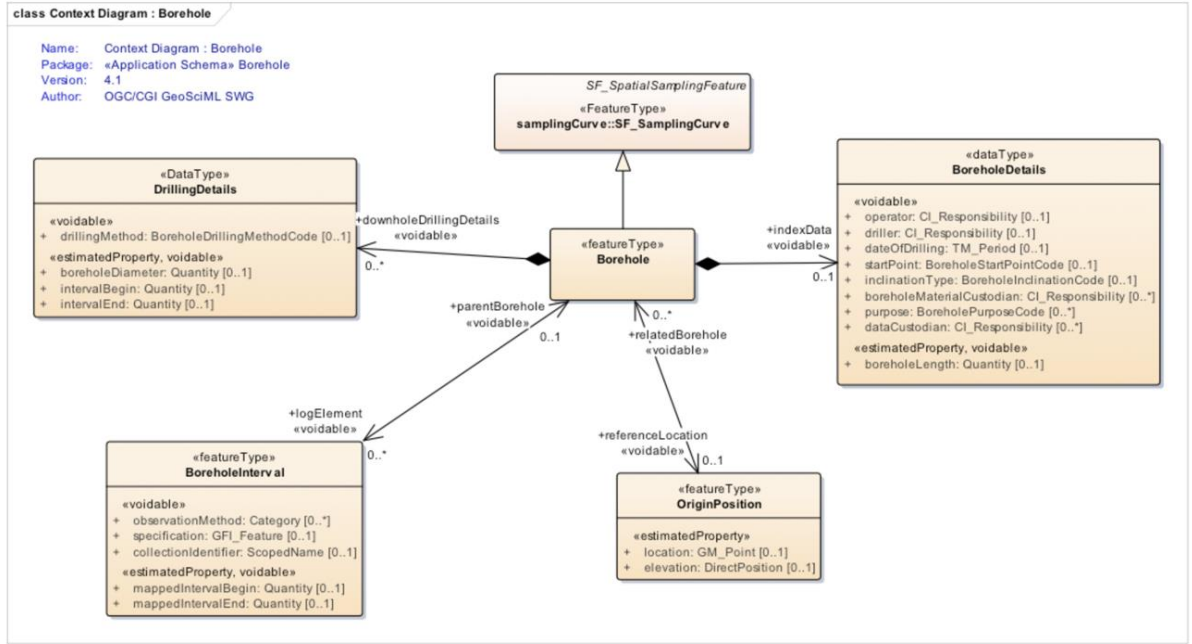
Çizelge 2. 1 İstanbul Birliği'nin kaya-stratigrafi birimleri [17]

Formasyon		Simge	Üye	Yaş
GEBZE GURUBU	Tepecik Formasyonu (TRgt)	TRgtk	Kazmalı Üyesi	Üst Triyas
		TRgtü	Übeyli Üyesi	
	Balıkkaya Formasyonu (TRgb)	TRgbs	Sortullu Üyesi	Orta Triyas
		TRgbk	Karabeyli üyesi	
	Demirciler Formasyonu (TRgd)	TRgdh	Hacılı Üyesi	
		TRgdg	Göksu Deresi Üyesi	
	Erikli Formasyonu (TRge)	TRged	Değirmen Üyesi	Alt Triyas
		TRgev	Yeniköy Kuntaşı Üyesi	
Kapaklı Formasyonu PTRk		PTRkkc	Kocadere Üyesi	Permien-Alt Triyas
		PTRkkky	Kayaran Tepe Üyesi	
		PTRkkkr	Karacatepe Volkanit üyesi	
		PTRkdd	Diyabaz Üyesi	
		PTRkkt	Tavşancıl Volkanit Üyesi	
Sancaktepe Graniti		Ps	Kovanlık Çakıtaşı Üyesi	Permien
Trakya Formasyonu (Ct)		Ctk	Küçükköy Üyesi	Alt Karbonifer
		Ctkk	Kartaltepe Üyesi	
		Ctc	Cebeciköy Üyesi	
		Cta	Acıbadem Üyesi	
Denizli Köyü Formasyonu		DCdb	Baltalimanı Üyesi	Alt Karbonifer
		DCda	Ayineburnu Üyesi	Alt Karbonifer-Üst Devoniyen
		DCdy	Yörtükali Üyesi	Orta-Üst Devoniyen
		DCdt	Tuzla Kireçtaşı Üyesi	Orta Devoniyen
Pendik Formasyonu		Dp	Ayrılmamış	Alt-Orta Devoniyen
		Dpk	Kozyatağı Üyesi	
		Dpkz	Kartal Üyesi	
Pelitli Formasyonu		SDp	Ayrılmamış	Üst Silüriyen-Alt Devoniyen
		SDpsğ	Soğanlık Üyesi	
		SDps	Sedefadaşı Üyesi	
		SDpd	Dolayoba Kireçtaşı Üyesi	
		SDpm	Mollafenari Üyesi	
Yayalar Formasyonu		OSy	Ayrılmamış	Üst Ordovisiyen (?) -Alt Silüriyen
		OSyş	Şeyhli Üyesi	
		OSyu	Umur Deresi Üyesi	
		OSyg	Gözdağ Üyesi	
Aydos Formasyonu		Oa	Ayrılmamış	Alt-Orta Ordovisiyen
		Oaa	Ayazma Kuvarsit Üyesi	
		Oab	Başıbüyük Üyesi	
		Oak	Kısıklı Üyesi	
Kınalıada Formasyonu		Okm	Manastır Tepe Üyesi	Alt-Orta Ordovisiyen
		Ogg	Gülsuyu Üyesi	
		Opks	Süreyyapaşa Üyesi	
POLONEZKÖY GURUBU	Kurtköy Formasyonu	Opkb	Bakacak Üyesi	Alt Ordovisiyen
		Opkc	Ayrılmamış	
	Kocatöngel Formasyonu			



Şekil 2. 2 INSPIRE sondaj sınıfının UML diyagramı [18]

GeoSciML: Jeolojik veriler için, temel harita verilerinden karmaşık ilişkisel jeolojik veri tabanlarına kadar kullanılan veri modeli ve veri aktarım standardıdır. GeoSciML önceki sürümlerinde, International Union of Geological Sciences (IUGS) standardı iken birkaç yıl içerisinde yapılan güncellemeler ile INSPIRE, the US Geoscience Information Network (USGIN), Australian AuScope and AusGIN Projects standartlarını da içermektedir. Yayınlanan 4. versiyonu ile birlikte 2017 yılında Open Geospatial Consortium (OGC) tarafından da standart olarak kabul edildi. OGC standartları, teknoloji geliştiricilerine coğrafi bilgi ve hizmetleri, herhangi bir uygulamayla erişilebilir kılarak, kullanışlı hale getirmeyi hedeflemektedir. Şekil 2.3'deki GeoSciML standartının sondaj sınıfı UML diyagramı verilmiştir. Temel sondaj sınıfı, sondaj detayları, sondaj konumu, sondaj aralığı ve sondaj delme detayları ile ilişkilendirilmiştir.



Şekil 2. 3 GeoSciML sondaj sınıfının UML diyagramı [19]

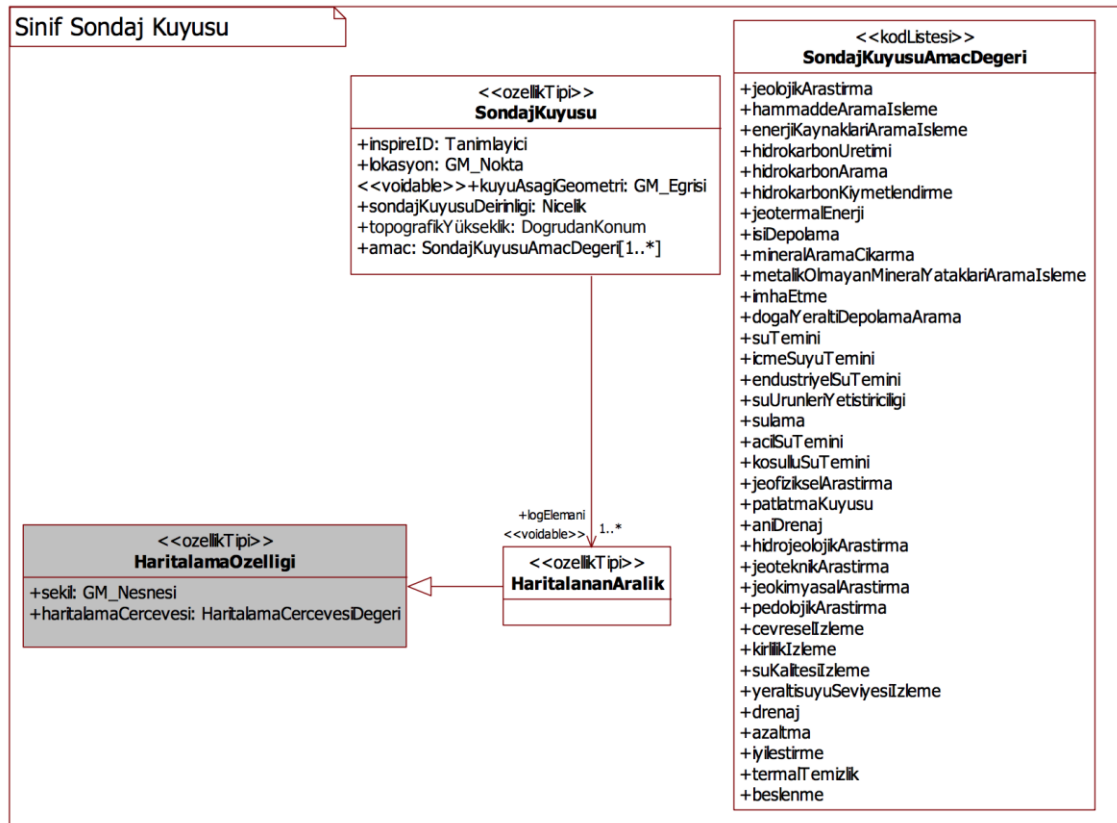
2.2.2 Ülkemizdeki Sondaj Veri Standartları

Ülkemizde Çevre ve Şehircilik Bakanlığı, Coğrafi Bilgi Sistemleri Genel Müdürlüğü'nün 644 sayılı Kanun Hükmünde Kararname ile tanımlanmış görevleri arasında “Coğrafi veri ve bilginin ulusal düzeyde üretimine, kalitesine ve paylaşımına yönelik standartlar ile bunlara ilişkin temel politika ve stratejilerin belirlenmesini sağlamak ve gerekli mevzuatı hazırlamak” ve “Bakanlık hizmetlerinin etkin bir şekilde yürütülebilmesi için Bakanlık mekânsal veri altyapısının oluşturulması ve geliştirilmesi ile Bakanlığın ihtiyaç duyacağı her türlü verinin iletilmesi ve temin edilmesi konularında çalışmalar yürütmek” bulunmaktadır. Tanımlanan bu görevden yola çıkılarak, Avrupa Birliği Mekânsal Bilgi Altyapısı (INSPIRE) Direktifine bağlı olarak yayınlanan uygulama kurallarından jeoloji veri tanımlama dokümanları temel alınarak, Türkiye’de jeoloji veri temasına ilişkin ulusal standartların oluşturulması ve ülkemizde yetkili kurumlar tarafından üretilen jeoloji veri temasına ait verilerin bu ulusal standartlara göre hazırlanması hedeflenmektedir. Bu nedenle Coğrafi Bilgi Sistemleri Genel Müdürlüğü tarafından bu çalışmanın sınırları tanımlanmış ve başlatılmıştır.

Proje kapsamında proje analizleri aşağıdaki kurumlar ile gerçekleştirilmiştir;

- Enerji ve Tabii Kaynaklar Bakanlığı Maden Tetkik ve Arama Genel Müdürlüğü
- Enerji ve Tabii Kaynaklar Bakanlığı Türkiye Petrolleri Anonim Ortaklığı
- Orman ve Su Bakanlığı Devlet Su İşleri
- Başbakanlık Afet ve Acil ve Durum Yönetimi Başkanlığı
- Ulaştırma Bakanlığı Karayolları Genel Müdürlüğü
- Ulaştırma Bakanlığı Devlet Demiryolları
- Çevre ve Şehircilik Bakanlığı İller Bankası Anonim Şirketi Genel Müdürlüğü

Jeoloji veri standartlarının belirlenmesinde sondaj kuyuları sınıfını içeren UML diyagramı Şekil 2.4’de verilmiştir. Bu diyagram INSPIRE D2.8.II.4 (2013)’den değiştirilerek oluşturulmuştur [20]. Sondaj sınıfı içerisinde lokasyonu, geometrisi, sondaj kuyusu derinliği, topoğrafik yüksekliği ve amacı yer almaktadır. Sondaj kuyusu amaç değerleri liste halinde verilmiştir ve bire çok ilişkilendirilerek içerisinden seçilmektedir.



Şekil 2. 4 Ulusal veri standardına göre sondaj kuyusu sınıfının UML diyagramı [20]

2.2.3 Sondaj Veri Standartlarının Değerlendirilmesi

Çevre ve Şehircilik Bakanlığı tarafından hazırlanan ulusal standart, büyük ölçüde Inspire D2.8.II.4 standardına benzemektedir. Örneğin; sondaj lokasyonu, sondaj kuyu derinliği, sondaj kuyusu açma amaç değeri gibi öznitelikler ve birbirleriyle ilişkilendirilen veri tabloları (haritalanan aralık ve haritalama özelliği) her iki standartta da mevcuttur. Bununla birlikte, örneğin sondaj amaç değerleri ulusal standartta açıkça belirtilmişken Inspire D2.8.II.4 standardında belirtilmemiştir. GeoSciML standardı, yukarıdaki iki standarda öznitelik olarak benzemekle birlikte, veri tablosu ve tabloların ilişkisel yapısı bakımından farklılıklar göstermektedir. Örneğin GeoSciML standardında, yukarıdaki iki standarttan farklı olarak, sondaj detayları, konumu ve aralığı ayrı tablolarda yer almaktadır. Bununla birlikte kazı detaylarını içeren bir tablo da yer almaktadır. Çalışmada kullanılması planlanan özniteliklerden sondaj formasyonları ve litoloji bilgileri yukarıda adı geçen üç standartta da yer almamaktadır. Mevcut masaüstü, web ve mobil sondaj uygulamalarında yerel veya uluslararası sondaj veri standartları aynen kullanılmamaktadır. Bunun yerine, kullanıcı beklentilerini karşılayacak şekilde uygulamaya özel yeni veri yapılarının hazırlandığı görülmüştür [1], [2], [3], [4], [5], [6], [7], [8]. Bu çalışmadaki veri yapısı ise bölüm 5.3.1’de detaylı olarak açıklanmaktadır.

AKILLI MOBİL CİHAZLAR VE PLATFORMLARI

3.1 Akıllı Mobil Cihazlar

1993 yılında piyasaya sürülen ilk dokunmatik ekranlı akıllı telefon IBM firmasının üretmiş olduğu Simon'dur. Akıllı mobil cihazların gelişimi o tarihlerden günümüze kadar artarak devam etmiştir [21]. 2007 yılında Apple firmasının iPhone cihazı tanıtımı yapıldı. Bu cihazın en önemli özelliği Apple'ın kendi işletim sistemi olan ve ilk defa tanıtılan iOS işletim sistemini kullanması ve geniş dokunmatik bir ekran sunmasıdır [21] (Şekil 3.1).



Şekil 3.1 iPhone 1. Jenerasyon

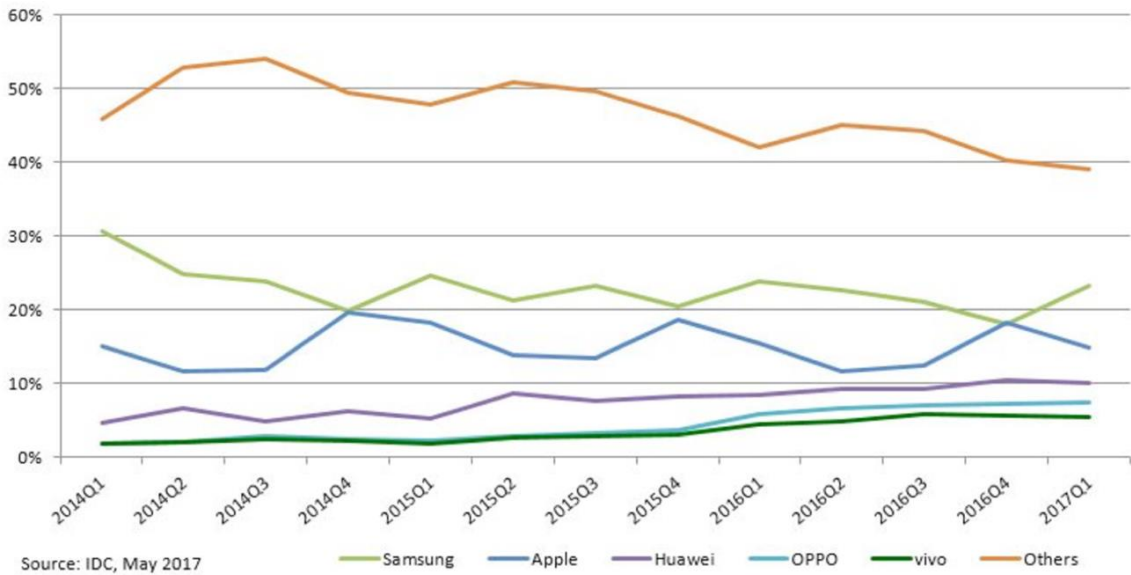
iPhone cihazının ardından diğer firmalar da yeni akıllı telefonlarını ve bu cihazlarda çalışacak işletim sistemlerini tanıttılar. iPhone cihazlarının çıkışından yaklaşık bir yıl sonra 2008 yılında Tayvan firması olan HTC, Google tarafından desteklenen ilk android işletim sistemli akıllı telefonunu piyasaya sürdü. "TC Dream" isimindeki bu cihazda Google'ın

Gmail, Google Takvim, Google Haritaları gibi bazı uygulamalarıyla birlikte sunuldu (Şekil 3.2). Samsung, Sony gibi bazı öncü telefon firmaları da Android işletim sistemini kullanan cihazlar ürettiler [21].



