



BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

MOBİL DESTEKLİ MEZUN BİLGİ SİSTEMİ
GERÇEKLEŞTİRİLMESİ

Musa TURKAN
Yüksek Lisans Tezi

Tez Danışmanı
Doç. Dr. Metin KESLER
İkinci Tez Danışmanı
Yrd. Doç. Dr. Uğur YÜZGEÇ

BİLECİK, 2014
Ref. No: 10047245



BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

MOBİL DESTEKLİ MEZUN BİLGİ SİSTEMİ
GERÇEKLEŞTİRİLMESİ

Musa TURKAN
Yüksek Lisans Tezi

Tez Danışmanı
Doç. Dr. Metin KESLER
İkinci Tez Danışmanı
Yrd. Doç. Dr. Uğur YÜZGEÇ

BİLECİK, 2014



BILECIK SEYH EDEBALI UNIVERSITY
Graduate School of Science
Department of Computer Engineering

**MOBILE ASSISTED ALUMNI INFORMATION SYSTEM
IMPLEMENTATION**

Musa TURKAN
Master's Thesis Degree

Thesis Advisor
Assoc. Prof. Dr. Metin KESLER
Second Thesis Advisor
Assoc. Prof. Dr. Uğur YÜZGEÇ

BİLECİK, 2014



**BİLECİK ŞEYH EDEBALI
ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS
JÜRİ ONAY FORMU**

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun tarih ve sayılı kararıyla oluşturulan jüri tarafından tarihinde tez savunma sınavı yapılan Musa TURKAN' ın "MOBİL DESTEKLİ MEZUN BİLGİ SİSTEMİ GERÇEKLEŞTİRİLMESİ" başlıklı tez çalışması Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak oy birliği/~~oy çokluğu~~ ile kabul edilmiştir.

JÜRİ

ÜYE (TEZ DANIŞMANI): Doç. Dr. Metin KESLER

ÜYE: Yrd. Doç. Dr. Sıtkı ÖZTÜRK

ÜYE: Yrd. Doç. Dr. Suhap ŞAHİN

ONAY

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 05.10.2014. tarih ve38...../.....4..... sayılı kararı.

**İMZA MÜHÜR
Doç. Dr. Mehmet KURBAN**



ÖZET

Bu çalışmada sosyal medya uyumu sağlanan bir mezun bilgi sistemi geliştirme aşamaları incelenmiş, MVC framework kullanılarak bir mezun sistemi gerçekleştirilmiştir.

Uygulamada Microsoft SQL Server 2012 veri tabanı yönetim sistemi kullanılmış, C# dili kullanılarak MVC mimarisi tercih edilmiştir. Uygulama gerçekleştirilirken jquery java script kütüphanesinden yararlanılmıştır. Bilecik Şeyh Edebali Üniversitesi öğrenci bilgi sistemiyle bütünleşik çalışan, sosyal medya hesaplarıyla kayıt olarak kullanılabilen bir uygulama ortaya koyulmuştur. Facebook sistemi, kullanımı yaygın bir sosyal medya sistemi olduğu için bu sistemle uyumlu çalışma yapılmıştır.¹ Yazılım projesinin başarıya ulaşması ve güncelliğinin sağlanabilmesi için sürüm takip sistemiyle çalışma devam ettirilmektedir.

Mezun bilgi sistemine destek uygulaması olarak mobil bir uygulama gerçekleştirilmiştir. Gerçekleştirilen mobil uygulama Google bulut mesaj servisini kullanmakta ve kullanıcıya konum bildirme, arkadaşlarının konum bilgilerini görme imkanı sağlamaktadır.

Mezunlarla ilgili yapılan etkinliklerin mezunlara duyurulması, mezunlara anket sorularının sorulabilmesi, toplu olarak eposta gönderilebilmesi işlemleri geliştirilen sistem sayesinde mümkün olabilir. Sisteme çeşitli firmalardan gelen iş ilanları da girilebilir ve mezunların uzmanlık alanlarına göre iş ilanlarına ulaşılabilir. Sistemde her bir mezunun uzmanlık alanları ve iş tecrübeleri bilgisi de tutulabilmektedir.

Anahtar sözcükler: MVC framework, , mezun takip, facebook entegrasyon, Mezun sistemi, Google bulut mesaj servisi, android konum servisi

¹<http://tr-tr.facebook.com/notes/promoqube/g%C3%BCncel-facebook-t%C3%BCrkiye-istatistikleri/230108337030128>

ABSTRACT

In this study, a graduate information provided social media compliance system development are examined, a graduated system was carried out using the MVC framework.

In practice, we used Microsoft SQL Server 2012 database management system and using C # language was implemented to MVC architecture. We used Repository Pattern while performing applications and used java script library as known Query. Ultimately, we have preferred a application that integrated with Bilecik Seyh Edebali University Student Information System and may be logined with social media accounts. Facebook system, which is widely used for a social media system compatible with this system has been conducted. The success of software projects and work with version tracking system to ensure timeliness is maintained.

Alumni support the application as a mobile information system has been implemented. Held mobile application to use the GCM(Google Cloud Message) service and notify the user location, provide the opportunity to see the location information of friends.

Graduates Graduates of the activities related to the announcement, the graduates of the survey questions can be asked, bulk email can be sent through the system developed procedures may be possible. Online job postings can also be entered from various companies and according to the expertise of graduates to jobs accessible. Online each graduate's field of expertise and work experience information may also be held.

Keywords: MVC Framework, alumni system, facebook integration, Google cloud message, android location service

TEŞEKKÜR

Bu çalışmanın yürütülmesi sırasında desteğini esirgemeyen danışmanım Doç. Dr. Metin KESLER'e, danışmanlığıyla yol gösteren Yrd. Doç. Dr. Uğur YÜZGEÇ'e yardımlarıyla bana destek olan Doç. Dr. Cihan KARAKUZU'ya Bilecik Şeyh Edebali Üniversitesi Bilgi İşlem Daire Başkanlığı çalışma arkadaşlarıma, özellikle Murat FİDAN'a, Murat ÖZALP'e, Yusuf MUŞTU'ya, Vedat MARTTİN'e, Arif GÜNEL'e çalışmalarım sırasında hep yanımda olan Sevgili Kardeşim Fatma TURKAN'a teşekkür ederim.

Musa TURKAN

Mayıs, 2014

İÇİNDEKİLER

Sayfa No

TEZ ONAY SAYFASI

ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER	iv
ÇİZELGELER DİZİNİ	vi
ŞEKİLLER DİZİNİ	vii
SİMGELER VE KISALTMALAR DİZİNİ	xii
1. GİRİŞ	1
1.1 Giriş	1
1.2 Literatür Çalışması.....	4
1.3 Yöntem.....	5
2. MEZUN SİSTEMLERİ	7
2.1 Mezun Sistemi İhtiyacı	7
2.2 Sosyal Medya Kullanımı	8
2.3 Mezun Takibi için Kullanılan Yöntemler	11
2.4 Mezun Sistemlerinin Özellikleri.....	14
3. GELİŞTİRİLEN MEZUN BİLGİ SİSTEMİ	16
3.1 Geliştirmede Kullanılan Araçlar Ve Mimari Desen.....	16
3.1.1 Facebook Uyumu	16
3.1.2 Visual Studio 2012	25
3.1.3 Redmine.....	25
3.1.4 MVC (Model View Controller).....	26
3.1.5 Entity Framework.....	27
3.1.6 Mssql Server 2012.....	31
3.2 Mezun Sistemi Mimarisi.....	32
3.3 Mezun Sisteminde Güvenlik.....	36
3.4 Mezun Sistemi Veri tabanı.....	37

3.4.1	Sisteme Eklenen İçerik Bilgilerini Tutulduğu Tablolar	37
3.4.2	Mezun Sistemi İş İlanı Tabloları	38
3.4.3	Mezun Sisteminde Mezunlara Ait Veriler	38
3.5	Mezun Bilgi Sistemi Web Uygulaması	41
3.6	Sistem Yöneticisi İşlemleri	42
3.7	Sisteme Kayıt Olan Kullanıcılara Ait İşlemler	47
4.	MEZUN BİLGİ SİSTEMİ MOBİL UYGULAMA	57
4.1	Android Geliştirme Ortamının Hazırlanması	57
4.1.1	Java Uygulama Geliştirme Seti Kurulumu (Jdk)	58
4.1.2	Eclipse Kurulumu	58
4.1.3	Android Uygulama Geliştirme Seti (Android Sdk) Yükleme	59
4.2	Google Bulut Mesaj Servisi	62
4.2.1	Google Bulut Mesaj Servisindeki Bileşenler	65
4.2.2	Bulut Mesaj Servisinde İşlem Adımları	67
4.3	Konum Bilgisinin Elde Edilmesi	69
4.4	Android Uygulamasıyla Web Servisinden Veri Okuma	72
4.5	Geliştirilen Android Uygulaması	73
4.5.1	Android Uygulaması Ana Ekranı	74
4.5.2	Oturum Açma Ekranı	76
4.5.3	Detay Ekranlar	78
4.5.4	Konum Bildirme	82
4.5.5	Bildirimlerin Görüntülenmesi	85
5.	SONUÇ	88
6.	KAYNAKÇA	89
7.	ÖZGEÇMİŞ	93

ÇİZELGELER DİZİNİ**Sayfa No**

Çizelge 2.1. Ülkelere göre facebook kullanıcı sayıları ve büyüme oranı (Gülbahar Y. vd,2010)	10
Çizelge 2.2. Üniversitelerin kullandıkları mezun izleme yöntemleri (Akadal E. , 2013)	12