Şekil 3. 2 HTC TC Dream

International Data Corporation (IDC) tarafından hazırlanan akıllı telefon şirketlerinin pazar paylarını gösteren grafikte görüldüğü gibi büyük ve öncü şirketlerin yanında diğer farklı şirketler tarafından da akıllı cihazlar üretilmektedir (Şekil 3.3).



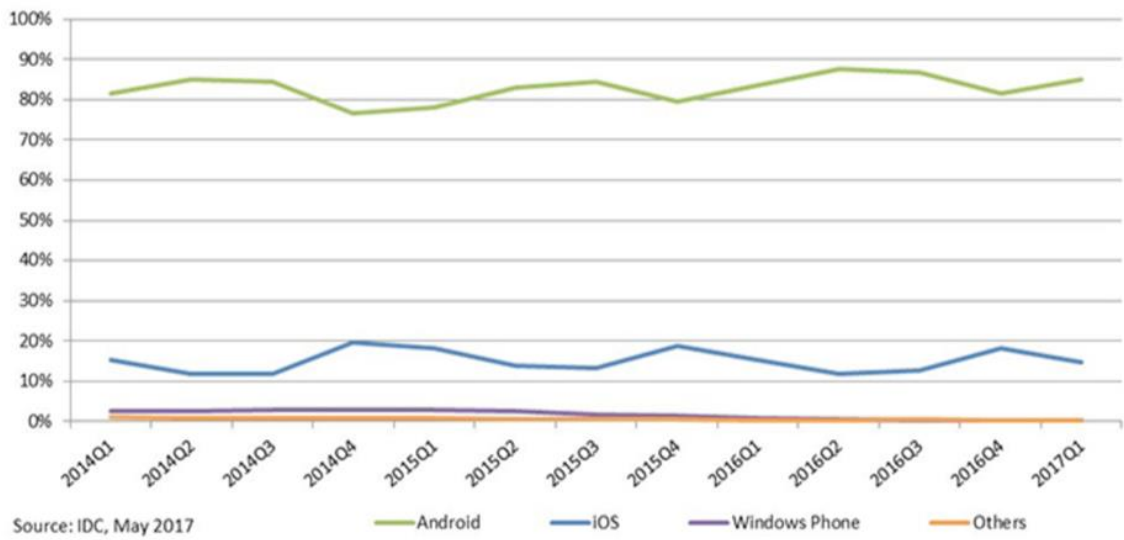
Şekil 3. 3 Akıllı telefon şirketlerinin pazar payları [22]

3.2 Mobil İşletim Sistemleri

Günümüzde yaygın olarak akıllı telefonlar ve tablet cihazları kullanılmaktadır. Bu cihazlar içerisinde çalışan işletim sistemleri mobil işletim sistemleri olarak adlandırılmaktadır. İşletim sistemi, akıllı olmayan telefonların aksine, bu cihazlar içerisinde yer alan uygulamaların haricinde kullanım amaçlarına uygun uygulamaların yüklenebilmesini sağlamaktadır. Bu sayede bu akıllı cihazlar bir bilgisayar gibi kullanılabilir. Bu sayede bu akıllı cihazlar bir bilgisayar gibi kullanılabilir.

Mobil cihazların ilk çıktığı zamanlardan günümüze kadar birçok işletim sistemi geliştirildi. Bunlardan Symbian, Web OS, MeeGo, Bada, Blackberry OS artık kullanılmayan ya da çok az kullanılan işletim sistemleridir. Günümüzde en yaygın olanları ise iOS, Android, Windows Phone'dur.

IDC tarafından hazırlanan grafikte işletim sistemlerinin pazar payları gösterilmiştir (Şekil 3.4). Grafikten açıkça görüleceği üzere Android işletim sistemi 2017 ilk çeyreğindeki değerine göre %85 ile en büyük payı elinde tutmayı başarmıştır. Android işletim sisteminin bu denli yaygın olmasının birkaç önemli sebebi vardır. Bunlar, iOS işletim sisteminin sadece kendi ekosisteminde çalışıyor olması ve Android işletim sisteminin açık kaynak olarak tüm geliştirici ve firmalara ücretsiz olarak veriliyor olmasıdır.



Şekil 3. 4 Akıllı telefonlarda kullanılan işletim sisteminin pazar payları [23]

3.2.1 Android

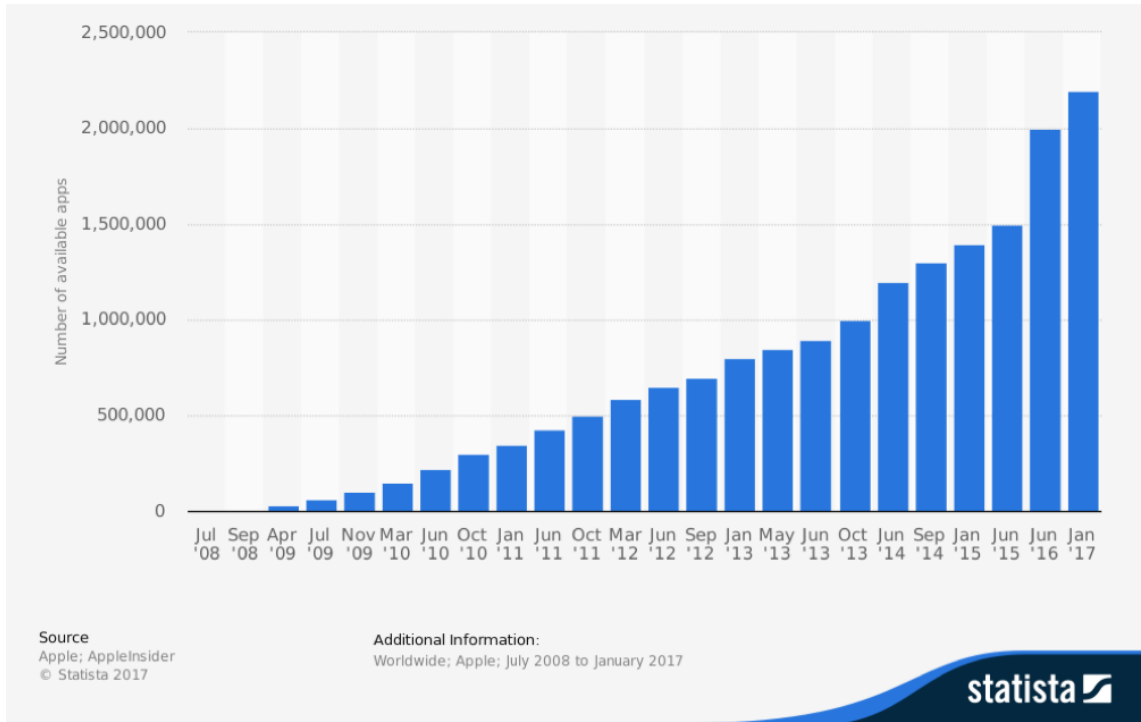
Google ve Open Handset Alliance (OHA) tarafından açık kaynak ve Linux tabanlı olarak geliştirilmiş ve Şubat 2008 yılında ilk Android versiyonu olan Astro yayınlanmıştır. Open Handset Alliance konsorsiyumu, Google öncülüğünde ve 84 firmanın yer aldığı, android işletim sisteminin gelişimi için kurulmuştur (Şekil 3.5). Android işletim sisteminin son sürümü Nougat 7.1.2 dir ve Mayıs 2017’de yayınlanmıştır.



Şekil 3. 5 OHA konsorsiyumu içerisinde yer alan firmalardan bazıları

3.2.2 iOS

Apple firmasının iPod, iPhone ve iPad cihazlarında kullanmak için geliştirdiği bir işletim sistemidir. iPhone cihazı ilk kez 2007 yılında tanıtıldıktan sonra Nisan 2010 yılında duyurulan iPad cihazlarında da iOS işletim sistemi kullanılmıştır. Apple 2008 yılında 3. Parti uygulamaların cihazlara indirilmesini sağlayan bir uygulama mağazası olan App Store uygulamasını yayınlamıştır. Bu uygulama ile mobil uygulama geliştirmesi yapan firma ve yazılımcılara yeni kapı açılmıştır. Statista’nın raporuna göre App Store’da Temmuz 2008’de yaklaşık 800 uygulama yer alırken 1 yıl sonra yani Temmuz 2009’da 65.000 uygulama ulaşmıştır (Şekil 3.6). Ocak 2017 tarihinde ise 2.2 milyon uygulama Apple’ın uygulama mağazasında mevcuttur. Şu an en güncel versiyonu iOS 11 dir.

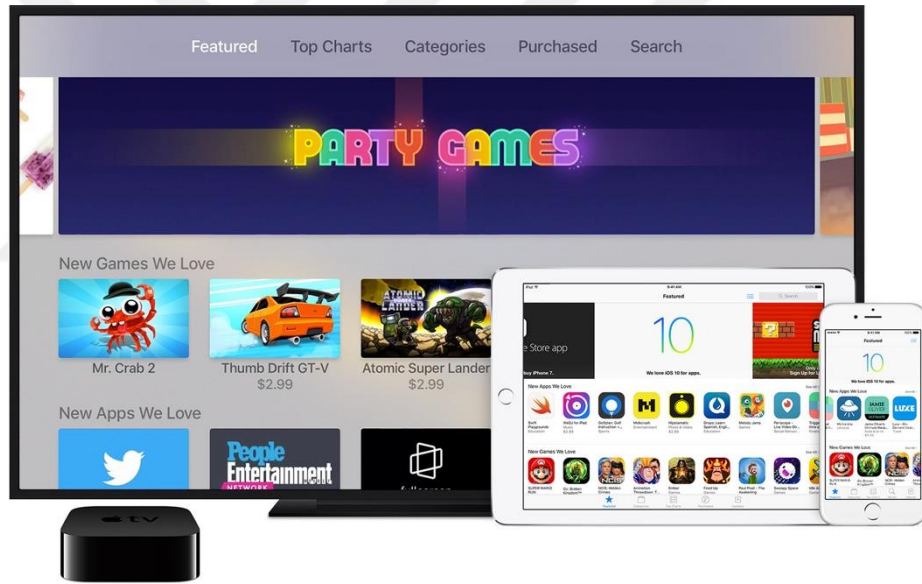


Şekil 3.6 2008 – 2017 yılları arasındaki App store’da yayınlanan uygulama sayısı [24]

BÖLÜM 4

iOS

Apple tarafından üretilen iPhone, iPad, iPod Touch ve Apple TV cihazlarında çalışan işletim sistemidir (Şekil 4.1). Apple'ın iOS işletim sisteminin sadece kendi cihazlarında kullanılmasını sağlamak için Android işletim sisteminin aksine kapalı kaynaklıdır.

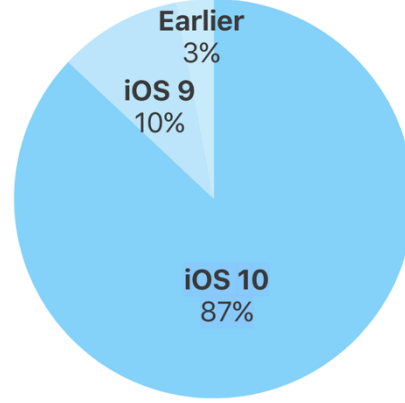


Şekil 4. 1 iOS işletim sistemi ile çalışan Apple cihazları

4.1 iOS'un Tarihçesi

2007 yılında iPhone tanıtıldıktan sonra 2008 yılında Apple Service Development Kit (SDK) geliştiricilere açıldı. Böylece iOS platformu için ilk uygulamalar geliştirilmeye başlandı. Günümüze kadar hem iPhone cihazları hem de iOS platformu sürekli güncellendi. iOS 11 versiyonu 13 Eylül 2017'de kullanıcılara açılmıştır. Bu versiyonun yeni olmasından dolayı Şekil 4.2'de Apple versiyonların dağılımı grafiğini güncellememiştir. Bu nedenle grafikte iOS 11 versiyonu görünmemektedir.

87% of devices are using iOS 10.



As measured by the App Store on July 28, 2017.

Şekil 4. 2 iOS versiyonlarının dağılım grafiği [25]

4.2 iOS Platformunda Uygulama Geliştirme

iOS platformunda bir uygulama geliştirmek ve geliştirilen uygulamayı, uygulama mağazasında (App Store) yayınlamak için bazı temel gereklilikler vardır. Apple firmasına ait Mac OS işletim sistemine sahip bir bilgisayara, iOS işletim sistemine sahip bir mobil cihaza, kendi geliştirme ortamı olan ve sadece Mac OS işletim sisteminde çalışan Xcode yazılımına ve son olarak Apple tarafından satılan ve yıllık olarak yenilenmesi gereken uygulama geliştirme üyeline sahip olmak gerekmektedir. Aşağıdaki Şekil 4.3’de yukarıda bahsedilen gereklilikler şematik olarak özetlenmiştir.



Şekil 4. 3 iOS platformunda uygulama geliştirmek ve yayınlamak için gerekli cihaz ve yazılımlar

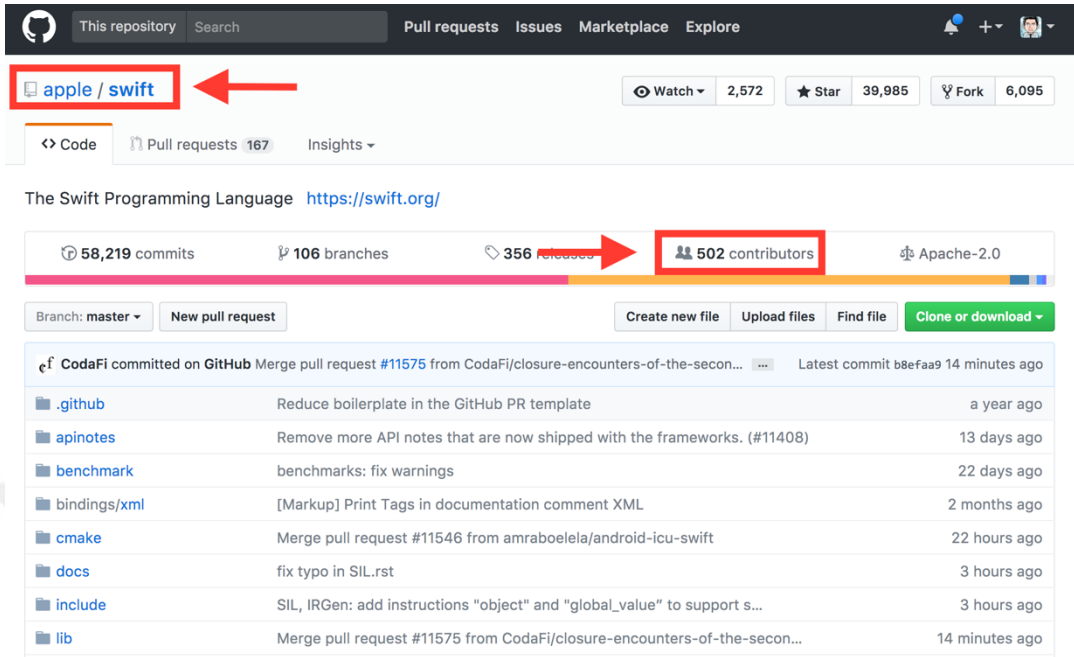
4.3 Swift Programlama Dili

Swift programlama dili günümüzdeki en yeni dillerden biridir. Apple tarafından Haziran 2014 yılında ilk kez tanıtılan bu dil, yine aynı firma tarafından geliştirilen iOS, OS X, WatchOS işletim sistemlerinde, daha güvenli, daha hızlı ve daha kolay uygulamalar geliştirmek için geliştirilmiştir.

Swift programlama dili duyurulmadan önce yukarıda bahsedilen işletim sistemlerine uyumlu uygulama geliştirmek için Objective C (Obj C) kullanılmaktaydı. Bu dil C dilinin üzerine yazılmış, nesne yönelimli bir dildir. Ancak günümüzde Swift programlama dilinin çıkmasıyla birlikte ve Obj C'nin Swift'e göre yavaş kalması ve söz diziminin (syntax) karışık olması nedeniyle önemini kaybetmeye başlamıştır.