ŞEKİLLER DİZİNİ

	Sayfa No
Şekil 2.1. Türkiye’deki Facebook kullanıcılarının yaş aralığı.	10
Şekil 3.1. Facebook uygulama oluşturma sayfası.	17
Şekil 3.2. Facebook uygulama bilgileri tanımlama.	18
Şekil 3.3. Facebook sisteminde tanımlanmış uygulama.	19
Şekil 3.4 Uygulamaya ait logo bilgilerinin tanımlanması.	20
Şekil 3.5. Facebook uygulaması ayarları.	21
Şekil 3.6. Facebook sisteminin sağladığı verilerden bir kısmı.	22
Şekil 3.7. Facebook uygulamasında alan adının güncellenmesi.....	23
Şekil 3.8. Visual studio facebook sdk yükleme.	23
Şekil 3.9. Visual studio 2012 ile proje özelliklerinden portu sabitleme.	24
Şekil 3.10. Asp.Net MVC Modeli.....	26
Şekil 3.11. Asp.Net MVC mimarisinin çalışması.	27
Şekil 3.12. Entity framework yapısı. (Lerman J. , 2010).....	28
Şekil 3.13. Entity framework yaklaşımları.	29
Şekil 3.14. Hangi yaklaşım ne zaman kullanılabilir.	30
Şekil 3.15. Visual Studio kullanılarak veri tabanından model oluşturma.....	31
Şekil 3.16. Sistemin genel mimari yapısı.	32
Şekil 3.17. Projenin genel mimari yapısı.	32

Şekil 3.18. Mezun bilgi sistemi proje bağımlılıkları.	33
Şekil 3.19. Controller bağımlılıkları.	34
Şekil 3.20. Data model katmanı sınıf görünümü.	34
Şekil 3.21. Mezun sistemi web uygulamasına ait sınıflar.	35
Şekil 3.22. Controller sınıflarının bağımlılıkları.	36
Şekil 3.23. Nitelik tanımlama.	36
Şekil 3.24. İçerik tabloları.	38
Şekil 3.25. İş ilanı ilişkileri.	38
Şekil 3.26. Temel kullanıcı verileri ilişkileri.	39
Şekil 3.27. Firma verileri arasındaki ilişkiler.	40
Şekil 3.28. Anket sorularının tanımlanması ve cevaplanması veri ilişkileri.	40
Şekil 3.29 Kullanıcı verilerine ait entity model görünümü.	41
Şekil 3.30. Duyuru yönetimi ekranı.	43
Şekil 3.31. Etkinlik Yönetimi ekranı.	44
Şekil 3.32. Oturum açma ekranı.	45
Şekil 3.33. Facebook kullanıcı kayıt ekranı.	46
Şekil 3.34. Anket yönetimi ana ekranı.	47
Şekil 3.35. Anket soruları yönetimi ekranı.	47
Şekil 3.36. Anket soruları için cevap yönetimi ekranı.	47
Şekil 3.37 Mezun oturum açma algoritması.	49

Şekil 3.38. Facebook bilgilerinin kullanıcıyla eşleştirilmesi	50
Şekil 3.39. Anket sorusu cevaplama.	51
Şekil 3.40. Kullanıcı uzmanlık alanı ekleme.	51
Şekil 3.41. İleti ekleme ekranı.	52
Şekil 3.42. Mezun profil sayfasındaki gizlilik ayarları.....	53
Şekil 3.43. Mezun arama ekranı.	53
Şekil 3.44. Diploma sorgulama.....	54
Şekil 3.45. Ana sayfada etkinliklerin görünümü.....	54
Şekil 3.46. Kayıt olan öğrenci sayılarının gösteren grafik.....	55
Şekil 3.47. Haritada mezun sayıları.	56
Şekil 4.1. JDK indirme sayfasının görüntüsü.	58
Şekil 4.2. Eclipse indirme sayfası.	59
Şekil 4.3. SDK Manager yükleme ekranı.	60
Şekil 4.4 Eclipse yardım menüsünden yeni eklenti yükleme.	61
Şekil 4.5. Android Development Tools eklentisi yükleme.....	61
Şekil 4.6. Android Development Tools ayarları.....	62
Şekil 4.7. Google bulut mesaj servisinin çalışması (Derinözlü, 2013).....	63
Şekil 4.8. Google API ayarlarından Android cihazlar için mesaj servisinin açılması. ...	64
Şekil 4.9 Google API proje numarası.	64
Şekil 4.10. Genymotion eklentisinin eclipse aracına kurulması.	68

Şekil 4.11. Genymotion emülatör simgesinin eclipse menüsünde görünümü.	69
Şekil 4.12. Eclipse ile Genymotion emülatörünün başlatılması.	69
Şekil 4.13. AnroidManifest dosyasında konum servisi izinlerinin açılması.....	71
Şekil 4.14. Android uygulamasının web servisi ile haberleşmesi.....	73
Şekil 4.15. Android uygulamasının haberleştiği servisler.....	74
Şekil 4.16. Android uygulaması ana ekranı.	75
Şekil 4.17. GCM kayıt işlemlerini yönetmek amacıyla hazırlanan sınıf.	76
Şekil 4.18. Oturum açma ekranı.	77
Şekil 4.19. Shared preferences dosyası oluşturma.	77
Şekil 4.20. Verileri listelemek amacıyla oluşturulan özel adapter sınıfının görünümü. .	78
Şekil 4.21. İçeriklerin sekmelerle görünümü.	79
Şekil 4.22. Duyuru verilerinin tarayıcı ile görüntülenmesi.....	80
Şekil 4.23. Eclipse aracında android uygulamasında kullanılan etkinlikler (activity). ..	80
Şekil 4.24. Android uygulamasında etkinliklerin görünümü.	81
Şekil 4.25. Eclipse aracında asenkron olarak tanımlanmış sınıfın görünümü.	82
Şekil 4.26. Konum servisinin dinlenmesini sağlayan sınıf.....	82
Şekil 4.27. Konum bilgisinin elde edilmesi.	83
Şekil 4.28. Genymotion emülatöründe gps butonu.....	84
Şekil 4.29. Genymotion emülatörü gps ayarları.	84
Şekil 4.30. Android uygulamasında konum bilgisinin kaydedilmesi.	85

Şekil 4.31. Sistem bildirimlerinin görüntülenmesi.	86
Şekil 4.32. GCM servisinden gelen mesajların bildirimini sağlayan metot.	86
Şekil 4.33. Mezun web uygulamasından GCM servisine mesaj gönderilmesi.....	87

SİMGELER VE KISALTMALAR DİZİNİ

API	Application programming interface
GCM	Google Cloud Message (Google Bulut Mesaj Servisi)
IDE	Integrated Development Environment (Bütünleşik Geliştirme Ortamı)
JDK	Java Development Kit
MS SQL	Microsoft SQL Server
MVC	Model View Controller
SSL	Secure Socket Layer (Güvenli Giriş Katmanı Protokolü)
SQL	Structural Query Language (Yapısal Sorgu Dili)
SDK	Software development kit
SVN	Apache Subversion
PHP	Hypertext Preprocessor
WWW	Dünya Geniş Ağı

1. GİRİŞ

1.1 Giriş

Yazılım teknolojisindeki gelişmeler insan hayatını oldukça fazla etkilemektedir. Gelişen teknolojiyi kullanım kolaylığı insanların günlük hayatlarında yazılım kullanım oranını oldukça fazla yükseltmektedir. Mobil aygıtların yaygınlaşmasıyla birçok yazılım uygulaması insanların kullanımına sunulmuştur. Yazılım dünyasındaki bu baş döndürücü gelişmeler sosyal medya kavramının da ortaya çıkmasını sağlamıştır. Sosyal medya ile entegre çalışan yazılımlar günlük hayatta daha fazla tercih edilmekte ve yoğun olarak kullanılmaktadır.

Sosyal medya kullanımının artmış olmasıyla birçok kurum geliştirdiği uygulamaları çeşitli sosyal medya sistemlerine bütünleşik çalıştırmaktadır. Kullanıcı adı ve parola gerektiren uygulamaların sayısı artınca kişiler birçok parola bilgisini hatırlamak zorunda kalmışlardır. Uygulamaların merkezi sistemlere entegre edilmesi sağlanarak tek bir sisteme ait parolaların başka uygulamalar tarafından da kullanılabilmesine imkan sağlanmıştır. Bu sayede uygulamalara erişim kolaylaşmış, uygulamaların kullanılma oranları artmıştır. Kullanıcılar sosyal medyada zaten sürekli olarak kullandıkları parola bilgileri ile uygulamalara kolayca giriş yapabilmişlerdir. Sosyal medya sistemleriyle bütünleşik çalışan uygulamalarda üretilen veriler sosyal medya ile hızlı bir şekilde yayılmakta ve kısa zamanda çok fazla sayıda insana ulaşabilmektedir. Bu sayede kurumlar insanlara ulaştırmak istedikleri bilgileri hızlı ve kolay bir yoldan kişilere ulaştırabilmektedir. Sürekli canlı bir ortam olarak kullanılan sosyal medya ortamlarında kullanıcılara kolay bir şekilde ulaşılabilir.

Tüm bu kullanım kolaylıkları üniversitelerin de sosyal medya sistemlerinde hesap oluşturmalarını ve öğrencilerine sosyal medya ortamları aracılığıyla ulaşabilmesini sağlamıştır. Üniversitelerden mezun olan öğrencilerin, mezun sistemlerini kullanma oranı düşük olduğundan mezunlara ait güncel verilere ulaşmakta zorluklar yaşanmaktadır. Mezun olan öğrencilere ulaşmanın en kolay yollarından biri de sürekli canlı bir ortam olan sosyal medyayı kullanmaktır. Bu sayede her öğrencinin en

güncel verisine ulaşılabilmekte ve öğrencilere bilgilendirmeler hızlı bir şekilde ulaştırılabilmektedir.

Üniversiteler, mezunlarını takip ederek üniversiteye ne gibi katkılar sağlayabileceklerini araştırabilir, mezunlara sorulan anket soruları ile eğitim alan öğrencilerin piyasada ne gibi sorunlarla karşılaştıklarını öğrenebilir, yapılan etkinlikleri mezunlara duyurarak öğrencilerin birbirlerinin tecrübelerinden faydalanmalarını sağlayabilirler. Tüm bu faydaların sağlanabilmesi için de mezunların en güncel verilerini analiz etmek, mezunlara anlık olarak ulaşabilmek önem kazanmaktadır.

Üniversiteler mezunlarına ulaşabilmek için çeşitli yöntemler kullanmaktadırlar. Bu yöntemlerin bir tanesi de mezunlara yönelik uygulamalar geliştirilmesi ve bu uygulamaların mezunlar tarafından kullanılmasının sağlanmasıdır. Geliştirilen uygulamaların geliştirme süreçleri ne kadar doğru planlanırsa sistemlerin kullanılabilirliği, yeniliklere uyum sağlaması da o oranda iyi olmaktadır.

Günümüzde geliştirilen yazılımların geliştirme süreçleri, sürüm takipleri, ve güncelleme sıklıkları önemli bir konu olarak karşımıza çıkmaktadır. Geliştirilen bir yazılımın başarıya ulaşması için standartlara uygun bir geliştirme sürecinin uygulanması önem arz etmektedir. Yeterince güncellenmeyen, tek bir yazılım personeline bağlı olan, sürümleri takip edilmeyen bir yazılım ilk kullanımda ne kadar iyi olursa olsun zamanla kullanılamaz hale gelecektir. Standartlara uygun şekilde geliştirilmeyen ve güncellenmeyen yazılımlar 5 yıl içinde kullanılamaz hale gelmektedir. Yazılımların doğru planlanması, gelişim ve yaşam süreçlerinin sağlıklı şekilde ortaya koyulması, bu işlemler için gerekli araçların kullanılması kurumsal uygulamaların ömrünü arttırmakta ve uygulama için harcanan iş gücünü de azaltmaktadır.

Bu çalışmada yazılım geliştirilirken Tourtoise SVN sürüm takip sistemi kullanılarak uygulamanın geliştirme adımları takip edilebilmiştir. Sistem tasarlanırken MS SQL Server 2012 veri tabanı yönetim sistemi kullanılmıştır. Çeşitli üniversitelerin mezun takip yöntemleri incelenmiş, mezun sistemlerinin genel ihtiyaçları ortaya çıkarılmıştır. Mezun bilgi sistemlerinin genel özelliklerinin yanında kurumun ihtiyaçları da göz önünde bulundurularak ilk sürüm ihtiyaçları tanımlanmıştır. Yapılan analize

uygun olarak mezun sisteminin ihtiyalarını karřılayabilecek bir veri tabanı tasarımı yapılmıřtır. Uygulama geliřtirilirken entity framework kullanılarak veri tabanı iřlemleri gerekleřtirilmiřtir. Repository pattern uygulanarak veri tabanı iřlemlerinin tek merkezden yapılması saėlanmıř bu sayede kod kalabalıėından sistemin kurtulması hedeflenmiřtir. Projeye sonradan dahil olan bir yazılım personelinin daha kısa srede sisteme adapte olabilmesi adına hazırlanan sınıf ve metotlara iliřkin aıklama blokları oluřturulmuřtur. Kullanıcıların facebook hesaplarıyla sisteme giriř yapabilmesi iin facebook sistemi zerinden bir uygulama tanımlanmıřtır. Tanımlanan bu uygulama sayesinde kullanıcıların doėrulanması saėlanmıřtır. İlk ařamada mezunların sosyal medya hesabı ile doėrulanabilmesi saėlanmıřtır. Mezunların konum bilgilerinin harita zerinde gsterilebilmesi amacıyla Google Maps entegrasyonu yapılmıřtır.

MVC framework ile geliřtirilen uygulamalar arama motorları ile daha kolay ve anlaşılır olarak indekslenebilmektedir. Sayfaların aspx gibi uzantılar kullanılmadan ulařılabilir olması ok daha kolay anlaşılır adresler oluřmasını saėlamaktadır. (Aktař V. vd., 2013). MVC mimarisi web api mantıėı ile alıřtıėından uygulamanın iinde tanımlanan fonksiyonlar bařka uygulamalara da veri saėlayabilecek yapıda planlanabilmektedir. Bu ve daha birok geliřmiř zelliėi sebebiyle MVC web uygulaması mimarisi tercih edilmiřtir.

Birinci blmde mezun bilgi sistemlerinde bulunması gereken zellikler incelenmiř, eřitli niversitelerdeki sistemler arařtırılmıřtır. Geliřtirilecek mezun sisteminin ihtiyalarının neler olduėu arařtırılmıř ve ortaya koyulmuřtur.

İkinci blmde alıřmada kullanılan yazılım geliřtirme yntemleri ve araları incelenmiřtir. Kullanılan araların temel zellikleri, saėladıkları faydalar belirtilmiřtir.

nc blmde alıřmada yapılan analizler ortaya koyulmuřtur. Sistemin bir analizi yapılmıř ve zellikleri aıklanmıřtır. Sistemdeki ihtiyaların nasıl olması gerektiėi belirtilmiřtir. Yazılım geliřtirilirken gereksinimlerin ve sınırların tanımlanarak ortaya koyulması bařarılı bir sre ynetimi iin en bařta gelen adımdır. Gereksinimler mmkn olduėunca geliřtirme srecinde gncellemeye ihtiya olmayacak řekilde tanımlanırsa verimli ve bařarılı yazılım projeleri ortaya ıkabilir. (Atbař H., 2012) Veri

tabanı tasarımı, yazılımın mimari yapısı, kullanılan sınıflar ve modeller, yazılım geliştirme sırasında kullanılan yöntemler açıklanmıştır. Sistemin bütünleşik çalıştığı uygulamalar tanıtılmıştır.

Dördüncü bölümde geliştirilen mezun android uygulaması açıklanmıştır. Bu bölümde uygulamada kullanılan temel özellikler tanıtılmıştır.

Son bölümde geliştirilen sistemin daha farklı ne tür çalışmalara konu olabileceği konusunda fikirler belirtilmiş sistemin iyileşmesi amacıyla yapılabilecek çalışmalar hakkında görüşler belirtilmiştir.

1.2 Literatür Çalışması

Literatür çalışması kapsamında YÖK sistemindeki tezler taranmış ve benzer tezler incelenmiştir. Konuyla ilgili yazılan makale ve yayınlar araştırılmıştır.

Güçlüoğlu (2013) yaptığı bir çalışmada mezun önerme sistemi geliştirmiştir. Web tabanlı bu sistem ile mezunların kimlik bilgileri sisteme tanımlanmaktadır. Tanımlanan kullanıcı bilgilerine göre firmalara uygun mezun önermesi sağlanması amaçlanmıştır. Veri tabanı bağlantılarının yönetildiği bir katman ve uygulama sayfalarının bulunduğu ikinci bir katman yapısı ile sistem tasarlanmıştır. Sistem Visual Studio 2010 ortamında geliştirilmiş ve entity framework kullanılmıştır.

Akadal (2011) de yaptığı bir çalışmada mezun bilgi sistemi geliştirmiştir. Çalışmasında mezun sistemlerinin gerekliliği incelenmiş, mezun takip yöntemleri belirtilmiş ve web tabanlı bir uygulama ortaya koyulmuştur. Çalışma kapsamında çeşitli üniversitelerin mezun sayfalarının içerikleri incelenmiştir. Çalışmasında 10 yurt içi 12 yurt dışı üniversitenin mezun sistemlerini incelemiş ve bu sistemlerde bulunan özellikleri çıkarmıştır. Sistem tasarlanırken PHP dili ve MYSQL veri tabanı yönetim sistemi kullanılmıştır.

Erkal (2011) de Microsoft firmasının yazılım geliştirme araçlarını (Visual Studio 2010 ve MS SQL Server 2008 R2) kullanarak bir mezun sistemi gerçekleştirmiştir. Çalışmasında yazılım geliştirme araçlarının gelişimini incelemiş, yazılım geliştirme

aşamalarını belirtmiştir.

Muştu (2012) de yapmış olduğu çalışmasında android tabanlı bir uygulama geliştirmiş ve öğrencilerle öğretim üyelerinin etkileşimini bilişim araçlarıyla arttırmayı hedeflemiştir. Uygulamasında sms tabanlı bir mesajlaşma sistemi kullanmıştır. Uygulamanın farklı sistemlerle uyumunu sağlamak amacıyla web servisleri geliştirmiştir.

Köktürk sosyal ağ kullanım performansına göre üniversite değerlendirmesi hakkında bir tez hazırlamıştır. Çalışmasında Twitter sistemini temel alarak üniversitelerin twitter hesaplarını kullanımını incelemiştir. Üniversitelerin kurumsal kimliklerini dikkate alarak bir etki puanı oluşturmuş ve bu puanlamaya göre üniversiteleri sıralamıştır. Etki puanını oluşturmak için yeni ölçü ve metrikler kullanmıştır.

Karal ve Kokoç öğrencilerin sosyal medya sistemlerini kullanma amacına yönelik bir ölçek geliştirme çalışması yapmışlardır. Makalede sosyal medya kullanım amacını belirlemeye yarayan bir ölçek geliştirmişlerdir. Güvenirlilik ve geçerlilik analizleri sonucunda toplam on dört madden oluşan bir ölçek belirlemişlerdir.

1.3 Yöntem

Mezun Bilgi Sisteminin yazılım geliştirme sürecinde sürüm takip sistemi kullanılmıştır. Yapılan işler ve güncellemeler sürüm takip sistemi sayesinde kayıt altına alınmış ve yazılımın tüm süreci takip edilebilmiştir. Sürüm takip sistemi olarak Tourtoise SVN isimli ücretsiz sistem kullanılmıştır. Projeyle ilgili iş takibi SVN sistemiyle bütünleşmiş çalışan Redmine sistemiyle yapılmaktadır. Visual SVN eklentisi Visual Studio programına eklenmiştir. Bu eklenti sayesinde uygulamada yapılan geliştirmeler kolayca sisteme aktarılabilir. Sistemin geliştirme aşamaları aşağıdaki sıralama ile yapılmıştır.