Swift ilk kez duyurulduktan bir yıl sonra 2015'de açık kaynak kodlu olarak GitHub web sitesi üzerinden herkese açılmıştır. Şekil 4.4'de görüldüğü gibi 502 geliştiricinin katkı sağladığı Swift programlama dilinin kaynakları açık olarak paylaşılmaktadır. Apple firması; Swift dilinin geliştirici topluluklarının desteğini alarak bu dili iyileştirmeyi, kullanım oranını artırmayı, güvenlik sorunlarını ve hataları kapatmayı hedeflemiştir. Bunun bir sonucu olarak kısa sürede bu dilin Linux işletim sistemi üzerinde de

çalıştırılabilmesi sağlanmıştır. Gelecekte ise Android veya Windows gibi farklı işletim sistemlerini de desteklemesi beklenilmektedir.



Şekil 4. 4 Swift programlama dilinin açık kaynak kodları

Swift dili henüz çok yeni bir dil olmasından dolayı, hemen her yıl güncellenerek yeni versiyonu çıkmaktadır. Şu an en stabil versiyonu Swift 4'dür.

Swift, çoklu-paradigmalı (protokol yönelimli, blok yapılı vb yazılım paradigmalarını içermektedir.) bir programlama dili olmasının rağmen, çoğunlukla nesne yönelimli programlama dili olarak kullanılmaktadır. Nesne yönelimli bir uygulama geliştirmek için kodlama yaparken; sınıf, nesne, fonksiyon kavramlarını bolca kullanmak gerekir. Nesne yönelimli bir dilde; soyutlama (abstraction), devralma (inheritance), sarmalama (encapsulation) ve çok biçimlilik (polymorphism) gibi dört temel özellik bulunur. Örneğin; 'Taşıt' isimli bir sınıf oluşturulabilir ve bu sınıfın 'markası', 'modeli', 'yılı' ve 'tekerlek sayısı' gibi özellikleri olabilir. Taşıt sınıfından araba ve motosiklet alt sınıfları üretilebilir ve devralma özelliği ile taşıt sınıfında bulunan özellikler alt sınıfa aktarılabilir. Bu durumda tekerlek sayısı arabada 4, motosiklette ise 2 olacaktır. Ayrıca alt sınıf olan araba sınıfına 'kapı sayısı' özelliği de eklenebilir. Bu özellik sadece araba sınıfı için geçerli bir özellik olur. Motosikletin kapısı bulunmadığından bu özellik motosiklet sınıfına eklenmez (Şekil 4.5).

```

1 import UIKit
2
3 // 'Taşıt' sınıfı
4 class Tasit {
5
6     var markasi = "null"
7     var modeli = "null"
8     var yili = 0
9     var tekerlekSayisi = 0
10
11     init(markasi: String, modeli: String, yili: Int, tekerlekSayisi: Int) {
12
13         self.markasi = markasi
14         self.modeli = modeli
15         self.yili = yili
16         self.tekerlekSayisi = tekerlekSayisi
17     }
18
19 }
20
21 // 'Taşıt' sınıfından türetilen 'Araba' sınıfı
22 class Araba: Tasit {
23
24     init(markasi: String, modeli: String, yili: Int) {
25
26         // 'Taşıt' sınıfından türetilen 'Araba' sınıfının varsayılan tekerlek sayısı değeri '4' olarak belirlenmiştir.
27         super.init(markasi: markasi, modeli: modeli, yili: yili, tekerlekSayisi: 4)
28     }
29 }
30
31 // 'Taşıt' sınıfından türetilen 'Motorbisiklet' sınıfı
32 class Motorbisiklet: Tasit {
33
34     init(markasi: String, modeli: String, yili: Int) {
35
36         // 'Taşıt' sınıfından türetilen 'Araba' sınıfının varsayılan tekerlek sayısı değeri '4' olarak belirlenmiştir.
37         super.init(markasi: markasi, modeli: modeli, yili: yili, tekerlekSayisi: 2)
38     }
39 }
40
41 var arabam1 = Araba.init(markasi: "Mercedes", modeli: "C-180", yili: 2017)
42 print("Arabam nesnesinin tekerlek sayısı: \(arabam1.tekerlekSayisi)")
43
44 var motorbisikletim = Motorbisiklet.init(markasi: "Honda", modeli: "CBR-600RR", yili: 2016)
45 print("Motorbisikletim nesnesinin tekerlek sayısı: \(motorbisikletim.tekerlekSayisi)")
46

```

Arabam nesnesinin tekerlek sayısı: 4
Motorbisikletim nesnesinin tekerlek sayısı: 2

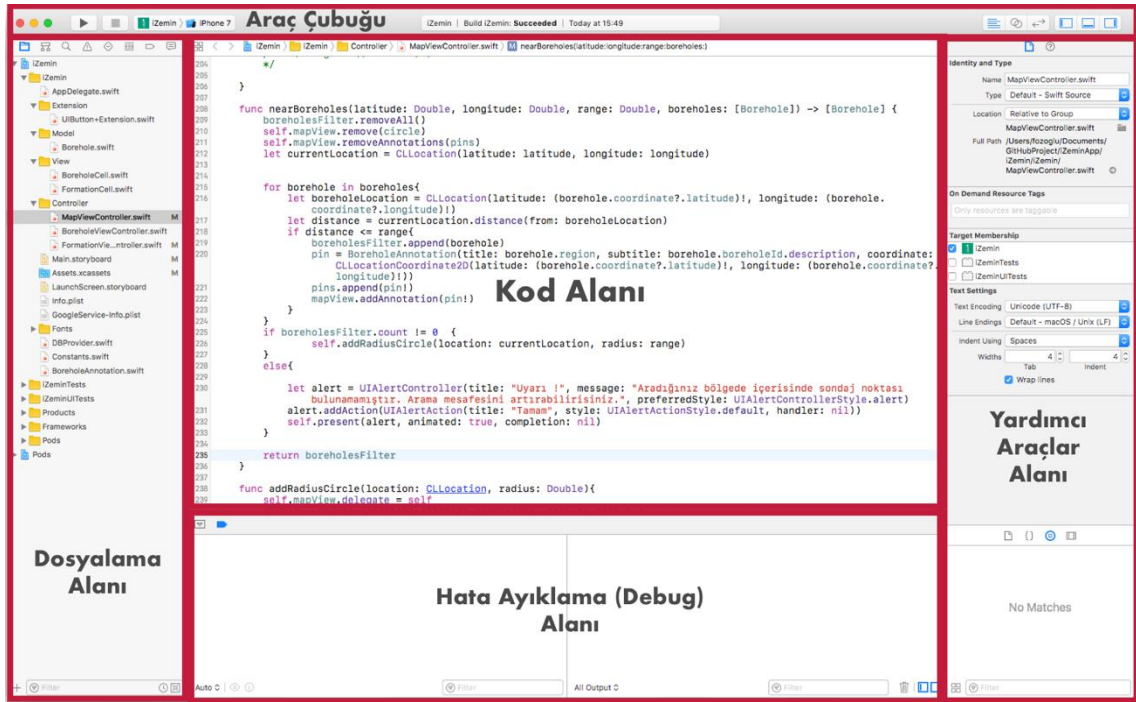
Şekil 4. 5 Swift programlama dilinde örnek sınıf yapısı

4.3.1 Xcode

Apple tarafından geliştirilen, Mac OS X işletim sistemine kurulan ve tamamen ücretsiz olan bir uygulama geliştirme yazılımıdır. Bu program ile Obj C ya da Swift dili kullanılarak iOS, OS X ve Watch OS işletim sistemlerine uygulama geliştirilmektedir. Gelişen işletim sistemi ve programlama diliyle birlikte Xcode yazılımı da sürekli olarak güncellenmektedir ve şu an 9. versiyonu kullanımdadır (Şekil 4.6).

Aşağıdaki linkten indirilebilmektedir.

<https://itunes.apple.com/tr/app/xcode/id497799835?l=tr&mt=12>

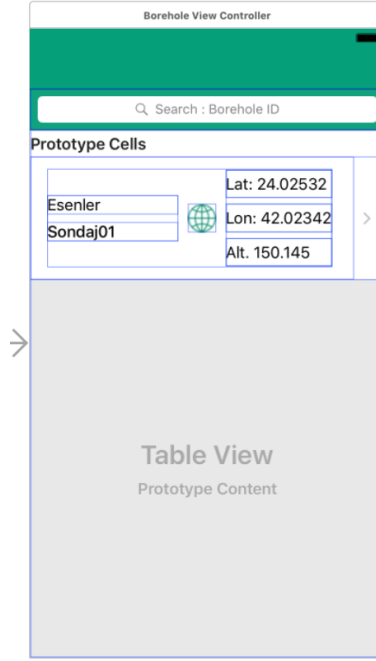


Şekil 4. 6 xCode yazılımının kullanıcı arayüzü

4.3.1.1 Storyboard

Storyboard, geliştirilen uygulamanın tasarımında ve akışında büyük kolaylıklar sağlamaktadır. Bir mobil uygulama geliştirme platformunda, storyboard gibi kullanıcı arayüzü oluşturma araçları olmasaydı, mobil uygulama geliştiricileri tasarımda yer alan butonları, görselleri, listeleri vb. diğer arayüz bileşenlerini ekrana kod ile oluşturması gerekirdi. Ancak storyboard yardımıyla çok daha az kod yazılarak geliştirme yapılabilmektedir. Bazı istisnai durumlarda, farklı zamanlarda ekranda oluşturulacak bileşenleri kod ile üretmemiz gerekebilmektedir.

Xcode yazılımında projeyi oluşturduğumuzda otomatik olarak 'main.storyboard' isimli bir dosya oluşturulur. Bu dosya ana storyboard dosyasıdır. Ancak geliştirici duruma bağlı olarak birden fazla storyboard dosyası oluşturabilir ve kullanabilir. Storyboard dosyası içerisinde önce kodlaması yapılacak ekranların oluşturulması ve sonra bu ekranların içerisinde yer alacak bileşenlerin yerleştirilmesi gerekmektedir (Şekil 4.7).

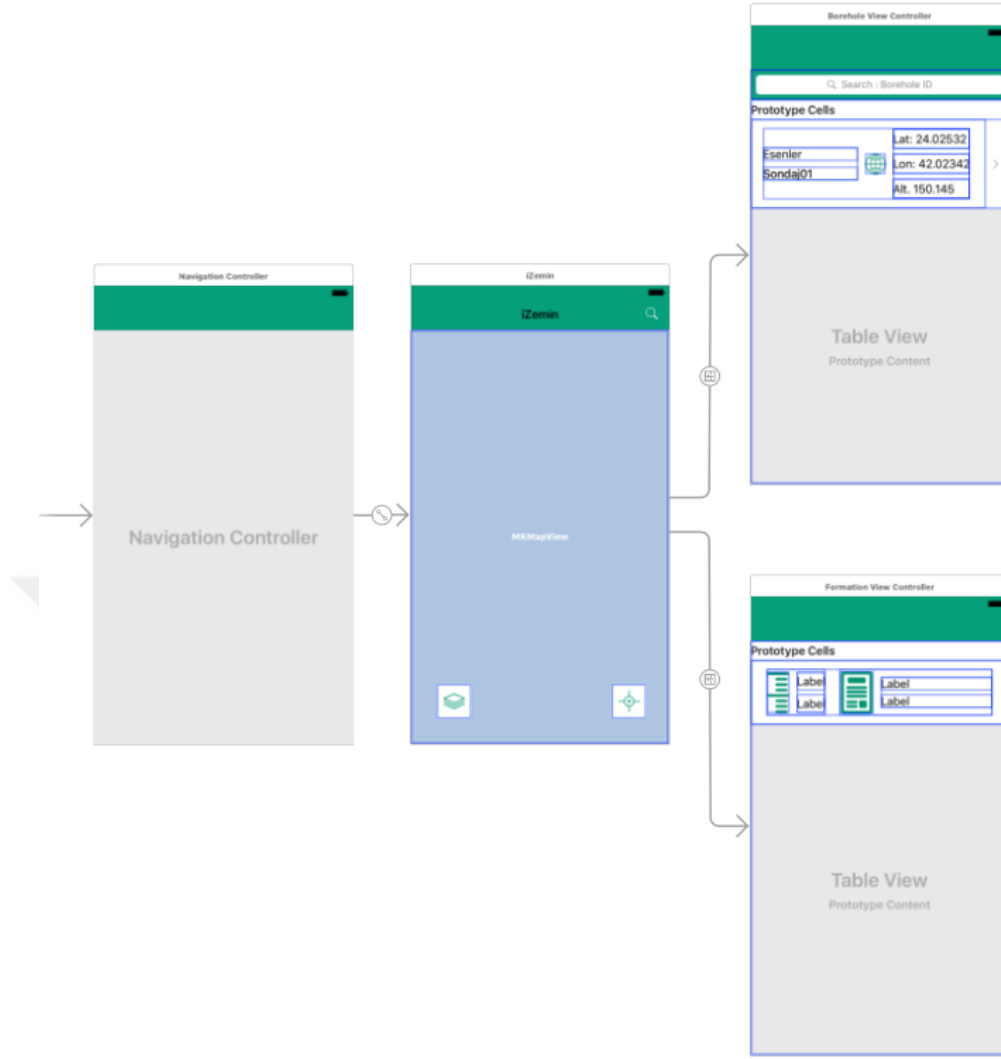


Şekil 4. 7 Storyboard'daki ekranlardan birine eklenen bileşenler

Hazırladığımız ekranlara işlevsellik kazandırmak için ekranlar arasında bağlar (segue) oluşturulur. Herhangi bir durumda tetiklenmesi istenen alanlara kod blokları tanımlanır. Örneğin bir butona dokunulduğunda ekranda bir uyarı göstermek istenirse storyboard'da yer alan butonun dokunulma olayına uyarı kodları tanımlanmalıdır. Aşağıdaki Şekil 4.8'de göstermiştir.

4.4 iOS Platformunda Uygulama Geliştirme Kütüphaneleri

iOS platformunda uygulama geliştirirken kullanılmak istenen veya gerekli olan özellikleri projeye eklemek için kütüphanelerden yararlanılır. Bu kütüphaneler yerel ve harici (3. parti) kütüphaneler olmak üzere ikiye ayrılır.



Şekil 4. 8 Storyboard'daki ekranlar ve ekranlar arasındaki bağlantılar

4.4.1 Yerel (Lokal) Kütüphaneler

Bu kütüphaneler işletim sistemi ve xCode yazılımı ile birlikte gelen kütüphanelerdir. Genellikle temel özellikleri kullanmamızı sağlar.

Bir mobil cihaz üzerinde çeşitli alıcı ve sensörler bulunmaktadır. Örneğin bir iPhone 7 cihazında bulunan alıcı ve sensörler aşağıda sıralanmıştır;

- GNSS (uydu sinyalleri) alıcısı
- Parmak izi okuyucusu
- Üç eksenli jiroskop

- İvmeölçer
- Yakınlık sensörü
- Ortam ışığı sensörü
- Barometre

Yukarıda bahsedilen cihaz üzerindeki sensör ve alıcılara kütüphaneler yardımıyla erişim sağlanır. Kısacası, kullanılmak istenen özelliğin kütüphanesi projeye eklenerek ve kütüphanelerin kod yazma talimatlarına uyularak, ilgili alıcı ve sensör geliştirilmekte olan uygulamaya uyumlu hale getirilmiş olur. Örneğin, bir geliştirici, projesine harita ekleyip, cihazın konumunu harita üzerinde göstermek istiyorsa, işletim sistemi üzerinde gelen CoreLocation ve MapKit kütüphanelerini eklemelidir. En temel yerel kütüphaneler Çizelge 4.1’de verilmiştir.

4.4.2 Harici (3. Parti) Kütüphaneler

Harici kütüphaneler, firma veya şahıslar tarafından geliştirilen ve genellikle geliştirme aşamasındaki ihtiyaçları karşılamak ve geliştiriciye kolaylık sağlamak için hazırlanmıştır. Bazı kütüphaneler çok büyük ve kapsamlı olabilirken, bazıları ise sadece çok küçük ve kritik bir noktayı çözmeye yönelik olabilmektedir. Örneğin, Google’ın geliştiricilere sağladığı Firebase kütüphanesi ile verilerini bulutta saklayabilmekte, uygulamayı analitik olarak takip edebilmekte veya uygulamada oluşan hataları gösterebilmekteyken; bir başka kütüphane ise sadece cihazın tipi, versiyonu gibi bilgileri almak için kullanılmaktadır. Geliştiriciler kendi kütüphanelerini yazıp yayınlatabileceği gibi dışarıdan hazır kütüphaneleri de kullanabilmektedir.

Yerel kütüphanelerin aksine işletim sistemi içerisinde bulunmamasından dolayı projeye dışarıdan harici olarak eklemek gerekmektedir. Hem ekleme hem de kullanma açısından farklılıklar gösterebilmektedir. En temel ekleme yöntemi kütüphane dosyalarının doğrudan projeye eklenmesidir. Ancak iOS platformunda son zamanlarda en sık kullanılan yöntem “CocoaPods” paket yönetim sistemi ile ekleme yöntemidir. Bu yöntem ayrı bir başlıkta ayrıntılı olarak ele alınmıştır.

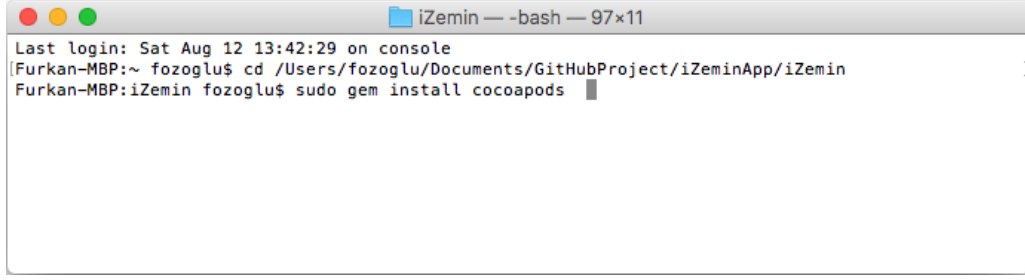
Çizelge 4. 1 iOS Platformunda Sık Kullanılan Kütüphaneler [26]

LOKAL KÜTÜPHANELER	GÖREVİ
UIKit	Kullanıcı arayüzü katmanı için temel sınıf ve metotları içerir.
Foundation	Temel işlevleri tanımlamak için kullanılır. Veri depolaması, metin işleme, hesaplama, sıralama gibi işlevler sağlar.
MapKit	Apple haritası ve geocoding işlemleri içerir.
WebKit	Uygulama içerisinde web sitesi içeriklerini göstermek için kullanılır.
Corelocation	GNSS alıcısından elde edilen koordinat değerlerini sağlar.
CoreData	Cihaz üzerinde lokal bir veri tabanı oluşturur ve verilerin depolanmasını sağlar.
MediaPlayer	Uygulama içerisinde tam ekran video oynatmayı sağlar.
MessageUI	Uygulama içerisinden e-mail göndermeyi sağlar.
HealthKit	Cihazın içerisinde tanımlı kişinin sağlık veya ilişkili bilgisini okumayı yazmayı sağlar. Adım, tansiyon, kan şekeri vb.
AddressBook	Cihaz içerisindeki kişi rehberinde bulunan iletişim bilgilerini okumayı veya düzenlemeyi sağlar.

4.4.2.1 CocoaPods

CocoaPods, firmalar ve şahıslar tarafından eklenmiş binlerce ve çok çeşitli konularda kullanılabilecek hazır kütüphaneleri barındıran bir platformdur.

CocoaPods kullanmadan önce Mac OS X işletim sistemli cihaza bir defaya mahsus olmak üzere kurulması gerekmektedir. Şekil 4.9'daki gibi Terminal yazılımında 'sudo gem install cocoapods' yazılarak bilgisayara kurulum sağlanmaktadır. Kurulum tamamlandıktan sonra Şekil 4.10'daki gibi proje içerisinde bir podfile (pod dosyası) oluşturularak, kurulması istenen kütüphanelerin isimleri listelenerek yükleme işlemi yapılmaktadır (Şekil 4.11).



```
iZemin — -bash — 97x11
Last login: Sat Aug 12 13:42:29 on console
Furkan-MBP:~ fozoglu$ cd /Users/fozoglu/Documents/GitHubProject/iZeminApp/iZemin
Furkan-MBP:iZemin fozoglu$ sudo gem install cocoapods
```

Şekil 4. 9 CocoaPods'un bilgisayara kurulumu



```
iZemin — nano podfile — 80x21
GNU nano 2.0.6 File: podfile Modified

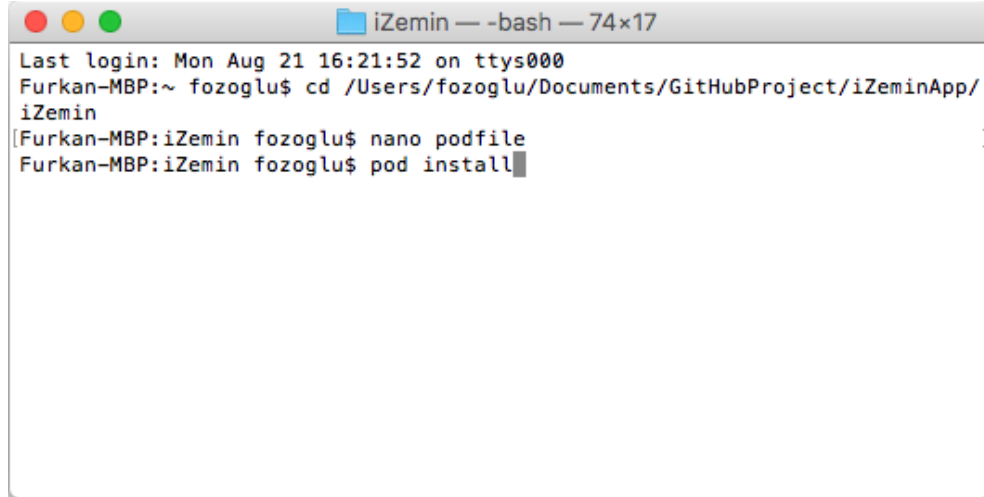
target 'iZemin' do

  use_frameworks!

  pod 'Firebase'
  pod 'Firebase/Storage'
  pod 'Firebase/Database'

end
```

Şekil 4. 10 CocoaPods pod dosyasının düzenlenmesi



```
iZemin — -bash — 74x17
Last login: Mon Aug 21 16:21:52 on ttys000
Furkan-MBP:~ fozoglu$ cd /Users/fozoglu/Documents/GitHubProject/iZeminApp/iZemin
Furkan-MBP:iZemin fozoglu$ nano podfile
Furkan-MBP:iZemin fozoglu$ pod install
```

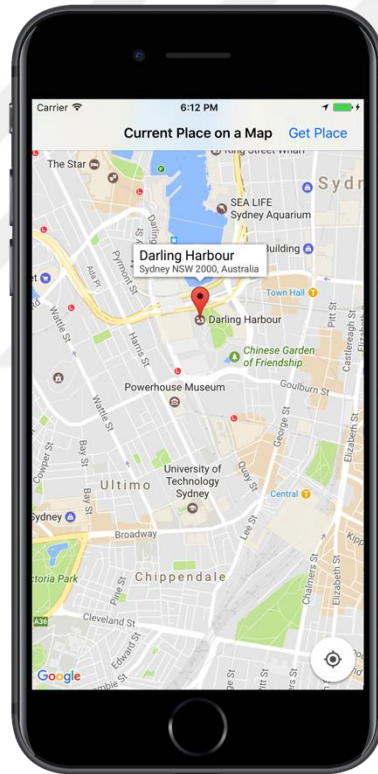
Şekil 4. 11 Pod dosyasındaki kütüphanelerin projeye yüklenmesi

4.4.2.2 Harita Tabanlı Kütüphaneler

iOS platformunda yerel MapKit kütüphanesi dışında farklı firmaların geliştirmiş oldukları harici harita kütüphaneleri de mevcuttur. Bu kütüphaneler genel olarak aynı temel

özellikleri (GNSS alıcısı yardımıyla cihaz konumunu harita üzerinde göstermek ve çeşitli atlık harita opsiyonları sağlamak) barındırmaktadır. Aşağıda iOS platformunda yaygın olarak kullanılan harita tabanlı kütüphanelerden bahsedilmiştir.

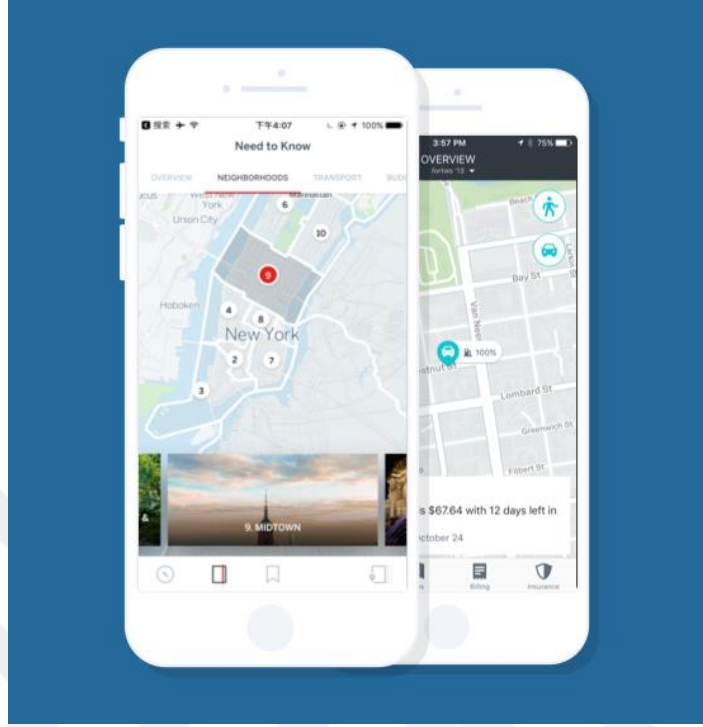
Google Maps iOS SDK: Google tarafından geliştirilmiş olan ve Apple iOS cihazlarda MapKit kütüphanesi geliştirilmeden önce varsayılan olarak kullanan harita kütüphanesidir (Şekil 4.12). Google haritalarının çok ayrıntılı ve kapsamlı bir dokümanı vardır. Şu an en güncel versiyonu 7.3 dür [27]. Cocoapods ile projeye kolay bir şekilde eklenebilmektedir. Google haritalarının gösterimi ve içerisinde yer alan geocoding, rota gibi hizmetler belirli sınırlar içerisinde ücretsiz olarak geliştiricilere sağlanmaktadır. Ancak bu sınırların aşılması durumunda Google geliştiricilerden ücret talep etmektedir.



Şekil 4. 12 Google Maps iOS SDK'dan örnek ekran görüntüsü

MapBox iOS SDK: En güncel versiyonu 3.6 olan kütüphanenin kodları açık olarak GitHub üzerinde yayınlanmıştır ve CocoaPods ile projeye eklenebilmektedir [28]. İçerisinde hazır harita stilleri, Çevrimdışı (offline) haritalar, vektör haritalar bulunmaktadır. Açık kaynak kodlu olmasından dolayı temel özellikler ücretsiz olsa da kullanımda bazı sınırlamalar (ayda 50 bin harita gösterimi, dakikada 600 geocoding isteği ve 250 mb veriseti

depolaması vb.) uygulanmaktadır. Bu sınırlamaları artırmak için ücretli versiyona geçiş yapılmalıdır (Şekil 4.13).



Şekil 4. 13 Mapbox iOS SDK'dan örnek ekran görüntüsü

ArcGIS Runtime iOS SDK: Haziran 2017'de güncellenmiş versiyonu 100.1'dir [29] (Şekil 4.14). Kütüphane CocoaPods ile projeye eklenebilir. Yukarıda bahsedilen kütüphanelerden farklı özelliklere sahiptir. Çünkü Esri firmasının geliştirdiği Dünyada yaygın olarak kullanılan CBS yazılımlarından biri olan ArcGIS yazılımının içerisinde bulunan özelliklerin mobil SDK'ya aktarılmasıyla birlikte kendi eko sistemleri içerisinde entegre bir yapı oluşturulmuştur. Örneğin; ArcGIS server'da bulunan harita servisleri kullanılarak kolay bir şekilde mobil ortama aktarılabilir.

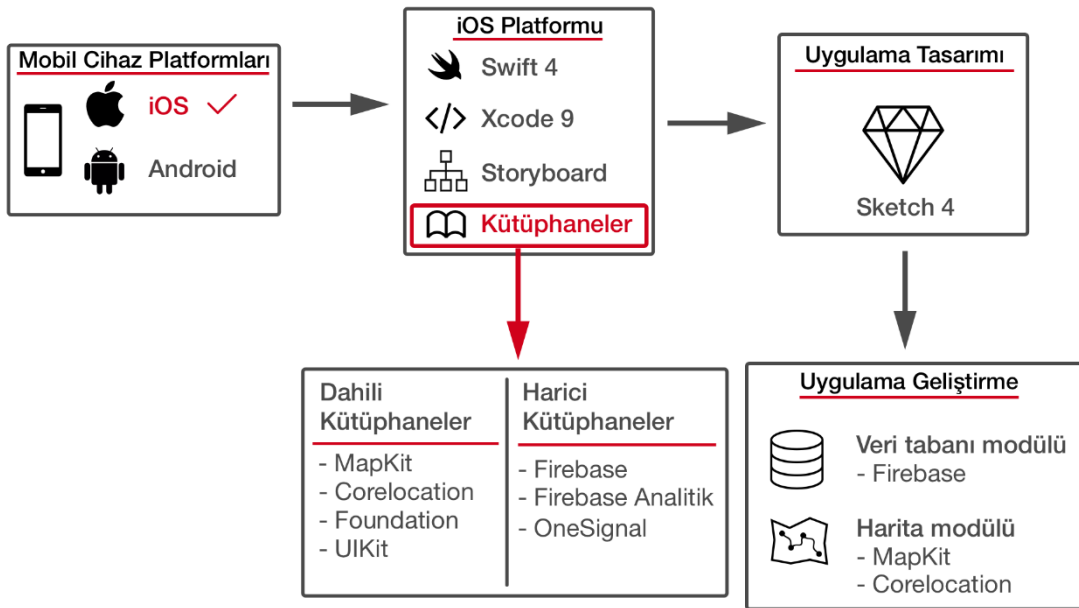


Şekil 4. 14 ArcGIS Runtime iOS SDK'dan örnek ekran görüntüsü

BÖLÜM 5

UYGULAMA

Bu çalışma kapsamında geliştirilen iOS tabanlı mobil uygulama, izemin olarak isimlendirilmiştir. Çalışma içerisinde bundan sonraki bölümlerde bu isimle anılacaktır. Ayrıca çalışmanın çerçevesi Şekil 5.1’de gösterilmiştir. Bu çerçevede; temel özelliklerin belirlenmesi, tasarımının hazırlanması ve uygulamanın geliştirilmesi başlıkları altında açıklanmıştır.



Şekil 5. 1 Çalışmanın çerçevesi

5.1 Temel Özelliklerin Belirlenmesi

5.1.1 Projenin Oluşturulması

iZemin mobil uygulaması Xcode yazılımında projeyi oluşturduktan sonra proje hedefleri doğrultusunda temel ayarlar yapılmıştır. Uygulamanın hem güncel hem de eski versiyonları desteklemesi için proje ayarlarından iOS versiyonu 9.3 olarak seçilmiştir. Bu sayede Dünyadaki iOS platformu kullanıcılarının %93'ü bu uygulamayı sorunsuz bir şekilde cihazlarına yükleyebilecektir. Ayrıca geliştirilen uygulama iOS platformunda sadece telefonlara uyumlu olacağı için sadece Apple firmasının ürettiği iPhone cihazlarında çalışacak şekilde ayarlanmıştır. Bu çalışmada Çizelge 5.1'de teknik özellikleri verilen iPhone 7 cihazı test cihazı olarak kullanılmıştır. Arayüz kısmında kullanıcak ikonlar ve görseller projeye eklenmiştir.

Çizelge 5. 1 iPhone 7 donanım özellikleri

İşlemci	Apple A7
Bellek	2048 MB LPDDR3 RAM
Ekran Çözünürlüğü	750x 1334 pixels (326 ppi)
GNSSa	A-GPS, GLONASS
Accelerometer	3-Axis accelerometer
Gyroscope	3-Axis Gyroscope

5.1.2 Kütüphanelerin Eklenmesi

iZemin uygulaması projesine ilgili yerel kütüphaneler eklenmiştir ve harici kütüphaneler de cocoapods entegrasyonu ile ilave edilmiştir. Projeye eklenen Corelocation ve MapKit kütüphaneleri harita ekranında kullanılmıştır. Cihaz konumunun, 2B harita veya uydu görüntüsü üzerinde gösterilmesi sağlanmıştır. Foundation ve UIKit kütüphaneleri mobil uygulama geliştirmede kullanılan temel fonksiyonları ve görsel bileşenleri içermektedir. Harici olarak eklenen Firebase kütüphanesi bize hem veri tabanı özelliğini kullanmayı hem de iZemin mobil uygulamasının kullanıma ait analitik değerlerini tutmayı sağlamıştır. OneSignal kütüphanesi ise kullanıcılara anlık bildirim (güncelleme ve

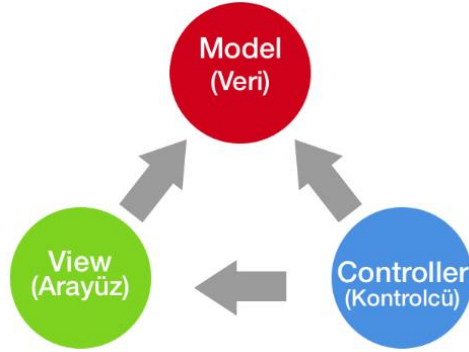
bilgilendirme duyuruları vb.) gönderebilmek ve gönderilen bildirimleri takip etmek için eklenmiştir. Çizelge 5.2’de kullanılan kütüphaneler ve kullanım amaçları özetlenmiştir.