- Sistem Analizi
- Veri tabanı tasarımı
- Yazılım Mimarinin Belirlenmesi

- Kodlama
- Test
- Devreye alma

Çalışma kapsamında mezunların bilgileri kullanılmakta olan öğrenci bilgi sisteminden doğrulanmıştır. Test çalışmalarında öğrenci bilgi sisteminin test veri tabanındaki veriler kullanılmıştır. Facebook sisteminden doğrulama testleri için test facebook hesapları oluşturulmuştur.

Sistem analizi kapsamında mevcut mezun bilgi sistemleri incelenmiş sistemlerin hangi amaçlara hizmet ettikleri araştırılmıştır. Mevcut amaçların yanında sosyal medya ve öğrenci bilgi sistemiyle uyumlu çalışacak bir sistemin analizi hazırlanmıştır. Sistemin geliştirilmesinde MVC deseni tercih edilmiş, C# programlama dili kullanılmıştır.

Mezun sistemine destek olmak üzere mobil bir uygulama geliştirilmiştir. Mobil uygulama android sistemlerde çalışan bir uygulama olarak geliştirilmiştir. Mobil uygulama ile mezunlara mesaj ve duyuru gönderme işlemleri Google bulut mesaj servisi ile sağlanabilir.

Sistem geliştirilirken tüm paydaşların tüm ihtiyaçlarının gerçekleştirilmesinden ziyade temel özelliklere sahip ilk prototip mezun sisteminin ortaya çıkarılması amaçlanmıştır. Bu sebeple paydaşlardan gelebilecek yeni ihtiyaçlar ve özellikler sonraki sürümlerde değerlendirilmek üzere kayıt altına alınmıştır. Sınırsız istek ve kısıtlı zaman olduğundan bu çalışma boyunca istekler ilk sürüm gereklilikleri ile sınırlandırılmıştır.

Mezun sistemi ihtiyaçları ve analizi ortaya koyulurken birçok üniversitenin sistemi incelenmiştir. İncelenen sistemlerin birçoğuna kullanıcı girişi yapılamadığından sadece ana özelliklerinin nelere olduğu incelenmiştir. Kullanıcı girişi yapılabilen uygulamalar daha detaylı olarak incelenebilmiştir. Sakarya Üniversitesi, Karadeniz Teknik Üniversitesi, Gazi Üniversitesi sistemlerine kullanıcı kaydı yapılarak inceleme yapılabilmektedir.

2. MEZUN SİSTEMLERİ

2.1 Mezun Sistemi İhtiyacı

Üniversiteler birçok farklı amaçlarla mezunları ile iletişim kurma ihtiyacı hissetmektedir. Mezunların canlı olarak iletişim ve adres bilgilerine ulaşma, mezunlara duyurular gönderme, mezunlara yönelik etkinlikler düzenleme, mezunların piyasadaki başarısını ölçme, güncel sanayi ihtiyaçlarını takip edebilme gibi amaçlarla mezunları takip etme ihtiyacı ortaya çıkmaktadır. Bu amaçları gerçekleştirmek için de mezunlara ulaşmanın bir yolu olarak mezun sistemleri kullanılmaktadır.

Üniversitelerin mezun sistemleri çeşitli amaçlara hizmet etmektedir. Üniversite mezun sistemleri ve mezun derneklerinin amaçlarını inceleyen Akadal genel olarak aşağıdaki amaçlarla kuruldukları gözlemiştir (Akadal E., 2013).

- Mezunlar arasında iletişim sağlamak, sevgi saygı ve birliktelik oluşturmak
- Mezunlarla mevcut öğrenciler arasında iletişimi sağlamak
- Okuyan öğrencilere burs sağlamak, kariyer oluşum ve gelişim yönlendirmesi yapmak
- Üniversite etkinliğini ve saygınlığını arttıracak çalışmalar yapmak
- Girişimcilik konusunda yol göstermek ve eğitim faaliyetinde bulunmak
- Bilimsel ve kültürel etkinliklerde bulunmak

Yukarıda belirtilen amaçlar incelendiğinde mezun sistemlerinin mezunlar arası iletişimi sağlaması, mezunlara kolay ulaşmayı sağlaması ve mezunların etkinliklerden haberdar edilmesi gibi amaçlar için kullanıldığı gözlemlenebilir. Genel olarak mezun sistemlerindeki ortak amaçlardan birisi de, üyeler arasındaki sevgi, saygı ve iletişimi güçlendirici çalışmalar yapmaktır (Akadal E., 2013). Gerçekleştirilen Mezun Bilgi Sisteminin bu amaçlara uygun olarak içerik ve özellikler barındırması amaçlanmıştır. Bu amaçla ilk sürümde sosyal medya ile doğrulama entegrasyonu sisteme dahil edilmiş ve mesajlaşma özelliği sisteme eklenmiştir.

Birçok üniversite mezunlarını takip etmek için kendi sistemi yerine sosyal

medya hesaplarını veya mezun derneklerini tercih etmektedir. Özellikle yabancı üniversiteler LinkedIn hesapları ile mezunları takip etmektedirler. İş dünyasına yönelik bir ağ olan bu sistem sayesinde mezunlara ait iş bilgileri de takip edilebilmektedir.

Çalışma kapsamında incelenen 20 kadar üniversitenin mezun sistemlerinin yapısı değerlendirilmiştir. Genel olarak mezun sistemlerine sosyal medya uyumu yapılmadığı gözlenmiştir. Sosyal medya ile uyumlu uygulama yerine ayrı bir sosyal medya hesabı açılmıştır.

Üniversite sanayi işbirliğine yönelik olarak mezunların iş dünyası ile buluşturulmasını sağlayacak, mezun bilgilerinin standart formatlarda paylaşılmasını sağlayacak bir mezun bilgi sistemi üniversite mezunları arasında bilgi paylaşımını arttıracak, öğrencilerin ve üniversitelerin uygulamadaki yeterli ihtiyaçlarını anlamasını sağlayacaktır. Etkili bir mezun takip sisteminden elde edilen veriler sayesinde üniversitelerin eğitim programlarının günceli takip etmesi sağlanabilir.

2.2 Sosyal Medya Kullanımı

Ülkemizdeki genç nüfusun sosyal medya kullanım oranı oldukça yüksektir. Sosyal medya kullanım oranlarının yayınlandığı birçok sosyal medya analiz sisteminden ve yapılan çeşitli çalışmalardan bunu görmek mümkündür. Bu kadar yoğun kullanılan sosyal medya ile bütünleşik çalışan bir sistemin kullanılma oranı da yüksek olacaktır. Bu başlık altında sosyal medya kullanımıyla ilgili çalışmalara yer verilmiştir.

Gençlik ve Spor Bakanlığının yaptığı sosyal medya araştırmasında gençlerin sosyal medya kullanım alışkanlıkları, sosyal medyaya bakışı incelenmiştir. Bu çalışmada, gençlerin sosyal medyayı önemli bir toplumsal platform olarak gördüğü gözlenmiştir. Gençlerin %56'sı sosyal medyanın kitleleri harekete geçirebileceğini düşünmektedir. Her iki gençten birinin mobil cihazı ile sosyal medya hesabını kullandığı bu çalışmadan elde edilen sonuçlardan birisidir. Araştırmada gençlerin sosyal medyada paylaşılan içeriklere yoğun olarak yorumlar yaptıkları, arkadaşlarının

paylaştıkları içerikleri değerlendirdikleri görülmüştür.¹

Gençlerin sosyal medya kullanımını yüksek olması hazırlanan uygulamaların sosyal medya ile uyumlu çalışması ihtiyacını ortaya çıkarmıştır. Sosyal medya ile uyumlu çalışan uygulamaların içerikleri gençler arasında çok daha hızlı yayılacaktır. Mezun sistemlerinin önemli bir sorunun da kullanım oranının düşüklüğü olduğu düşünüldüğünde sosyal medya yoluyla bu soruna çözüm üretilebileceği görülebilir.

“Bugün dünya nüfusunun üçte ikisi, herhangi bir sosyal ağı ziyaret ediyor. Ziyaret ettikleri platformlarda harcadıkları zaman ise internette harcanan zamandan üç kat hızlı artıyor. Milyonlarca kişi sosyal medyada içerik üretiyor. 2012’de sadece Twitter’ın kullanıcı sayısının bir milyar kişiyi geçmesi bekleniyor.” (Bavlı U., 2011)

Sosyal ağlarla ilgili uygulamalar geliştirmek yeni bir endüstri oluşturuyor. Facebook ve diğer platformlardaki büyüme ekonomiye katkı sağlıyor.² Gülbahar, Kalelioğlu ve Madran yazdıkları bir makalede ülkelere göre facebook kullanıcı sayılarını Çizelge 2.1’de belirtmiştir.

Sosyal medya kullanımıyla ilgili sayısal istatistikler hazırlayan birçok sistem bulunmaktadır. Bu sistemlerden birisi de socialbakers.com isimli sistemdir. Bu sistemde ülkelere göre sosyal medya kullanım oranları görülebilmektedir. Bu sistem dışında birçok farklı sistem de sosyal medya kullanım alışkanlıkları konusunda araştırma yapmakta ve istatistikler yayınlamaktadır.

Mobil cihazların kullanımının artması sosyal medyanın da kullanımını arttırmıştır. Bu cihazlar sayesinde kullanıcılar her an sosyal medya sistemlerine ulaşabilmekte ve yoğun olarak kullanabilmektedirler.