Çizelge 5. 2 iZemin uygulamasının geliştirilmesinde kullanılan kütüphaneler

Kütüphaneler	Sınıflandırma	Kullanım	Detay
Corelocation	Yerleşik	Harita	GNSS Alıcısı
MapKit	Yerleşik	Harita	2B haritalar ve uydu görüntüleri
Firebase	Harici	Veri tabanı	Bulut tabanlı veri tabanı
Foundation	Yerleşik	iOS uygulaması temelleri	iOS platformu için temel fonksiyonlar
UIKit	Yerleşik	iOS uygulaması temelleri	iOS platformu için temel arayüz bileşenleri
OneSignal	Harici	Genel	Anlık bildirim gönderme servisi
Firebase Analitik	Harici	Genel	Kullanım istatistikleri tutulur.

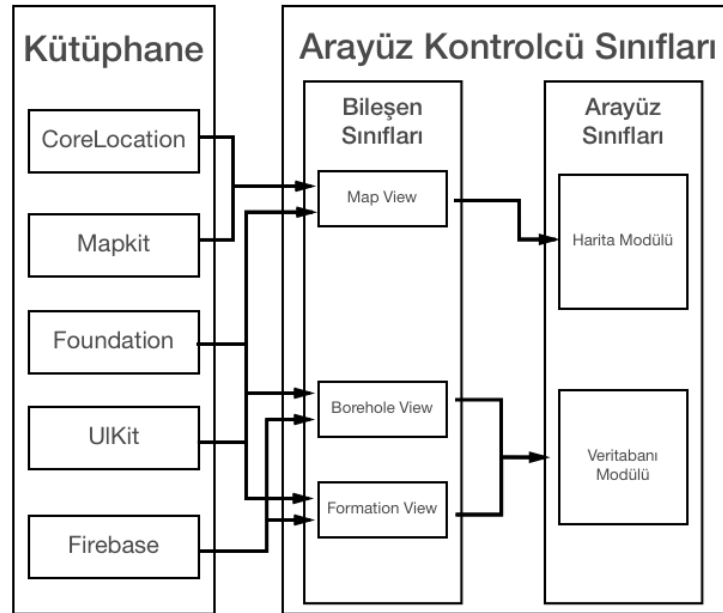
Bölüm 4’te açıklanan, mobil uygulama geliştirmede yaygın olarak kullanılan MVC mimarisi için xCode yazılımında kodlamaya başlamadan önce proje dosyaları ve klasörleri hazırlanmıştır. Bu şekilde hazırlanması bize geliştirme esnasında aradıklarımızı kolay bir şekilde bulmayı ve yönetmeyi sağlamıştır (Şekil 5.2).

Veri - Arayüz - Kontrolcü Mimarisi
Model View Controller (MVC)



Şekil 5. 2 Veri-Arayüz-Kontrolcü Mimarisi (MVC)

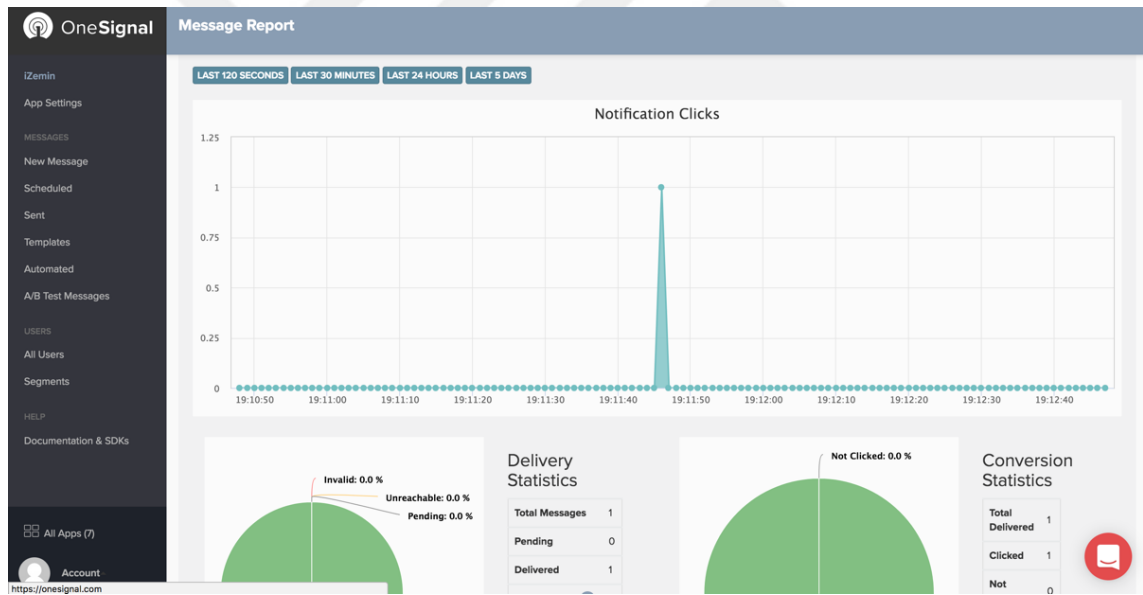
Geliştirilen iZemin uygulamasında dört adet kontrolcü vardır. Üçü uygulamanın temel ekranlarını temsil ederken, biri ekranlar arasında geçişleri kontrol eder. Kullanılan kütüphanelerin arayüz kontrolcü sınıfları ile ilişkileri Şekil 5.3’de gösterilmiştir. Bu kontrolcüler içerisinde bir ana arayüz nesnesi bulunmaktadır ve bu arayüz nesnesi içerisinde; buton, liste, resim ve harita nesneleri bulunmaktadır. IBAction ve IBOutlet bağlantı sınıfları ile arayüz ve kontrolcü arasında bir köprü oluşturulmaktadır.



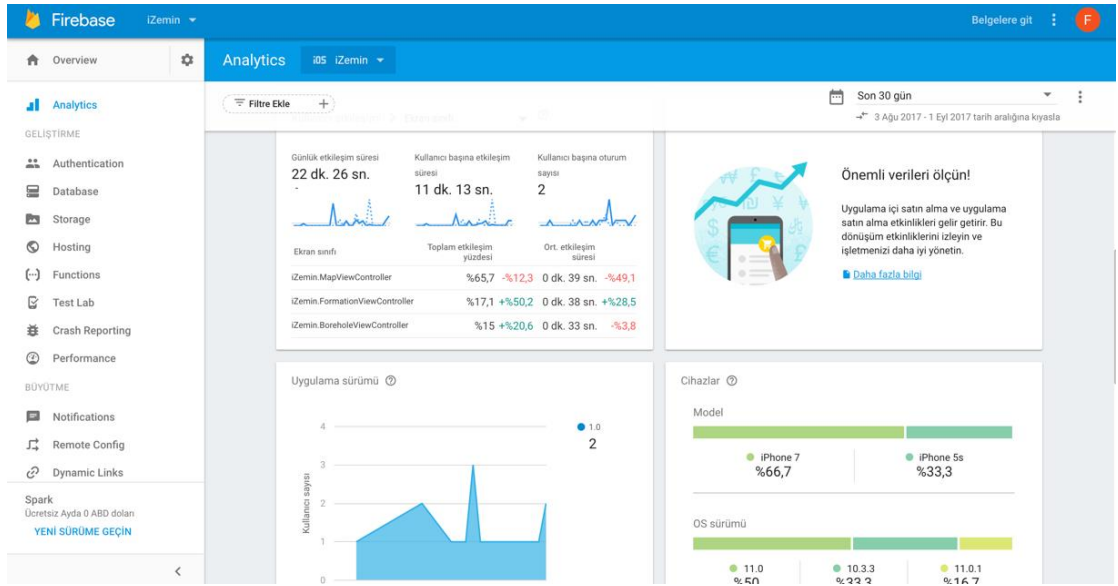
Şekil 5. 3 iZemin’de kullanılan kütüphanelerin arayüz sınıfları ile ilişkileri

iZemin mobil uygulamasında, kullanıcıların uygulama üzerindeki etkileşimini artırmak amacıyla, bildirim gönderme özelliği eklenmiştir. Bu özellik sayesinde kullanıcılara belirli

bir zaman aralığında ya da anlık olarak bildirim gönderilebilmektedir. Bildirim gönderme özelliği için Onesignal firmasının geliştiricilere ücretsiz olarak sağladığı servis kullanılmıştır. Web sitesi üzerinden göndermek istediğimiz bildirimlerin istediğimiz ayarlar ile birlikte gönderilmesini sağlamaktadır. Gönderilen bildirimin kaç kişiye ulaştığı, ulaştırılan bildirimlere dokunulup dokunulmadığı bilgisini takip edebileceğimiz Onesignal arayüzü Şekil 5.4’de verilmiştir. İZemin mobil uygulamasında etkileşimi artırmak için kullanılan bir diğer özellik ise, uygulamanın kullanımını inceleyen analitik özelliktir. Bu özellik ile uygulamanın ne kadar süre ve hangi cihazlarda kullanıldığı, bu cihazların hangi versiyonda oldukları ile ilgili sayısal istatistikler toplanmaktadır. Bunun için Firebase içerisinde yer alan uygulamanın analitik olarak takibini yapma imkânı veren özelliği kullanılmıştır (Şekil 5.5). Bu istatistiklerin özellikle geliştiricilere sağladığı avantaj, uygulamaya yapılacak güncelleme ve güncellemede kullanıcılara sunulacak ek özelliklerin daha iyi analiz edilmesidir.



Şekil 5. 4 OneSignal bildirim takip ekranı



Şekil 5. 5 Firebase analitik takip ekranı

5.2 Tasarımın Hazırlanması

5.2.1 Mobil Cihazlarda Uygulama Tasarımı

Masaüstü uygulamaların tasarımı sadece bilgisayar ekranına odaklanırken, web tabanlı uygulama tasarımları ise hem bilgisayar hem de mobil cihaz ekranları göz önüne alınarak gerçekleştirilir. Önceleri, web tabanlı uygulamaların mobil cihazlarda kullanımı durumunda, ayrı bir uygulama tasarımı geliştirilmezken, yeni web tabanlı uygulamalarda mobil cihazlar için ayrı tasarımlar da yapılmaktadır. Ancak buradaki geliştirme yöntemi web tabanlı arayüz tasarımının mobil cihaz arayüz tasarımına dönüştürülmesi şeklindedir. Masaüstü ve web tabanlı uygulamaların aksine, mobil cihazlar için geliştirilen uygulamaların tasarımları daha da önem arz etmektedir. Mobil cihaz ekranlarının bilgisayar ekranlarına göre daha küçük (3.5, 4, 4.7, 5.5 inç vb.) olması bunun başlıca sebebidir. Yukarıda değinildiği gibi kimi mobil cihaz tasarımları, doğrudan web site tasarımlarından dönüştürüldükleri halde, özgün (native) mobil uygulamalar ve tasarımları belirli ve/veya bütün mobil cihazlarda görüntülenecek şekilde gerçekleştirilir. Mobil cihazlara odaklı uygulama tasarımında en küçük ekrandan (3.5 inç) başlayarak bütün mobil cihazlara uyumlu tasarımlar yapılmaktadır.

Mobil cihaz ekranlarının dokunmatik olmasından dolayı diğer uygulamalara göre tasarımda farklılıklar bulunur. Mouse kullanılarak etkileşimde bulunulan uygulama

tasarımları ile dokunma ile etkileşimde bulunan mobil uygulama tasarımları birbirinden oldukça farklıdır. Örneğin, parmaklarımızın mouse imlecinden büyük olması tasarımda buton boyutlarının bu farklılık gözetilerek belirlenmesini gerektirmektedir.

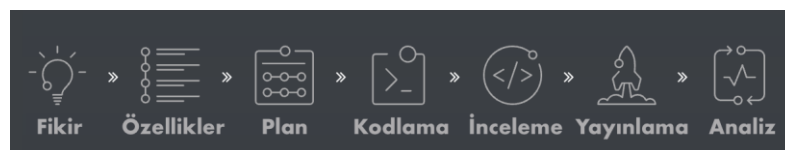
5.2.2 İZemin Mobil Uygulama Tasarımı

Geliştirilen mobil uygulamanın, en küçük ekran boyutundan (iPhone 4S, 3.5 inç) en büyük ekran boyutuna (iPhone X, 5.8 inç) sahip tüm cihazlara uyumlu olması hedeflenmiş ve dokunmatik ekranın verdiği avantajlar kullanılarak bir uygulama tasarlanmıştır. Bu tasarımda, Apple tarafından geliştiriciler için hazırlanan arayüz tasarım ilkeleri dokümanından yararlanılmıştır [30]. İlgili dokümanda, mobil uygulama kullanıcısının uygulamayı hızlı kavraması ve rahat kullanmasına yönelik bilgiler yer almaktadır.

Mobil uygulama tasarımı Sketch yazılımı kullanılarak gerçekleştirilmiştir. Bu yazılım sayesinde, öncelikle uygulamanın wireframe (karakalem, taslak) çizimleri mobil cihaz arayüzlerine dönüştürülür. Bu yazılımda üretilen arayüzler ayrıca, mobil uygulamanın nihai durumunu gösterir. İkinci aşamada ise tasarımın görsel bileşenleri (ikonlar, renk kodları vb.) geliştirme platformuna (xCode yazılımı) aktarılır.

5.3 Uygulamanın Geliştirilmesi

Bir mobil uygulama geliştirme sürecinde uygulama fikri kabul edildikten sonra uygulamanın içerisinde yer alacak özellikler belirlenir ve listelenir. Yapılacak işler planlanır. Tüm planlamalar tamamlandıktan sonra kodlamaya geçilir. Kodlama işlemleri bitirildikten sonra geliştirilen uygulama incelenir ve test edilir. Eğer yayınlanmak için hazırsa uygulama yayınlanır, değilse kodlama aşamasında geri dönülür ve hatalar düzeltilir. Uygulama yayınlandıktan sonra uygulama takip edilmeye devam edilir. Yapılan geri dönüşler doğrultusunda uygulamaya düzeltmeler yapılır veya ilave özellikler kazandırılır (Şekil 5.6).

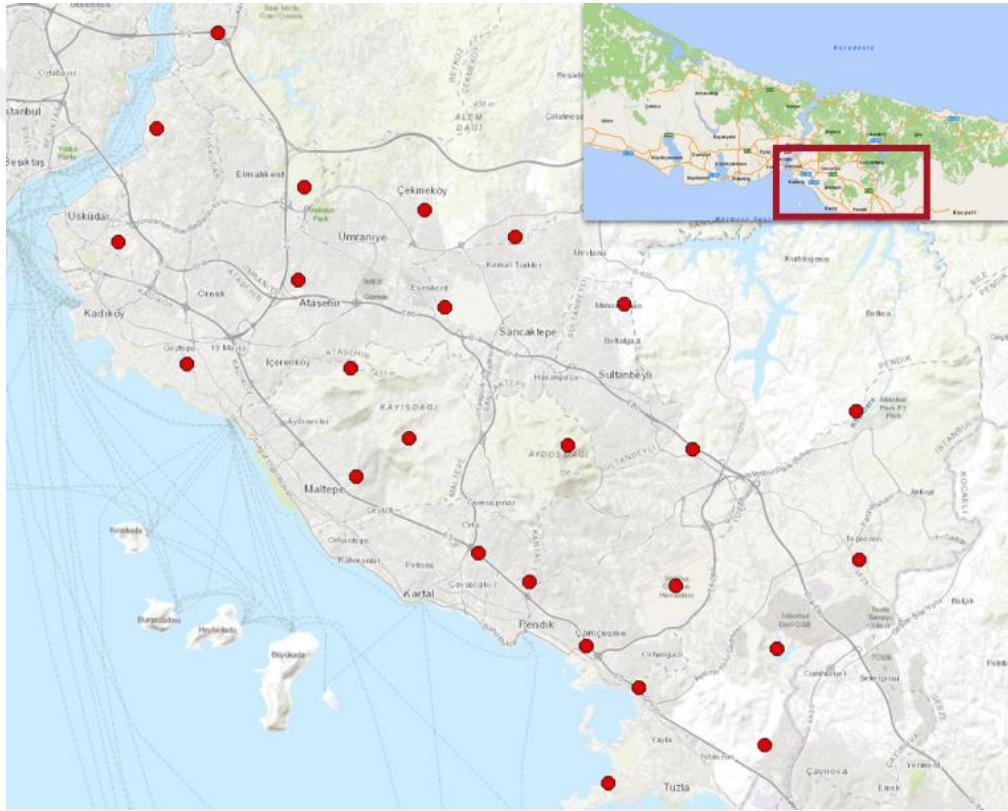


Şekil 5. 6 Uygulama geliştirme süreci

5.3.1 Veri Tabanı Modülü

5.3.1.1 Uygulama Alanı

iZemin mobil uygulaması kapsamında kullanılacak veriler, İstanbul Büyükşehir Belediyesi (İBB) Deprem ve Zemin İnceleme Müdürlüğünden alınmıştır. Şekil 5.7’de görüldüğü gibi İstanbul Anadolu yakasına ait 25 farklı noktadan oluşan bir sondaj veri setidir. Belirtilen bölgenin geniş bir alanda olması sondaj verilerinde bulunan formasyonların çeşitli olmasını ve uygulama içerisindeki Tampon Bölge (Buffer) analizi gibi işlemlerin test edilebilirliğini artırmıştır.