¹ <http://www.gsb.gov.tr/HaberDetaylari/3/3816/genclik-ve-spor-bakanligi-turkiyenin-en-kapsamli-sosyal-medya-arastirmasini-yapti.aspx>

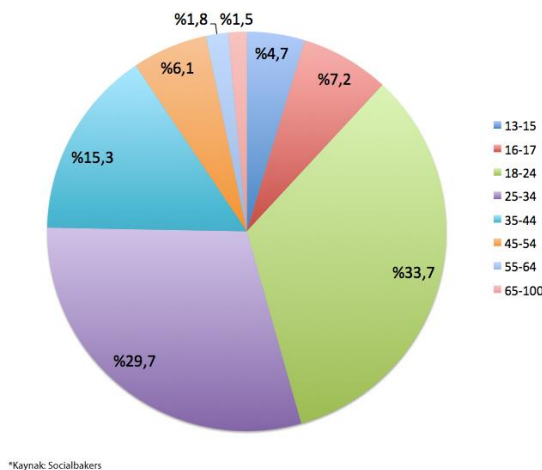
² <http://www.iletisimvediplomasi.com/mehmet-kartal-turkiyede-sosyal-medya-raporu/>

Çizelge 2.1. Ülkelere göre facebook kullanıcı sayıları ve büyüme oranı (Gülbahar Y. vd,2010)

Sıra	Ülke	Kullanıcı Sayısı 2009	Kullanıcı Sayısı 2010	12 aylık büyüme %	24 aylık büyüme %
1	Amerika	69,378,980	125,881,220	81.4%	352.6%
2	İngiltere	18,711,160	26,543,600	41.9%	137.6%
3	Endonezya	6,496,960	25,912,960	298.9%	12253.6%
4	Türkiye	12,382,320	22,552,540	82.1%	550.9%
5	Fransa	10,781,480	18,942,220	75.7%	669.7%
6	İtalya	10,218,400	16,647,260	62.9%	3289.8%
7	Kanada	11,961,020	15,497,900	29.6%	61.1%
.....				
14	Almanya	3,136,680	9,949,760	217.2%	1509.8%

Socialbakers.com adlı internet sitesinin yaptığı analizlere göre ülkemizdeki facebook kullanıcılarını yaş aralığı Şekil 2.1'deki gibidir. Grafik incelendiğinde üniversite çağındaki gençlerin sayısının yüksek bir yüzdede olduğu görülmektedir.

Türkiye'de bulunan Facebook kullanıcılarının yaş dağılımı



Şekil 2.1. Türkiye'deki Facebook kullanıcılarının yaş aralığı.

2.3 Mezun Takibi için Kullanılan Yöntemler

Mezunların takibi için aşağıdaki yöntemlerden birisi kullanılabilir. (Akadal E., 2013)

- Mezun Otomasyonu
- Web Sayfası
- Sosyal Medya Sayfası
- Mezun Dernekleri

Bu yöntemler genel olarak mezunları bilgilendirmek, yapılan etkinliklerden haberdar etmek amacıyla kullanılmaktadır. Mezun otomasyon sistemi olarak ifade edilen sistemlerde ise mezunlara ait güncel bilgilerin tutulmaya çalışıldığı, mezunlara ait bilgilerin sürekli güncellendiği, kariyer ve uzmanlık alanıyla ilgili bilgilerin tanımlandığı sistem olduğu görülmektedir. Mezun portal sistemlerinde mezunların mesajlaşabilmesi ve diğer arkadaşlarına ulaşabilmesi de mümkün olmaktadır. Sosyal medya yöntemindeyse mezunlar için bir facebook hesabı oluşturulmakta ve duyurular buradan bildirilmektedir. Mezun dernekleri yoluyla yürütülen çalışmalarda ise mezunlara yönelik etkinlikler düzenlenmesi, kariyer sahibi mezunların mevcut öğrencilerle buluşturulması, ekonomik katkı sağlayan bir takım alışveriş işlemleri yürütülebilmektedir.

Yurt dışındaki üniversitelerde mezunları izlemek için çeşitli yöntemler kullanılmaktadır. Akadal (2013) yaptığı çalışmada yerli yabancı birçok üniversiteyi incelemiş ve bu farklılıkları ortaya çıkarmaya çalışmıştır. Farklılıklar incelendiğinde sosyal medyadaki çeşitlilik arttıkça bu ortamların mezun izleme amacıyla kullanıldığı gözlenebilir. Yurt dışındaki üniversitelerin mezun takibi için daha çok linked In sayfası kullandıkları görülmektedir. Çizelge 2.2 bu çalışmayı göstermektedir.

Çizelge 2.2. Üniversitelerin kullandıkları mezun izleme yöntemleri (Akadal E. , 2013)

Üniversite Adı	Web Sayfası	Mezun Bilgi Sistemi (Otomasyon)	Facebook Sayfası	Twitter Sayfası	Friend Feed Sayfası	Youtube Sayfası	Flicker	Linked In	Portal	RSS
Istanbul Teknik Üniversitesi	✓	✓	✓	✓	✓	✓				
Boğaziçi Üniversitesi	✓		✓	✓						
Kocaeli Üniversitesi		✓								
Orta Doğu Teknik Üniversitesi	✓		✓	✓		✓				✓
Yıldız Teknik Üniversitesi		✓								
Williams College	✓		✓	✓		✓	✓	✓		
Harvard University	✓		✓	✓				✓	✓	
Indiana University	✓		✓	✓		✓	✓	✓	✓	

Ortadoğu Teknik Üniversitesinin ana sayfasında mezunlara yönelik bir bağlantı bulunmaktadır. Mezun bağlantısına tıklandığında mezunlara yönelik etkinliklerin duyurulduğu bir sayfa açılmaktadır. Sayfada ayrıca üniversitenin kariyer merkezine yönelik bağlantılar bulunmaktadır. Ayrıca mezun defteri isimli bir bağlantıda mezunların düşüncelerini dile getirebilecekleri bir ortam sunmaktadır. Mezun girişi yapılamadığı için sisteme giriş yapıldığında ne tür hizmetler olduğu incelenememiştir. Sistemde ayrıca bir etkinlik takvimi de bulunmaktadır.

Boğaziçi Üniversitenin ana sayfasında da mezunlar bağlantısı bulunmaktadır. Bağlantıya tıklandığında mezunlara yönelik haberlerin bulunduğu siteye yönlendirme yapılmaktadır. Sistemde özellikle başarı hikayeleri, eski mezunların başarı hikayeleri göze çarpmaktadır. Ayrıca mezunlara yönelik olarak yüksek lisans olanakları, teknoloji merkezlerinin bağlantıları da sayfada bulunmaktadır. Mezunların sisteme üye olması için ad, soyad, birim bölüm gibi bilgilerini elle girmeleri istenmektedir.

Karadeniz Teknik Üniversitesinin ana sayfasında mezun bağlantısı bulunmaktadır. Üye girişi sayfasında sosyal medya uyumu bulunmamaktadır. Mezuniyet bilgileri sisteme girilerek üye olunmaktadır. Bilgiler sisteme girildiğinde belirtilen eposta adresine üyelik bilgileri gönderilmektedir.

Hacettepe üniversitesinde çeşitli bölümlerin kendilerine ait mezun sistemleri bulunmaktadır. Sayfada etkinlik takvimi, istatistikler, duyurular bulunmaktadır.

İstanbul Teknik Üniversitesinin ana sayfasında mezunlar bağlantısı bulunmaktadır. Bağlantı mezunlara yönelik bilgilendirmelerin olduğu bir web sitesine yönlendirme yapmaktadır. Sayfada üniversiteye destek ol kısmında çeşitli hediyelik eşya satışı, duyurular, etkinliklerle ilgili bilgilendirmeler bulunmaktadır. Sisteme üye olmak için bir anlaşma metni gösterilmekte metni kabul ediyorum işaretlendikten sonra kullanıcı bilgileri sisteme girilerek üye olunmaktadır.

İstanbul Üniversitesi mezun takibi için mezun derneğine ait bir siteyi kullanmaktadır. Mezun derneğinin sitesinde haberler, etkinlikler duyurulmaktadır.

Yıldız Teknik Üniversitesinin mezun takibi için mezun koordinatörlüğü isimli bir birimi bulunmaktadır. Mezunlara duyurular, iş ilanları, üniversiteden haberler ve etkinlikler bu sayfada duyurulmaktadır. Mezunlara yönelik kuruluşların bir listesi de sayfada belirtilmektedir. Sayfada üniversiteden mezun olmuş ünlü kişilere ait bir kayan yazı da bulunmaktadır. Mezun sistemi kullanabilmek, detaylı mezun araması yapmak için üyelik gerekmektedir. Mezunlar sisteme bilgilerini girerek üye olmaktadır.

Çanakkale On Sekiz Mart Üniversitesinin mezun sistemi sayfasında duyurular ve iş ilanları göze çarpmaktadır. Sistemin ana sayfasında sistem dahil edilen son firma isimleri yayınlanmaktadır. Öğrenci numarası ve kimlik bilgileri girilerek sisteme üye olunmaktadır.

Namık Kemal Üniversitesinin mezun sisteminde de etkinliklerle ilgili duyurular bulunmaktadır. Mezunlara ait güncel verileri toplamak, bu veriler üzerinde çok çeşitli sorgulamalar ve analizler yapmak amacıyla oluşturulmuş bir sistemdir. Sisteme öğrenci numarası ve kimlik numarası bilgileriyle üye olunabilmektedir. Sistemin ana sayfasında mezunlar aranabilmekte, mezunların telefon ve eposta bilgilerine ulaşılabilir.

Zirve Üniversitesinin mezun sisteminde öğrenci bilgileriyle üye olunabilmektedir. Özgeçmiş ekleme, profil ayarları özellikleri bulunmaktadır.

Uşak Üniversitesinin mezun sisteminde mezunların isimlerinin listelendiği bir arama kısmı bulunmaktadır. Kimlik ve bölüm bilgileri ile sisteme üye olunarak kullanılmaktadır.