Şekil 5. 7 Uygulama alanı

5.3.1.2 Sondaj Verilerin Düzenlenmesi

İBB’den alınan örnek veri seti shape formatındadır. Veri seti içerisindeki tabloda koordinat ve formasyon bilgileri yer almaktadır. iZemin uygulamasıyla aktarımı sağlamadan önce mevcut veriler üzerinde düzenleme yapılmıştır. Bu bağlamda, hazırlanacak olan veri tabanında yer alacak öznitelikler belirlenmiştir. İBB veri setindeki

özniteliklerin tümü iZemin uygulamasında etkin bir şekilde gösterilemeyeceği için bazı temel öznitelikler seçilmiştir (Çizelge 5.3).

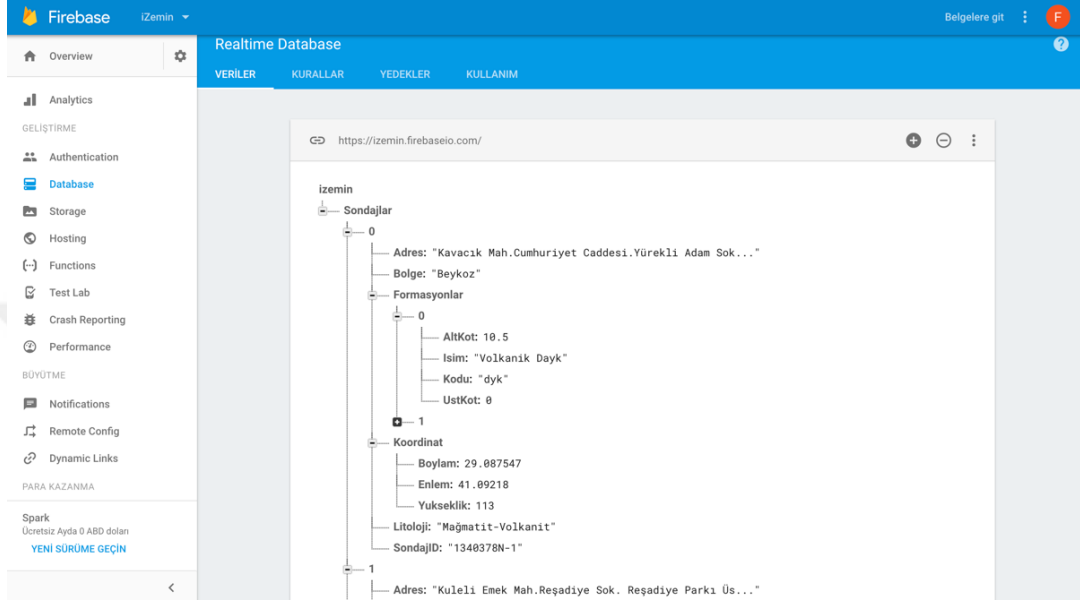
Çizelge 5. 3 Uygulamada kullanılan öznitelikler ve veri tipleri

Öznitelikler	Veri Tipleri
SondajID	String (hem yazı hem rakam)
Bölge	String
Adres	String
Litoloji	String
(Koordinat) Enlem	Double
(Koordinat) Boylam	Double
(Koordinat) Yükseklik	Double
(Formasyon) Alt Kot	Float
(Formasyon) Üst Kot	Float
(Formasyon) İsmi	String
(Formasyon) Kodu	String

5.3.1.3 Verilerin Uygulama ile Entegrasyonu

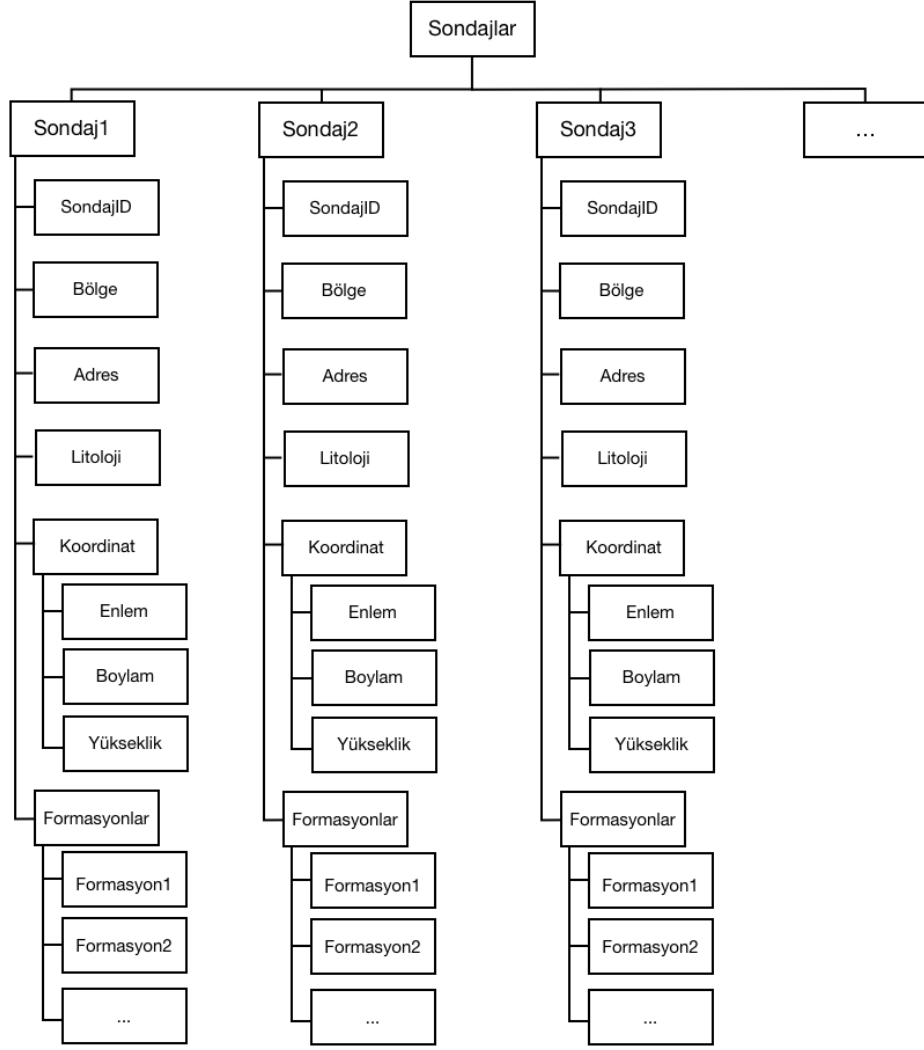
iZemin mobil uygulamasında kullanılacak sondaj verilerinin depolanması için bir veri tabanına ihtiyaç duyulmaktadır. Özellikle son zamanlarda bulut teknolojisinin yaygınlaşmasıyla birlikte geliştirilen uygulamalar bulut teknolojisi üzerinden çalıştırılmaktadır. İnternet ve sanal bilgisayarlar (virtual machine) üzerinde tutulan veriler mobil uygulamalarda kullanabilmektedir. iZemin mobil uygulamasında -diğer çalışmalardan farklı olarak- SQLite, Realm, CoreData vb. lokal olmayan bir veri tabanı yerine, bulut tabanlı bir veri tabanı tercih edilmiştir. Mobil cihaz içinde bulunan lokal veri tabanındaki veriler gerçek zamanlı olarak güncellenememektedir. Verilerin güncellenme ihtiyacı olması durumunda bu durum bir dezavantaj olarak karşımıza çıkmaktadır.

iZemin mobil uygulamasında Google firmasının geliştiriciler için sağlamış olduğu kısmen ücretsiz bir bulut veri tabanı olan Firebase kullanılmış olup iOS kütüphanesi harici olarak projeye eklenmiştir. Şekil 5.8’de Firebase’in web sitesi arayüzü görünmektedir. Bu arayüz sayesinde internet tarayıcısı üzerinden veri tabanı yönetiminin yanı sıra veri düzenleme işlemleri de yapılabilmektedir.



Şekil 5. 8 Firebase veri tabanını görüntüleme ve düzenleme ekranı

İBB’den alınan verileri iZemin mobil uygulamasında kullanmak için Firebase’de oluşturduğumuz veri tabanına aktarmadan önce belirlediğimiz öznelilikler doğrultusunda bir veri tabanı şeması oluşturulmuştur (Şekil 5.9). Hazırlanan bu şemadan görüleceği üzere veri, obje yapısındadır. Yani bir tablo düzeninde ve ilişkisel veri tabanı şeklinde değildir. Firebase bir NoSQL veri tabanıdır ve veriler JavaScript Object Notation (JSON) formatında tutulmaktadır. Şema doğrultusunda hazırlanan veriler JSON formatına uyumlu bir şekilde Firebase’e yüklenmiştir (Şekil 5.8).



Şekil 5. 9 Firebase veri tabanında oluşturulan veri şeması

Firestore'in bir diğer özelliği ise tutulan verilerin tiplerini otomatik olarak belirlemesidir. Geliştirilen programda oluşturulan Sondaj sınıfı bu veri tiplerine uyumlu olacak şekilde hazırlanmıştır. Böylece kodlama tarafında ilçe adı gibi bilgiler sözel bilgiler 'String' tipinde tutulurken, koordinat bilgileri ise 'Double' tipinde tutulmaktadır.

iZemin mobil uygulaması çalıştırıldıktan sonra internetten verilerin indirilmesi gerekmektedir. Bu nedenle xCode'da 1 adet sondaj (Borehole.swift) sınıfı (Ek-A) ve 1 adet Singleton pattern'inde yazılmış olan DBProvider sınıfı (Ek-B) yazılmıştır. Uygulama açıldığında en güncel verileri almak için DBProvider sınıfı veri tabanına istekte bulunur ve gelen verileri sondaj sınıfında bir dizide tutar. Böylece dizideki veriler harita üzerinde veya sondajların listelendiği ekranda kullanılmıştır.

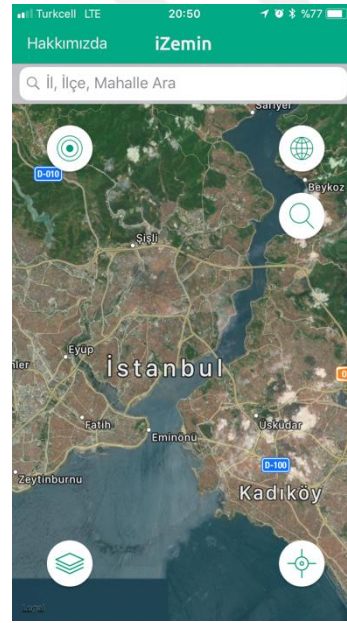
Sondaj noktalarının listelendiği ekranda UITableView ve UITableViewCell bileşenleri kullanılmıştır. Bu bileşenler sayesinde sondaj noktalarına dair bilgiler dinamik olarak görüntülenmiştir. iOS platformunda liste ve listedeki satırlara karşılık gelen tablo ve hücreler vardır. Bu bağlamda, tasarımımıza uygun olarak Custom hücreler oluşturulmuştur.

5.3.2 Harita Modülü

Bölüm 4'te mobil platformlarda kullanılan harita kütüphanelerinden kısaca bahsedilmiştir. İZemin mobil uygulamasında, Apple firmasının geliştiricilere ücretsiz olarak sağladığı 2 boyutlu harita kütüphanesi (MapKit) kullanılmıştır. Bu kütüphane, hem iOS platformunda çalışan ve aynı zamanda Apple'ın geliştiricileri tarafından geliştirildiği için en kararlı kütüphane olmasından hem de kullanımı ile ilgili olarak herhangi bir sınırlama bulunmamasından dolayı seçilmiştir. MapKit içerisinde bulunan uydu görüntüsü ve ana ekrana eklenen altlık harita, değiştirme butonu sayesinde kullanıcıya opsiyonel olarak sunulmuştur (Şekil 5.10).



(a)



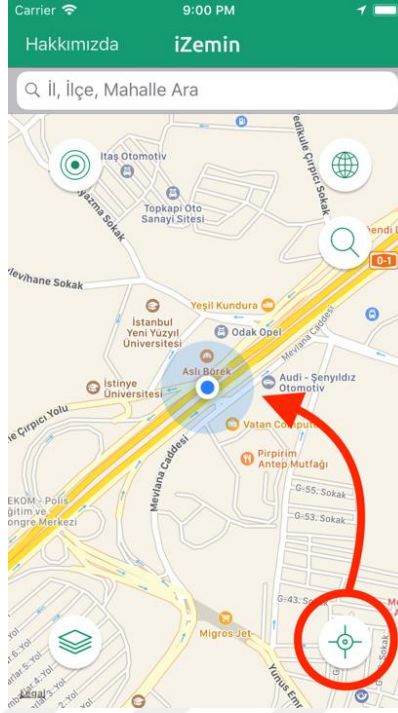
(b)

Şekil 5. 10 Apple MapKit Kütüphanesi 2B harita (a), uydu görüntüsü (b)

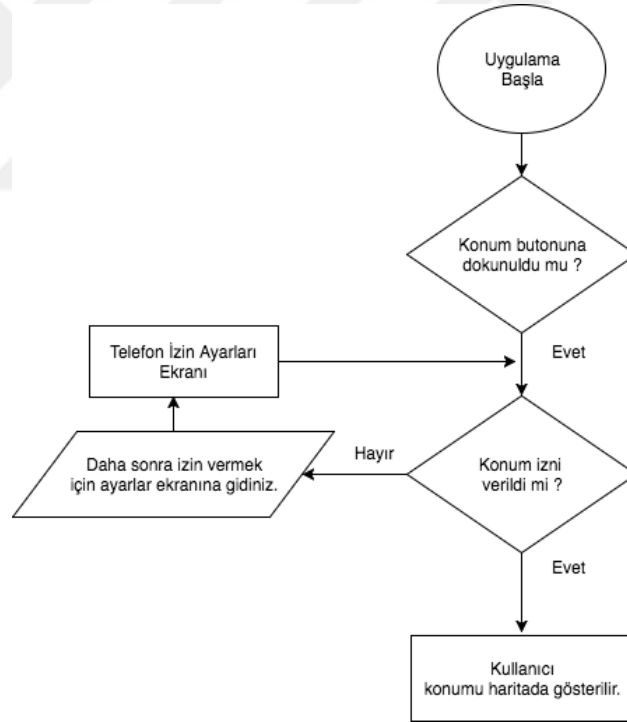
MapKit kütüphanesinin geometrik altlık haritası, Web Mercator projeksiyon sisteminde coğrafi koordinatlar ile çalışmaktadır. Bu nedenle iZemin mobil uygulamasında kullanılacak veriler bu koordinat sistemde çalışacak şekilde coğrafi koordinatlara dönüştürülmüştür.

Mobil cihaz üzerinde bulunan GNSS alıcısının xCode'da projeye eklenen Corelocation kütüphanesinin kullanılmasıyla, kullanıcı "konuma git" butonuna dokunduğunda Şekil 5.11'de görüldüğü gibi mobil cihazın konumu harita üzerinde gerçek zamanlı olarak mavi bir ikonla gösterilmektedir. Ancak bu işlemin yapılabilmesi için kullanıcıdan bir sefere mahsus olmak üzere konumunun alınabilmesi için uygulamaya izin vermesi istenmektedir. Bu izin, kullanıcının konuma git butonuna ilk defa dokunduğunda istenmektedir ve izin vermesi veya vermemesi durumunda, bu butona tekrardan dokunduğunda artık izin istenmeyecektir (Şekil 5.12). İzin vermemesi durumunda bu buton işlevsiz hale gelecektir. Daha sonra bu işlevden yararlanabilmesi için kullanıcının cihaz ayarlarından kendisi elle izin vermesi gerekmektedir. Eğer kullanıcı konumunun kullanılmasına izin vermişse ve mobil cihazın konumu ekran sınırları dışında kalmış ise, diğer mobil harita uygulamalarında olduğu gibi ana ekran üzerinde bulunan konuma git butonu sayesinde cihazın konumu ekranın merkezinde olacak şekilde harita yeniden konumlandırılır.

Harita ekranına eklenen UISearchBar bileşeniyle MapKit kütüphanesinde yer alan geocoding özelliği kullanılarak; il, ilçe, mahalle ve cadde gibi adres bilgileri üzerinden arama yapılabilir. Bu arama işleminde kullanıcı tarafından yazılan kelime Apple geocoding servisine gönderilerek bir sonuç dönmesi beklenmektedir. Dönen sonucun koordinat bilgilerinden yararlanarak harita üzerinde o noktaya hedeflenmesi sağlanmaktadır. Bu sayede harita üzerinde konumu bilinmeyen bir noktaya kolay bir şekilde ulaşılmaktadır.



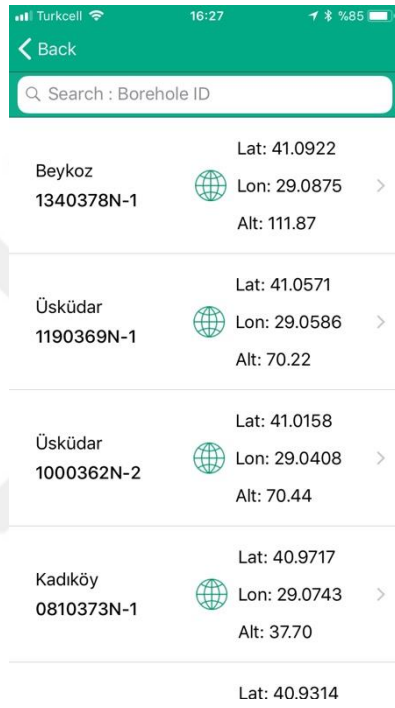
Şekil 5. 11 Konuma git butonu ve cihaz konumunu



Şekil 5. 12 Konum izni akış diagramı

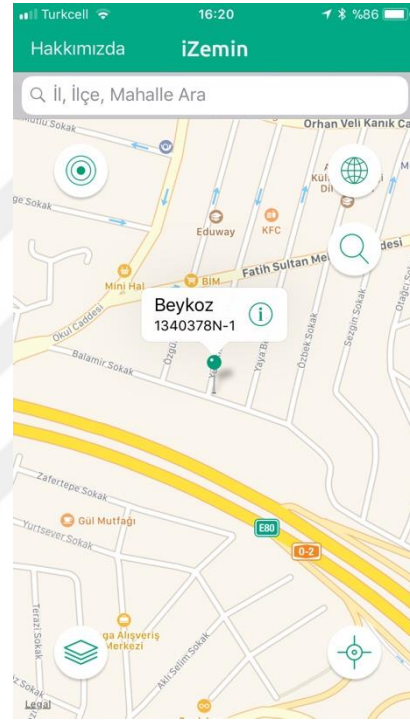
iZemin uygulamasında sondaj arama işlemi üç şekilde gerçekleştirilmektedir. Birincisi; id, adres veya bölgesi bilinen sondaj noktalarının sondaj listesinde aratılarak bulunmasıdır. Bu arama işlemi uygulama çalıştırıldığında tüm sondaj verisi yüklenerek cihaz üzerinde geçici olarak saklanmasıyla sağlanmıştır. Bu şekilde bulunan bir sondaj noktası Şekil

5.13’de görüldüğü gibi yeşil bir pin ile kullanıcıya gösterilmektedir. Bu pin’e dokunulduğunda üzerinde açılan ek bilgi penceresi ile sondajın bölge ve id bilgisi kullanıcıya sunulmaktadır. Bu işlev MapKit kütüphanesinde bulunan Annotation sınıfı ile sağlanmaktadır. Eğer kullanıcı bu noktanın formasyon bilgilerini görmek isterse bu pencereye dokunarak formasyon ekranına geçiş yapabilmektedir. Böylece formasyon ekranından litoloji ve adres gibi sondaja dair daha fazla bilgi edinebilmektedir (Şekil 5.14).



Search : Borehole ID		Lat	Lon	Alt
Beykoz	1340378N-1	41.0922	29.0875	111.87
Üsküdar	1190369N-1	41.0571	29.0586	70.22
Üsküdar	1000362N-2	41.0158	29.0408	70.44
Kadıköy	0810373N-1	40.9717	29.0743	37.70
		Lat: 40.9314		

(a)



(b)

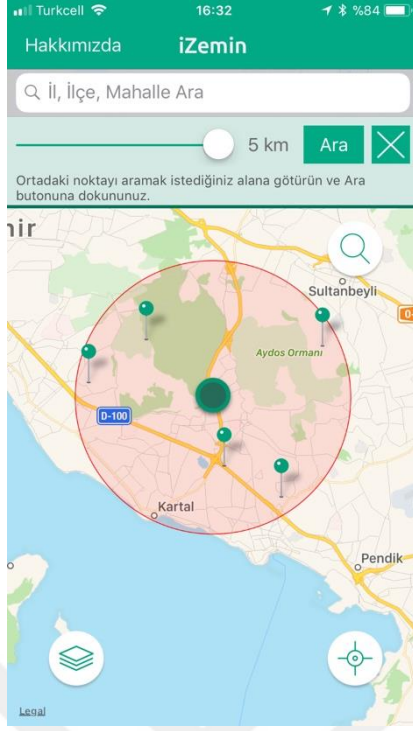
Şekil 5. 13 Sondaj noktalarının liste gösterimi (a), Sondaj noktasının harita üzerindeki gösterimi (b)



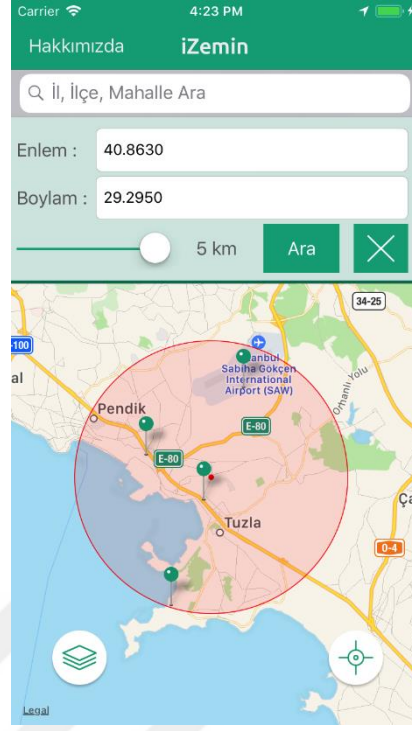
Şekil 5. 14 Formasyon ve litoloji bilgilerinin gösterilmesi

İkincisinde, harita üzerinde herhangi bir noktaya uygulanan tampon bölge analizi ile hedeflenen bölgedeki sondaj noktaları yeşil pin'lerle gösterilmektedir. Haritada arama butonuna dokunulduğunda ayar menüsü açılmakta ve harita ekranının ortasında yeşil dairesel bir imleç ortaya çıkmaktadır. Bu imleç analiz yapılacak noktaya getirilmekte ve tampon bölge mesafesi seçilmektedir. Arama işlemi tamamlandıktan sonra harita üzerinde gösterilen pin'ler üzerinden bilgi penceresine ve formasyon ekranına ulaşılabilir (Şekil 5.15a).

Üçüncüsünde, aranan sondajların id veya adres bilgileri yoksa ve bu sondajların haritada hangi bölgede yer aldıkları bilinmiyorsa bununla birlikte koordinatları biliniyorsa, koordinat bilgileri kullanılarak tampon bölge analizi yapılabilir (Şekil 5.15b).



(a)



(b)

Şekil 5. 15 Herhangi bir noktaya belli bir mesafedeki sondaj noktalarının aranması (a), Koordinatlarıyla bilinen bir noktaya belli bir mesafedeki sondaj noktalarının aranması (b)

Tampon bölge analizinde kullanıcıya 1 ila 5 km yarıçap aralığında tampon bölgeler çizdirilebilmektedir. MapKit kütüphanesi içerisinde bulunan mesafe metodu ve MKCircle sınıfı kullanılarak hazırlanan fonksiyonla belirtilen mesafe içerisinde kalan sondaj noktaları tespit edilebilmekte ve harita üzerinde daire içerisinde gösterilebilmektedir. Ancak, tampon bölgenin, kullanıcı tarafından girilen mesafe elipsoidal mesafemi mi yoksa düzlem mesafe mi kabul edilerek oluşturulduğuna dair MapKit kütüphanesi dokümanlarında herhangi bir bilgi yer almamaktadır. Oysaki bu fark kullanıcı açısından önemlidir. Kullanıcı harita/düzlem değil gerçek/elipsoidal mesafe girerek tampon bölge oluşturmak ister. Bu konunun araştırılması ve netleştirilmesi için ArcGIS programından da yararlanarak şöyle bir uygulama yapılmıştır:

1. ArcGIS ortamında altlık harita (base map) olarak Web Mercator (auxiliary sphere) (EPSG: 3857) projeksiyonunda topoğrafik harita (World Topographic Map) yerleştirilmiştir. Özgün datum ve projeksiyonu ITRF1996 Transverse Mercator (DOM: 30) olan sondaj noktaları WGS84 datumunda ve Web Mercator (auxiliary

sphere) (EPSG: 3857) projeksiyonunda yeniden konumlandırılmıştır. Tüm sondaj noktalarının WGS84 coğrafi koordinatları hesaplatılarak öznitelik tablosuna yazdırılmıştır. Buffer aracı ile 0940391N-1 numaralı sondaj noktası merkez olacak şekilde düzlem (planar) mesafe 5 km girilerek bir tampon bölge oluşturulmuştur (Şekil 5.16). Tampon bölge mesafesi olarak düzlem/harita mesafesi girildiği için 5 km yarıçaplı bir daire meydana gelmiştir.

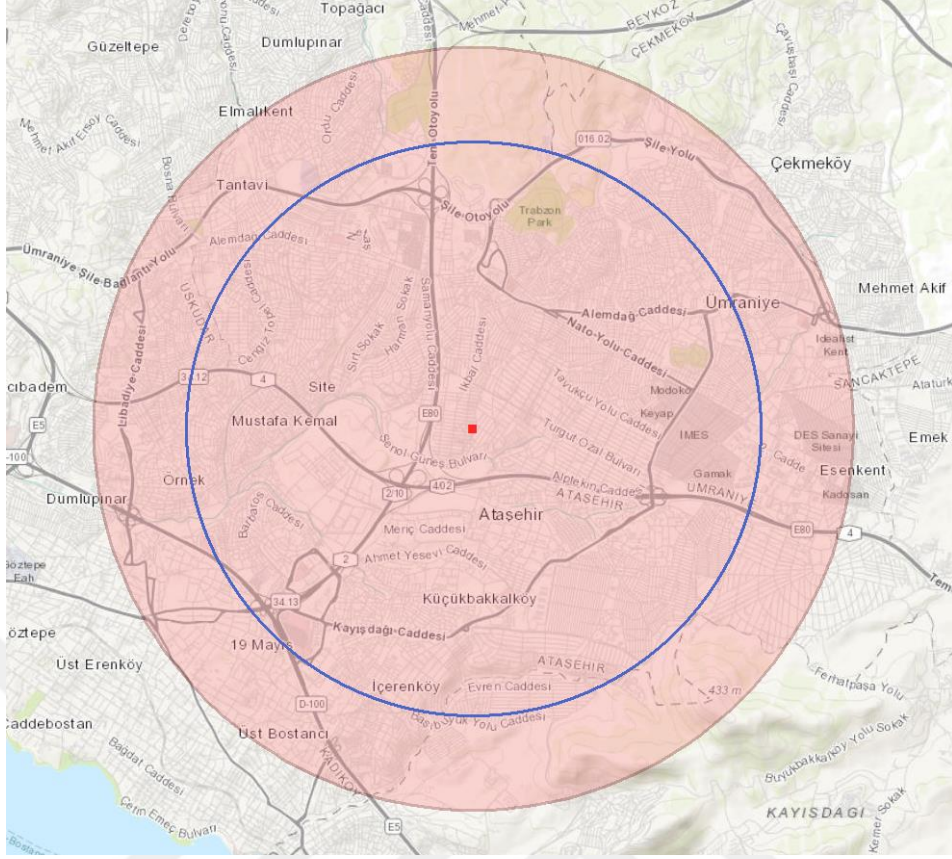
2. ArcGIS Buffer aracı ile aynı sondaj noktası (0940391N-1) merkez olacak şekilde küresel (geodetic) mesafe 5 km girilerek bir tampon bölge oluşturulmuştur (Şekil 5.16). Elde edilen tampon bölge şekil ve büyüklük bakımından birinci aşamada elde edilen tampon bölgeden farklı olmuştur. Çünkü;

- a. ArcGIS Web Mercator (auxiliary sphere) (EPSG: 3857) projeksiyonunda referans yüzey küre olmasına karşın küresel coğrafi koordinatlar değil, elipsoidal coğrafi koordinatlar (WGS84) kullanılmaktadır. Bu farklılıktan dolayı Web Mercator (auxiliary sphere) projeksiyonu tam olarak konform bir projeksiyon değildir. Bu nedenle, bu aşamada oluşturulan tampon bölge tam olarak bir daire olmamıştır.

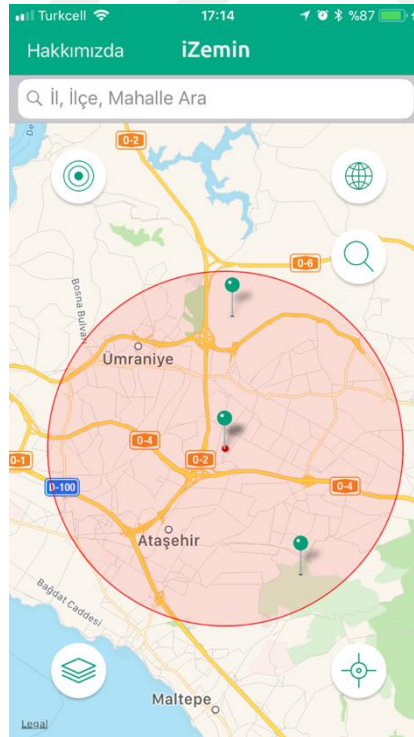
- b. ArcGIS Web Mercator (auxiliary sphere) (EPSG: 3857) projeksiyonunda uzunluklar deforme olmaktadır. Üstelik, tam olarak konform bir projeksiyon olmadığı için uzunluklar farklı doğrultuda farklı miktarlarda deforme olmaktadır. Başka bir deyişle, normal konform projeksiyonlarda bir noktada her doğrultuda uzunluk deformasyon oranları aynı olmasına karşın, Web Mercator (auxiliary sphere) (EPSG: 3857) projeksiyonu tam olarak bir konform projeksiyon olmadığı için bir noktada her doğrultuda uzunluk deformasyon oranları aynı değildir. Bu nedenle küresel 5 km'lik tampon bölge yarıçapı haritaya doğu ve batı doğrultularında 6.6144695 km, kuzey doğrultusunda 6.6434150 km ve güney doğrultusunda 6.6389301 km'lik bir uzunluk olarak izdüşmüştür.

3. iZemin Sondaj Ara (Tampon Bölge) aracı ile 0940391N-1 numaralı sondaj noktası merkez olacak şekilde 5 km mesafeli bir tampon bölge oluşturulmuştur (Şekil 5.17). Elde edilen tampon bölgeyi birinci ve ikinci aşamada elde edilen tampon

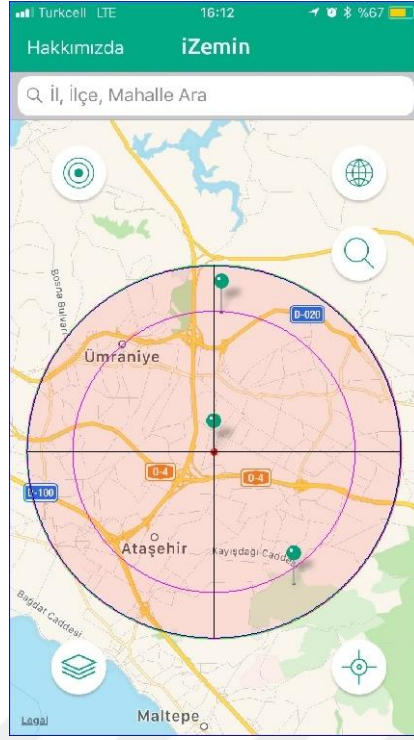
bölgeler ile karşılaştırmak için iZemin ekran görüntüsü alınmış ve birinci ve ikinci aşamada elde edilen tampon bölgeler ile birlikte AutoCAD ortamına taşınmıştır. Birinci aşamada elde edilen tampon bölge AutoCAD ortamına bir çember (circle) nesnesi olarak gelirken, ikinci aşamada elde edilen tampon bölge bir çember değil, bir çoklu çizgi (polyline) nesnesi olarak gelmiştir. iZemin ekran görüntüsündeki tampon bölgenin merkezi diğer iki tampon bölgenin merkezinde olacak şekilde ekran görüntüsünün ölçeği ayarlanmıştır. Ayrıca, iZemin ekran görüntüsündeki tampon bölge üzerine bir çember (circle) yerleştirilmiş ve bu çemberin tam olarak tampon bölge sınırlarında yer aldığı tespit edilmiştir (Şekil 5.18). Bununla birlikte, bu çemberin ikinci aşamada elde edilen tampon bölgenin sınırını temsil eden çoklu doğrudan farklı olduğu görülmüştür. Bu çember doğu-batı doğrultusunda çoklu doğru ile çalışırken, kuzey-güney doğrultusunda çakışmamaktadır. Çemberin yarıçapı kuzey doğrultusunda çoklu doğrudan 28.9455 m, güney doğrultusunda ise 24.4606 m daha kısadır (Şekil 5.19). Bu analiz sonuçlarına göre Apple MapKit kütüphanesinin MKCircle sınıfının, kullanıcı tarafından girilen tampon bölge mesafe değerini düzlem/harita mesafesi olarak kabul ettiği, tampon bölge sınırını bir çember olarak meydana getirdiği anlaşılmıştır. Sonuç olarak, iZemin Sondaj Ara (Tampon Bölge) aracı ile meydana getirilen tampon bölge, kullanıcının girdiği yarıçap değerinden yaklaşık % 32 daha büyük yarıçaplı bir bölge olmaktadır. Eğer bunun tersi bir durum ortaya çıksaydı, kullanıcı açısından önemli olurdu ve kullanıcıyı bu konuda bilgilendirmek gerekirdi. Başka bir deyişle, tespit edilen durumun kullanıcıyı olumsuz yönde etkilemeyeceği ve bu nedenle kullanıcıyı bilgilendirmenin gerekli olmadığı düşünülmektedir.