Sakarya Üniversitesinin mezun bilgi sisteminde mezunlar arası mesajlaşma, harita üzerinde mezunların nerede olduğu bilgisi, kariyer servisleri özellikleri bulunmaktadır. Sistemde gruplar oluşturulabilmekte, iş ilanı eklenebilmektedir. Sisteme üye olmak için kimlik bilgileri ve öğrenci bilgileri kullanılmaktadır. Üye olan kullanıcılar profil ayarlarını yapabilmekte, fotoğraf ve video ekleyebilmektedir.

Üniversitelerin mezun sistemleri genel olarak mezunlar arası iletişim, etkinlik ve duyuruların mezunlara bildirilmesi amacıyla yönelik olarak geliştirilmiş olduğu gözlenmektedir. İncelenen üniversitelerde herhangi bir sosyal medya uyumu gözlenmemiştir.

2.4 Mezun Sistemlerinin Özellikleri

Farklı üniversiteler incelendiğinde mezun sayfalarında genel olarak aşağıdaki özelliklerin olduğu gözlenmiştir.

- Duyurular: Mezunlarla ilgili duyurular ve etkinliklerin bildirildiği alanlar bulunmaktadır. Mezunlara yönelik etkinlikler, haberler bu başlık altında sunulmaktadır.
- İş İlanları: Firmalara ait iş ilanları mezun sistemi üzerinden mezunlara ulaştırılmaktadır. İş ilanları genel olarak açık şekilde sayfada görülebilmektedir.
- Mezun Arama: Mezunlar arasında arama yaparak mezunlara ait izin verilen bilgilere erişim sağlanabilmektedir.
- Başarı Hikayeleri: Mezunların başarı hikayeleri mezun sayfalarına yayınlanabilmektedir.
- Bağlantılar: mezunlarla ilgili bağlantı bilgileri mezun sayfalarında bulunmaktadır. Genel olarak mezun derneklerinin web adresleri, kariyer olanakları ile ilgili sayfa bağlantılarına yer verilmektedir.

- Diploma Sorgulama: Diploma sorgulama sayfasında gerekli bilgiler girilerek diplomanın gerçek olup olmadığı sorgulanabilmektedir.

Bazı üniversiteler ise mezun sistemlerinin yanında sosyal medya hesaplarıyla da mezunları ile iletişime geçmeye çalışmaktadır. Üniversiteler genel olarak ana sayfalarında mezun sistemleriyle ilgili bağlantılar bulundurmaktadır. Mezunlara sağlanan olanaklar da mezun sayfalarında duyurulmaktadır. Üniversitelerin sosyal tesislerinden yararlanma, mezunlara yönelik çıkartılan dergilerle ilgili bilgilendirmeler web sayfasından yapılmaktadır. Eski mezunların başarıları da bazı sistemlerde siteden duyurulmaktadır.

Yurt dışındaki üniversitelerden genel olarak kapsamlı bir mezun sistemine sahip olmadıkları görülmektedir. Üniversiteler genel olarak mezunlarla iletişime geçmek için sosyal medyayı kullanmaktadır (Akadal E., 2013).

Mezun bilgi sistemlerine üyelik işlemleri için üniversitelerde genel olarak eposta adresi, okul numarası, pasaportu numarası ve tc kimlik numarası bilgileri kullanılmaktadır. Bazı üniversiteler sisteme kayıt olmak için eposta adresinin yazılmasını istemekte kullanıcı bilgilerini bu eposta adresine göndermektedir. Bazı üniversitelerde ise varsayılan olarak pasaport numarası veya tc kimlik numarası parola olarak atanmaktadır. Mezunlara ait telefon numaralarının açık olarak yayınlandığı sistemler de bulunmaktadır.

Geliştirilen mezun bilgi sisteminde sosyal medya uyumu sağlanmıştır. Sistem üzerinden mezunların iletişim bilgilerine ulaşılabilir, çeşitli anketler tanımlanarak mezunlara yönelik anketler yapılabilir, iş ilanları ve firmaların bilgileri tanımlanabilir. Bu sayede mezunlara kariyer planlama konusunda veri sağlanması, sistemin kullanılma oranının artması sağlanmıştır.

3. GELİŞTİRİLEN MEZUN BİLGİ SİSTEMİ

Bu bölümde geliştirilen mezun sisteminin alt yapısı hakkında bilgi verilmekte, geliştirmede kullanılan araçlar tanıtılmakta ve tasarlanan sistem sayfaları hakkında bilgi verilmektedir.

3.1 Geliştirmede Kullanılan Araçlar Ve Mimari Desen

Mezun sistemi geliştirilirken Microsoft firmasına ait geliştirme araçları kullanılmıştır. Facebook hesaplarıyla uyum sağlanmıştır. Geliştirmede kullanılan araçlar aşağıda kısaca tanıtılmıştır.

3.1.1 Facebook uyumu

Facebook sistemiyle entegre uygulamalar geliştirilmesi için facebook tarafından sağlanan sdk kullanılır. Bunun yanında facebook sistemine kayıt olarak bir uygulama tanımlanmalıdır. Sisteme tanımlanan bu uygulama bilgileri kullanılarak doğrulama ve diğer uyum işlemleri gerçekleştirilebilir.

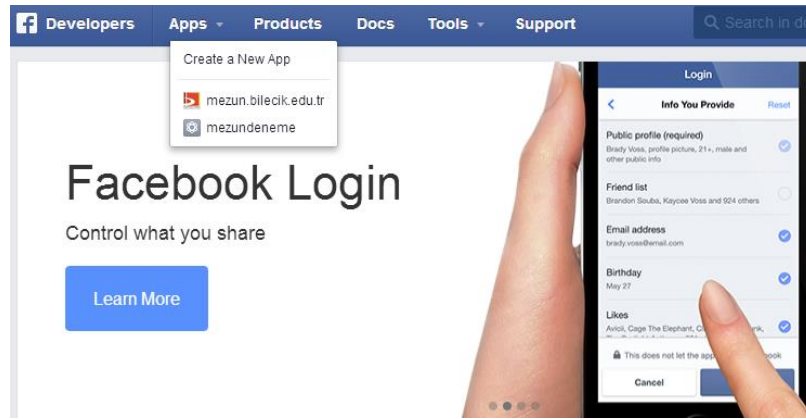
Facebook uyumu konusundaki yardım dokümanlarına <https://developers.facebook.com/> adresinden Docs menüsü altından ulaşılabilir. Bu menü altında ios, android, web ve oyun programlama için nasıl entegrasyon yapılacağı konusunda açıklamalara ulaşılabilir. Facebook kütüphanesinin .Net sistemi ile nasıl kullanılabileceği konusundaki dokümanlara <http://facebooksdk.net/> adresinden ulaşılabilir.

3.1.1.1 Facebook developers kayıt

Facebook giriş uyumu yapabilmek için facebook developers sisteminde bir uygulama oluşturulması gerekir. <https://developers.facebook.com/> adresinden kullanıcı girişi yapılarak yeni uygulama oluşturulabilir. Bu sayfada ayrıca uyum işlemleri için yardım dokümanlarına da ulaşılabilir. Facebook uygulamalarının uyum işlemleri için birçok farklı yapıda kütüphaneye destek verilmektedir. Java script, .Net veya mobil uygulama geliştirirken kullanılabilen kütüphaneler ve kullanım dokümanlarına bu

sayfadan ulařılılabilmektedir.

Uygulama tanımlandıktan sonra facebook.com'dan veri ekebilmemiz ya da yazabilmemiz iin ‘‘Graph Api’’ denilen bir servisi kullanılır. Bu servis ile ilgili detaylara <http://developers.facebook.com/docs/api> baėlantısından ulařılabilir.



Őekil 3.1. Facebook uygulama oluřturma sayfası.

3.1.1.2 Facebook uygulama tanımlama

Sayfaya girildiėinde ste mende bulunan Apps mens altında bulunan ‘‘Create New App’’ seeneėi ile yeni uygulama oluřturma sayfasına gidilir. Bu iřlemden sonra ekrana gelen pencerede bir takım bilgilerin girilmesi istenir. Bu pencerede tanımlanan uygulamanın bir test uygulaması olup olmadıėı, uygulamanın hangi kategoride olduėu gibi bilgiler seilir. Eėer uygulama bařka bir uygulamanın test uygulaması olacaksa ilgili butonunu iřaretlenmesi gerekir. Bařka bir uygulamanın testi olacaksa kategori kısmından ana uygulama seilmelidir. Yeni uygulama oluřturulacaksa uygulamanın kategorisi seilir. Uygulamaya verilen isim ve uygulama iin kullanılacak isim uzayı bilgileri bu sayfada girilir. (Őekil 3.2)

Ekrandaki bilgileri doėru Őekilde girdikten sonra doėrulama kodu sorulur. Gvenlik kodunu da girdikten sonra facebook bizim iin bir uygulama tanımlar. Apps mensnden tanımlanan uygulamalara ulařılabilir. Oluřturulan uygulama facebook sistemi iki farklı deėer retilir. App Id ve App Secret adı verilen bu iki deėer uygulama

çalışırken kullanılması gereken değerlerdir.

Create a New App
Get started integrating Facebook into your app or website

Display Name
Yeni Uygulama

Namespace
YeniUygulama

New! Create test versions of existing apps.

Is this a test version of another app? Daha Fazla Bilgi Al.