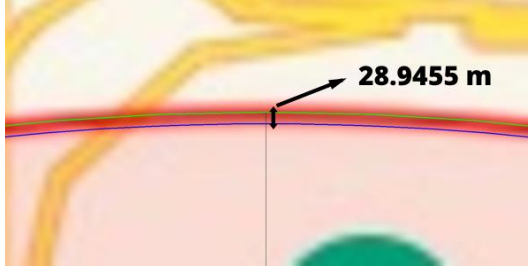
Şekil 5. 16 ArcGIS buffer aracı ile tampon bölgeler (mavi: düzlem, kırmızı: elipsoidal)



Şekil 5. 17 iZemin uygulamasından 5 km'lik örnek tampon bölge



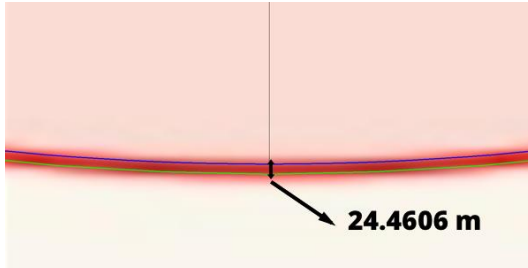
Şekil 5. 18 ArcGIS Buffer aracının düzlem mesafe seçeneği ile (pembe), ArcGIS Buffer aracının elipsoidal mesafe seçeneği ile (yeşil) ve iZemin Sondaj Ara (Tampon Bölge) aracı ile (kırmızı/mavi) elde edilen tampon bölgeler.



(a)



(b)



(c)



(d)

Şekil 5. 19 a) Kuzey, b) Doğu, c) Güney ve d) Batı sınırlarında tampon bölgeler arasındaki farklılıklar.

SONUÇ VE ÖNERİLER

Gerçekleştirilen çalışmada, İBB sondaj veri içeriği ve uygulama amacı dikkate alınarak özgün bir sondaj veri yapısı tasarlanmıştır. Bu veri yapısına göre, bulut tabanlı bir veri tabanı hazırlanmıştır ([Bölüm 5.3.1.2](#)). Böylece verileri cihaz üzerinde tutmak yerine tek bir merkezde tutmakla verilerin standart ve güncel olması sağlanmıştır. Klasik tablolar yerine JSON nesneleri meydana getirilmiş ve böylece sondajlar nesne dizisi şeklinde veri tabanında tutulmuştur ([Bölüm 5.3.1.3](#)). Bir nesne yönelimli programlama dili kullanılarak MVC ve Singleton yazılım geliştirme mimarisine göre oluşturulmuş bu yapının mobil platformlar için daha uygun olduğu görülmüştür ([Bölüm 5.1.2](#)).

Ayrıca, tezin amacı kapsamında belirlenen hedefler bağlamında;

- iZemin için arayüz tasarımları yapılmıştır. iZemin arayüzü hem Apple tasarım ilkeleri ile uyumlu hem de mobil platformuna uygun şekilde hazırlanmıştır. Temel bileşenlerin boyutları (butonlar, uyarı ekranları vb.) ve yazıların okunaklılıklarına dikkat edilerek kullanıcı dostu (User-Friendly) olması sağlanmıştır ([Bölüm 5.2.2](#)).
- Geriye dönük üç temel iOS versiyonu (alt limit olarak iOS 9.3 versiyonu) desteklenmiştir ([Bölüm 5.1.1](#)).
- Uygulama geliştirilmesinde xCode yazılımının ve Swift programlama dilinin en güncel versiyonları kullanılmıştır ([Bölüm 5.1.1](#)).
- Corelocation ve MapKit kütüphaneleri kullanılarak, cihaz konumunu harita üzerinde gösterilebilmiştir. Böylece kullanıcı sondaj noktalarına göre konumunu karşılaştırabilmektedir ([Bölüm 5.3.2](#)).

- Kullanıcı, hem sözel arama hem de koordinat veya harita üzerinden belirli bir tampon mesafesi içerisinde arama yapabilmektedir. Sondaj noktalarının konum bilgileri ile birlikte formasyon ve litoloji bilgilerine de kolayca erişebilmektedir ([Bölüm 5.3.2](#)).
- Çalışmada tamamen ücretsiz veya kısmen ücretli servisler kullanılmıştır ([Bölüm 5.1.2](#)).
- Uygulamaya ilave olarak eklenen kütüphaneler yardımıyla hem analitik olarak kullanımını takip etmek hem de bildirim servislerinden yararlanmak mümkün olmuştur. Kullanıcıların uygulama ekranlarını kullanım oranları ve hangi arama şeklinin daha çok kullanıldığı tespit edilebilecektir ([Bölüm 5.1.2](#)).

Gelecekte, iZemin hem Android platformlarına hem de tablet cihazlara uyumlu hale getirilebilir. Çalışma içerisinde kullanılan veriler zenginleştirilebilir ve öznetelikler kullanım amacına uygun olarak revize edilebilir. Sahada sondaj çalışması yapan ekip tarafından yeni veya eksik sondajların mobil cihaz üzerinden merkezi veri tabanına eklenebilmesi gibi işlevler kazandırılabilir.

KAYNAKLAR

- [1] Zervas, D., Nichols, G. J., Hall, R., Smyth, H. R., Lüthje, C., ve Murtagh, F., (2009). "SedLog: A shareware program for drawing graphic logs and log data manipulation", Computers & Geosciences, 35(10):2151-2159.
- [2] Wolniewicz, P., (2014). "SedMob: a mobile application for creating sedimentary logs in the field", Computers & Geosciences, 66:211-218.
- [3] Chang, Y. S., ve Park, H. D., (2004). "Development of a web-based geographic information system for the management of borehole and geological data" Computers & Geosciences, 30(8):887-897.
- [4] Lee, S., Suh, J., ve Park, H. D., (2015). "BoreholeAR: A mobile tablet application for effective borehole database visualization using an augmented reality technology" Computers & Geosciences, 76:41-49.
- [5] Selçuk, M. ve Gökgöz, T., (2001), "Geological and Geotechnical Information System for Istanbul", Fourth Turkish-German Joint Geodetic Days, 3-6 April 2001, Berlin, Germany, vol.2, 853-858.
- [6] Gökgöz T., Yutaka K., Selçuk M. (2002). "Geological Database of Istanbul", Third International Symposium on Remote Sensing of Urban Areas, Istanbul, 1: 444-451.
- [7] The Geological Survey of Denmark and Greenland, GEUS Map, <http://data.geus.dk/geusmap/version2> , 10 Eylül 2017.
- [8] German Geological Survey Organizations, Borehole Map Germany, <https://geoviewer.bgr.de/mapapps/resources/apps/boreholemap/>, 10 Eylül 2017.
- [9] Baker, B. M., (2016). Beesis: An iOS Application For The Collection Of Citizen Science Data On Bee Activity In The United States, Master's Thesis, Mills College Computer Science, USA.
- [10] Wahl, M. A., (2013). Mapping Native Plants: A Mobile GIS Application For Sharing Indigenous Knowledge In Southern California, Master's Thesis, Faculty Of The USC Graduate School University Of Southern California Geographic Information Science And Technology, USA.

- [11] Omote, Y., Takata, Y., (2014). "The iOS/Android application "e-SoilMap" for creating a new user-oriented soil map", Soil Science and Plant Nutrition, 60:470-474.
- [12] Keçeli, D. A., (2009). Uygulamalı Jeofizik, Ekim Ajans Matbaacılık, Ankara.
- [13] Akdeniz, E., Mutlu, S., Yücel, G., Özdemir, V., (2012). "Zemin Etüt Raporlarının Esaslara Uygunluğunun Değerlendirilmesi: Eskişehir Örneği", Journal of Construction Technologies, 8(2):26-37.
- [14] Ayyıldız, M., (2006). Afyonkarahisar Zemin Etüt Bilgi Sistemi, Yüksek Lisans Tezi, Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü, Afyon.
- [15] Göktürk, A., (1983). Sondaj Tekniği, İTÜ Maden Fakültesi Yayınları, İstanbul
- [16] Ocakoğlu, F., (2013). Stratigrafi İlkeleri Ders Notları, Eskişehir Osmangazi Üniversitesi, Eskişehir
- [17] Özgül, N., (2011). İstanbul İl Alanının Jeolojisi, İstanbul Büyükşehir Belediyesi, Deprem ve Zemin İnceleme Müdürlüğü, İstanbul
- [18] INSPIRE D2.8.II.4_v3.0, (2013). Data Specification on Geology, INSPIRE, Europe
- [19] OGC/CGI GeoSciML SWG 4.1, (2017). Context Diagrams: Borehole, Open Geo Consortium
- [20] T.C. Çevre ve Şehircilik Bakanlığı, Türkiye Jeoloji Veri Standartlarının Belirlenmesi ve Jeoloji Temasının Hazırlanması, <http://www.csb.gov.tr/db/jvsb/webmenu/webmenu12150.pdf> , 10 Eylül 2017.
- [21] Ertuğrul, S., (2015), Akıllı Telefon Tarihi, <http://mediatrend.mediamarkt.com.tr/akilli-telefon-tarihi> , 25 Ağustos 2017
- [22] IDC, (2017), Worldwide Smartphone Company Market Share, <https://www.idc.com/promo/smartphone-market-share/vendor>, 25 Ağustos 2017
- [23] IDC, (2017), Worldwide Smartphone OS Market Share, <https://www.idc.com/promo/smartphone-market-share/os>, 25 Ağustos 2017
- [24] Statista, (2017), Number of Available Apps in the Apple App Store from July 2008 to January 2017, <https://www.statista.com/statistics/263795/number-of-available-apps-in-the-apple-app-store>, 25 Ağustos 2017
- [25] Apple, (2017), iOS Versions Distribution Chart, <https://developer.apple.com/support/app-store>, 25 Ağustos 2017
- [26] Apple, (2017), Apple Developer Documentation, <https://developer.apple.com/documentation>, 25 Ağustos 2017
- [27] Google, (2017), Google Maps iOS SDK Documentation, <https://developers.google.com/maps/documentation/ios-sdk/start>, 25 Ağustos 2017
- [28] MapBox, (2017), MapBox iOS SDK Documentation, <https://www.mapbox.com/ios-sdk/>, 25 Ağustos 2017

- [29] ESRI, (2017), ArcGIS Runtime SDK for iOS, <https://developers.arcgis.com/ios/latest/swift/guide/introduction.htm>, 25 Ağustos 2017
- [30] Apple, (2017), Apple Human Interface Guidelines, <https://developer.apple.com/ios/human-interface-guidelines/overview/themes>, 25 Eylül 2017



SONDAJ SINIFI SWIFT KODLARI

```
//  
// Borehole.swift  
// iZemin  
//  
// Created by Furkan Ozoglu on 30.04.2017.  
// Copyright © 2017 Furkan Ozoglu. All rights reserved.  
//  
  
import UIKit  
import Foundation  
  
class Borehole : NSObject{  
  
    var region : String = ""  
    var address : String = ""  
    var boreholeId : String = ""  
    var coordinate : Coordinate?  
    var formations : [Formation]?  
    var litoloji : String = ""  
  
    override init() {  
        coordinate = Coordinate()  
        formations = [Formation()]  
    }  
  
    struct Coordinate {  
        var latitude : Double = 0.0  
        var longitude : Double = 0.0  
        var altitude : Double = 0.0  
    }  
  
    struct Formation {  
        var name : String = ""  
        var altKot : Int = 0  
        var ustKot : Int = 0  
        var code : String = ""  
    }  
}
```

BULUT VERİ TABANINDAN UYGULAMAYA SONDAJ VERİLERİNİ YÜKLEYEN SINGLETON METOTU

```
//
// DBProvider.swift
// FirebaseDatabase
//
// Created by CBS MOBIL on 23.03.2017.
// Copyright © 2017 CBS MOBİ L. All rights reserved.
//

import Foundation
import FirebaseDatabase
import FirebaseStorage

protocol FetchData: class {
    func dataReceived(boreholes: [Borehole])
}

class DBProvider {
    private static let _instance = DBProvider()
    weak var delegate: FetchData?
    private init() {}
    static var Instance: DBProvider {
        return _instance
    }
    var dbRef: DatabaseReference {
        return Database.database().reference()
    }
    var boreholesRef: DatabaseReference {
        return dbRef.child(Constants.SONDAJLAR)
    }
    func getBoreholes(){
        boreholesRef.observeSingleEvent(of: DataEventType.value) {
            (snapshot: DataSnapshot) in
            var boreholes = [Borehole]()
            if let myBoreholes = snapshot.value as? NSArray {
                for i in 0 ..< myBoreholes.count{
                    if let boreholeData = myBoreholes[i] as? NSDictionary {
                        let newBorehole = Borehole()
                        if let boreholeId = boreholeData[Constants.SONDAJID] as? String {
                            newBorehole.boreholeId = boreholeId
                        }
                        if let region = boreholeData[Constants.BOLGE] as? String {
                            newBorehole.region = region
                        }
                    }
                }
            }
        }
    }
}
```

```

        if let address = boreholeData[Constants.ADRES] as? String {
            newBorehole.address = address
        }
        if let coordinate = boreholeData[Constants.KOORDINAT] as? NSDictionary {
            for (key, value) in coordinate {
                if key as! String == Constants.ENLEM {
                    newBorehole.coordinate?.latitude = value as! Double
                }
                else if key as! String == Constants.BOYLAM {
                    newBorehole.coordinate?.longitude = value as! Double
                }
                else if key as! String == Constants.YUKSEKLIK {
                    newBorehole.coordinate?.altitude = value as! Double
                }
            }
        }
    }
    if let formations = boreholeData[Constants.FORMASYONLAR] as? NSArray {
        newBorehole.formations?.removeAll()
        print(formations)
        for j in 0 ..< formations.count {
            if let formation = formations[j] as? NSDictionary {
                var myFormation = Borehole.Formation()
                for (key, value) in formation {
                    if key as! String == Constants.ISIM {
                        myFormation.name = value as! String
                    }
                    else if key as! String == Constants.ALTKOT {
                        myFormation.altKot = value as! Int
                    }
                    else if key as! String == Constants.USTKOT {
                        myFormation.ustKot = value as! Int
                    }
                    else if key as! String == Constants.KODU {
                        myFormation.code = value as! String
                    }
                }
                newBorehole.formations?.append(myFormation)
            }
        }
    }
    if let litoloji = boreholeData[Constants.LITOLOJİ ] as? String {
        newBorehole.litoloji = litoloji
    }
    boreholes.append(newBorehole)
}
}
self.delegate?.dataReceived(boreholes: boreholes)
}
}
}
}
}

```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Furkan ÖZOĞLU
Doğum Tarihi ve Yeri : 09.06.1990 / Nevşehir
Yabancı Dili : İngilizce
E-posta : f.ozoglu@gmail.com

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Y. Lisans	Harita Müh.	Yıldız Teknik Üniversitesi	-
Lisans	Harita Müh.	Yıldız Teknik Üniversitesi	2013
Lise		Barbaros Lisesi	2007

İŞ TECRÜBESİ

Yıl	Firma/Kurum	Görevi
2014	İstanbul Büyükşehir Belediyesi	Mobil uygulama geliştiricisi

YAYINLARI

Makale

1. Özoğlu Y., Kaysi F., Özoğlu F. (2014), "Mobil Cihazlar İçin Epub Standardında Müfredat Ders İçerik Üretimi", Eğitim ve Öğretim Araştırmaları Dergisi - Journal of Research in Education and Teaching, 3(19): 196-206.

Bildiri

1. Özoğlu F., Yalçın S., Oğuz H.K., Kılıç F. (2013), "Yeşil Binalar ve Lokasyonlarının CBS ile Değerlendirilmesi: İstanbul Örneği", Coğrafi Bilgi Sistemleri Kongresi, Ankara, Türkiye.
2. Özoğlu Y., Özoğlu F., Kaysi F. (2013), "Epub 3.0 Üretim Kriterlerinin Belirlenmesi - Determination of Epub 3.0 Production Criteria", International Conference on Quality in Higher Education (ICQH), Sakarya, Türkiye.
3. Özoğlu Y., Kaysi F., Özoğlu F. (2013), "Öğrenim Sürecinde Epub Kullanımı İle Öğretmenlere Sunduğu İmkanların Değerlendirilmesi", International Perspectives on New Aspects of Learning in Teacher Education- Building Bridges Conference, Türkiye
4. Özoğlu Y., Kaysi F., Özoğlu F. (2013), "Müfredat Derslerinin Mobil Teknolojilerle Kullanimini Sağlamak Üzere Epub Standardında İçerik Üretiminin Gerçekleştirilmesi", 2nd World Conference on Educational and Instructional Studies - WCEIS, Türkiye.

ÖDÜLLERİ

1. Teknik Gençlik Oturumu En İyi Sunum, TMMOB 4. CBS Kongresi, 11-13 Kasım, 2013, Ankara, Türkiye.