Kategori
Bir kategori seç

By proceeding, you agree to the Facebook Platform Policies

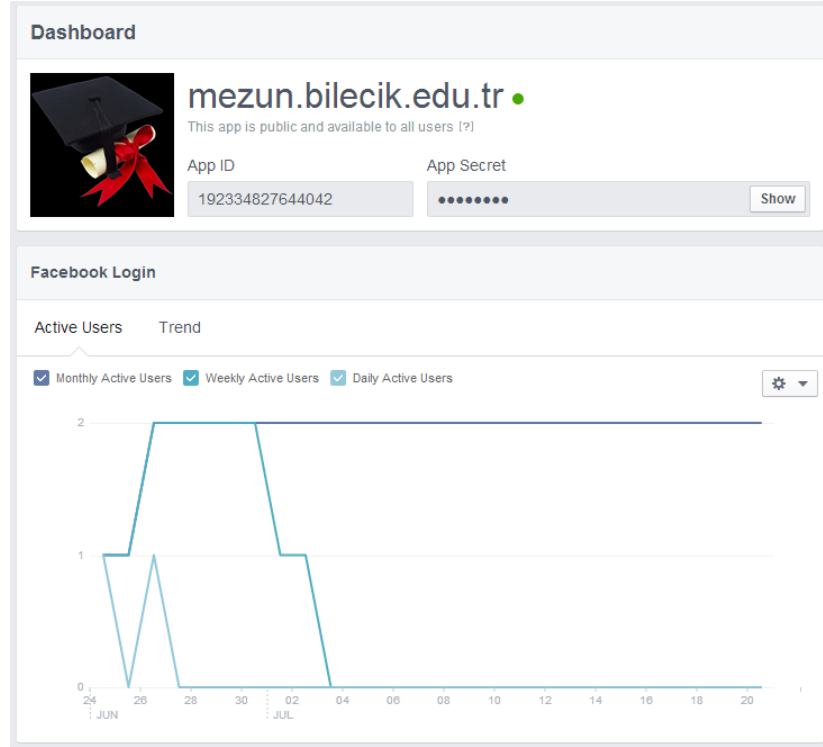
Şekil 3.2. Facebook uygulama bilgileri tanımlama.

3.1.1.3 Facebook uygulama ayarları

Uygulama tanımlandıktan sonra kullanılabilmesi için bir kısım verilerin sisteme girilmiş olması gerekir. Uygulama için iletişim adresi, uygulamanın simgesi gibi bir kısım alanlar doldurulduktan sonra uygulama herkes tarafından kullanılabilir. Bu kısımda zorunlu olarak tanımlanması gereken alanların mutlaka doldurulması gerekmektedir. Zorunlu alanlar doldurulmadığı sürece uygulama üzerinden entegrasyon işlemleri kullanılamamaktadır.

Tanımlanan uygulamaya ait birçok bilgiye ulaşılabilir. Uygulamayı test edecek kullanıcılar, uygulamada görev alacak yazılım geliştiriciler, test yapacak kullanıcılar ayrı ayrı tanımlanabilmektedir. Uygulamanın ne kadar kullanıldığı hakkında sayısal bilgilere ulaşılabilir, uygulama hakkında yapılan yorumlara ulaşılabilir. Günlük kullanım istatistikleri görüntülenebilmektedir. Şekil 3.3 tanımlanmış bir facebook uygulamasının ana ekrandaki bilgileri görülmektedir. Bu bilgilere ek olarak uygulamanın dil desteğiyle ilgili yerleştirme ayarları, uygulama için kullanılan test amaçlı çalışmalar, uygulamaya bağlanacak kullanıcılara ait rol tanımlamaları (geliştirici, test amaçlı kullanıcı gibi), detaylı istatistik bilgileri

bulunmaktadır. Uygulama geliştikçe, bu özelliklerin daha etkin olarak kullanılması mümkündür. Uygulamanın belli tarihlerden ne kadar hizmet verdiğini, örneğin kaç tane login işlemine deste verdiği bilgisini istatistiklerden görmek mümkündür.



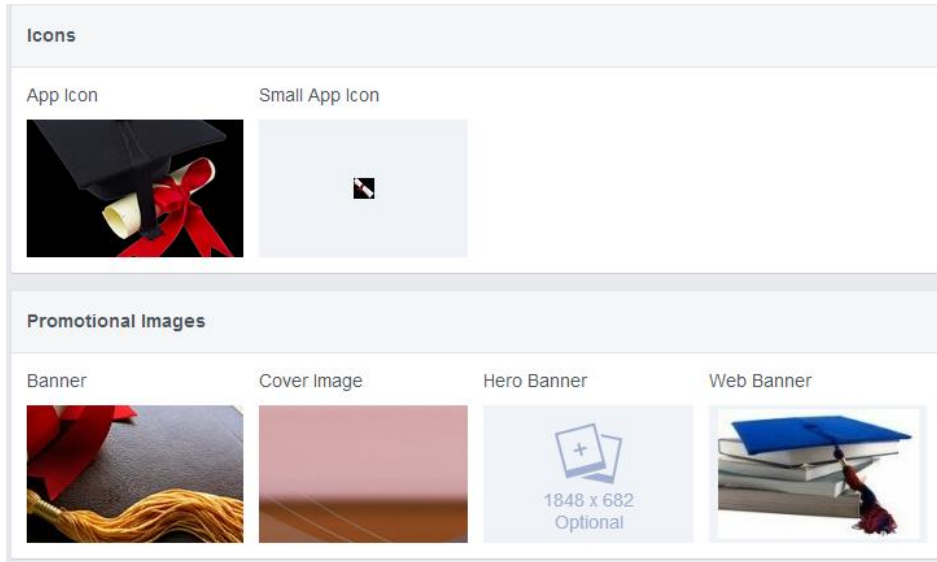
Şekil 3.3. Facebook sisteminde tanımlanmış uygulama.

Facebook sistemi tanımlanan uygulamalara ait çeşitli istatistiki hizmetler de sunmaktadır. Aktif kullanıcı sayısı tarih bazlı görülebilmektedir.

3.1.1.4 Uygulamanın kullanmak istediği veriler

Oluşturulan uygulamanın settings menüsünden uygulama ile ilgili ayarlar yapılır. Basic sekmesinde uygulamanın domain ismi belirtilir. Bu isim facebook sistemine doğrulama amacıyla bağlanabilecek olan sistemi belirtir. Uygulamaya ait bir takım bilgiler sağlandıktan sonra uygulamanın kullanıma açılması sağlanabilir. Uygulama için iletişim bilgileri, uygulamanın simgesi, uygulama için bir banner resmi alanları doldurulduktan sonra uygulama kullanılabilir. Bu bilgileri sağlamak için sağ menüdeki Status-Review menü elemanı seçilir.

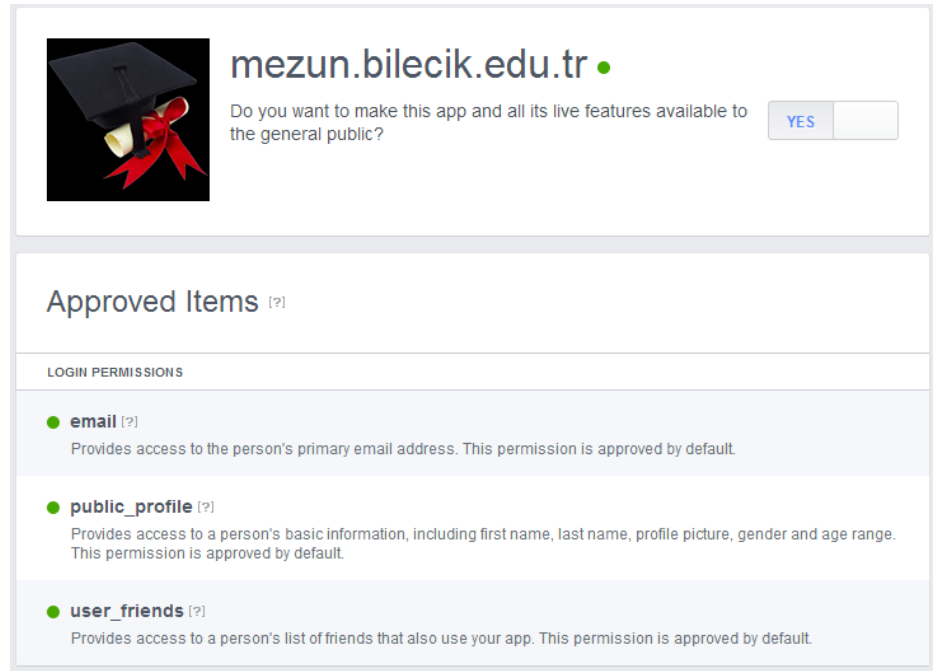
Uygulamaya ilk kayıt esnasında facebook sistemi, kullanıcıya hangi verilerine ulaşılabileceğiyle ilgili bir bilgilendirme gösterir ve onaylayıp onaylamadığını sorar. Kullanıcı verileri onaylarsa kayıt işlemi gerçekleşir ve kullanıcı facebook uyum ile uygulamayı kullanabilir.



Şekil 3.4 Uygulamaya ait logo bilgilerinin tanımlanması.

“Add Items to this Submission” düğmesine tıklanarak uygulamanın kullanıcıların hangi verilerine erişmek istediği belirtilebilir. Örneğin uygulama kullanıcıların arkadaş listesine ulaşmak istiyorsa bu kısımdan arkadaş listesine ulaşmak istediği işaretlenir. Bu kısımda facebook sisteminin sağladığı birçok veri alanı seçilip işaretlenebilir. (Şekil 3.5 ve Şekil 3.6)

Şekil 3.5 facebook uygulamanın daha önceden ayarlanmış olan erişim alanlarını göstermektedir. İlgili alanlar seçilerek istenirse kaldırılabilir. Şekil 3.6 ise facebook sisteminin sağladığı kullanılabilen alanlardan bir kısmını göstermektedir. Çok fazla sayıda alan olduğunda burada tamamı gösterilmemiştir. Örneğin kullanıcının fotoğraflarına da ulaşmak isteniyorsa user_photos alanı işaretlenmesi gerekir. Uygulamalar geliştirilirken kullanıcıyı profil ayarlarından sadece ihtiyaç duyulanlara ulaştırılmalıdır. Çok fazla alana gereksiz izin istenmesi durumunda kullanıcılar uygulamaya izin verme konusunda direnç gösterebilirler.



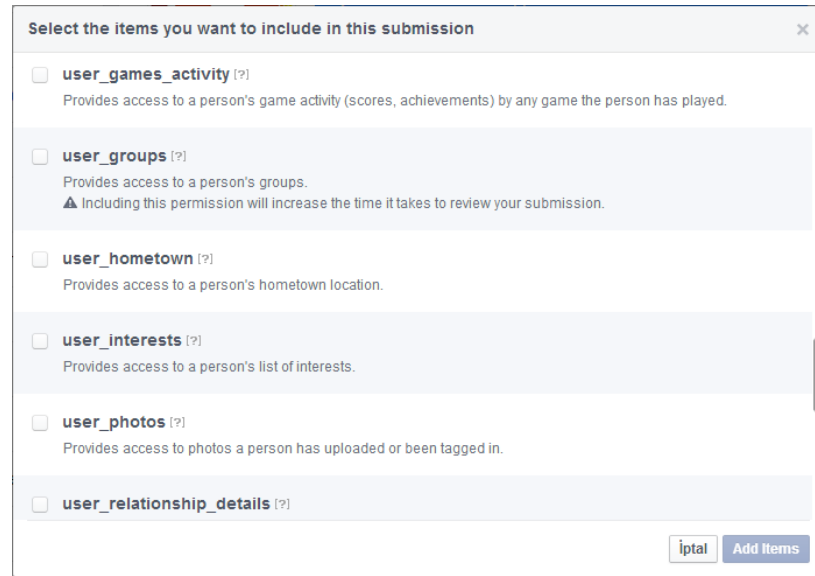
Şekil 3.5. Facebook uygulaması ayarları.

Uygulamayı kullanan facebook üyeleri bağlantı sağladıklarında facebook sistemi tarafından bilgilendirilir. Facebook kullanıcıların bağlanacakları sistemin kullanıcıların hangi verilerine erişmek istediğini kullanıcılara bildirir. Kullanıcı onay verirse sistemle uyumlu şekilde uygulamayı kullanabilir.

Facebook sisteminde kullanıcı doğrulandıktan sonra uygulamanın istediği veriler adres çubuğundan veri olarak sunulur. Facebook sistemi doğrulama işleminden sonra belirtilen adrese adres çubuğu üzerinden bilgileri gönderir.

Facebook sisteminden veri çekebilmek ve yazabilmek için Graph Api denilen bir servis kullanılır. Facebook sisteminin sayfasında bu servisle ilgili detaylı bilgiye ulaşılabilir.¹

¹ <http://developers.facebook.com/docs/api> 12.05.2014



Şekil 3.6. Facebook sisteminin sağladığı verilerden bir kısmı.

Facebook sisteminde kullanıcı doğrulandıktan sonra uygulamanın istediği veriler adres çubuğundan veri olarak sunulur. Facebook sistemi doğrulama işleminden sonra belirtilen adrese adres çubuğu üzerinden bilgileri gönderir.

Facebook sisteminden veri çekebilmek ve yazabilmek için Graph Api denilen bir servis kullanılır. Facebook sisteminin sayfasında bu servisle ilgili detaylı bilgiye ulaşılabilir.¹

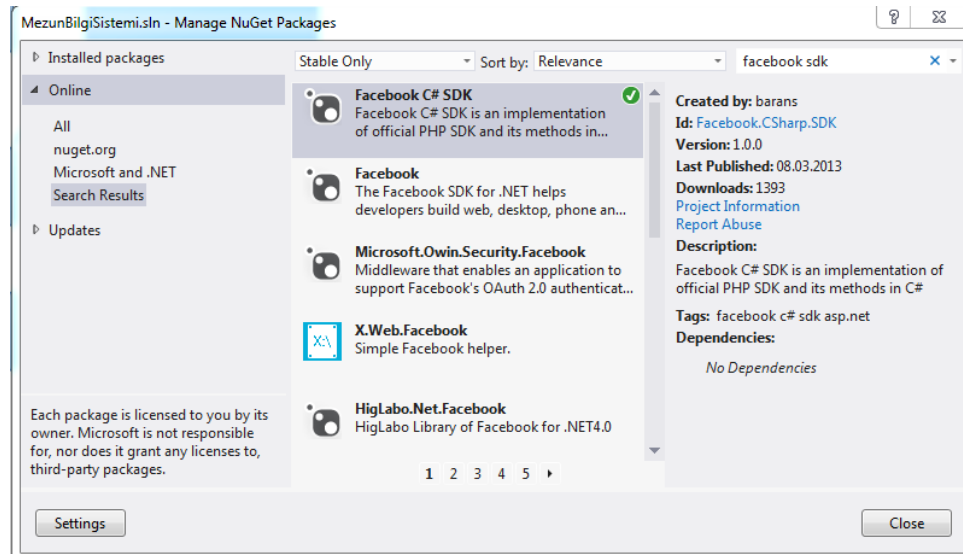
Facebook uygulamalarını kullanabilmek için sabit bir alan adı olması gerekir. Alan adının uygulamanın test işlemleri boyunca sabit olabilmesi için visual studio ayarlarından uygulamanın çalışacağı port numarasının sabitlenmesi sağlanır. Uygulama gerçek sunucusuna atıldığında facebook sistemindeki uygulama ayarlarından alan adı güncellenir. Facebook sisteminden uygulamanın alan adı ayarlarının yapıldığı sayfa Şekil 3.7'de görülmektedir. Website kısmındaki siteurl seçeneğine uygulamanın çalışacağı alan adı tanımlanır.

¹ <http://developers.facebook.com/docs/api> 12.05.2014

Şekil 3.7. Facebook uygulamasında alan adının güncellenmesi

3.1.1.5 Visual studio facebook sdk yükleme

Facebook entegrasyonu kullanacak uygulamada facebook işlemleri için facebook sdk yüklenmesi gerekir. Visual Studio programında nuget package manager ile sdk kolayca yüklenebilir. (Şekil 3.8)

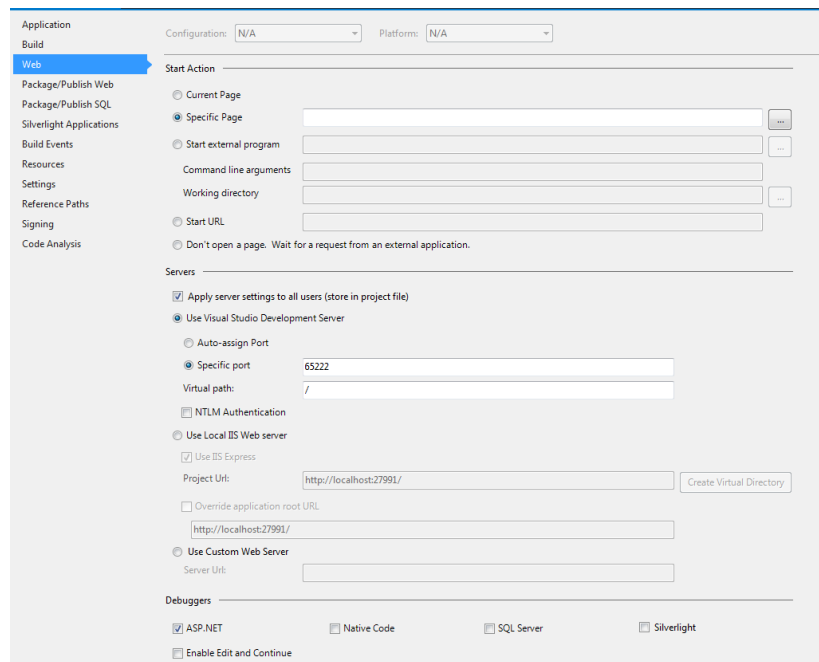


Şekil 3.8. Visual studio facebook sdk yükleme.

Facebook sdk yüklendiğinde yazılımcıya facebook sistemindeki verilerle ilgili işlem yapma imkanı veren sınıflar sunulur. Bu kütüphane kullanılarak facebook sisteminden okunan veriler projede kullanılabilir.

Facebook sisteminde tanımlana bir uygulama ile çalışırken tanımlanan uygulamanın kullanacağı adres belirtilir. Belirtilen adresten farklı bir alan adı ile istek gönderilirse facebook sistemi “Uygulama ayarlarına uygun olmayan bir istek gönderdiniz” uyarısı verir. Proje geliştirilirken deneme amaçlı doğrulama işlemleri yapılması ihtiyacı söz konusu olur. Visual Studio programı varsayılan ayarlarda proje her derlendiğinde projeyi farklı bir porttan çalıştırır. Bu durumda port numarası sürekli değişeceğinden facebook uygulaması ile ilgili hata mesajı alınacaktır. Bu sorunu aşmak için Visual Studio projesinin özelliklerinden kullanılacak port numarası sabitlenir. Sabitlenen port numarasıyla birlikte localhost adresi facebook sisteminde belirtilir.

Şekil 3.9 Visual studio 2012 projesinin portunun sabit bir değere nasıl ayarladığını göstermektedir. Bu ayarlama işlemi geliştirme yapılırken sürekli olarak uygulamanın portunun yeniden ayarlanması ihtiyacını ortadan kaldırır. Varsayılan ayarlarda projenin debug işlemleri yeni portlar üzerinden gerçekleştirilir.



Şekil 3.9. Visual studio 2012 ile proje özelliklerinden portu sabitleme.

3.1.2 Visual studio 2012

Geliştirme ortamı olarak visual studio IDE'sinin 2012 sürümü kullanılmıştır. Visual studio Microsoft tarafından yazılım geliştirme aracı olarak sunulan güçlü bir araçtır. Yazılım geliştirmede kolaylık sağlayan birçok özellik sunmaktadır. Geliştirilen yazılımların mimarilerine ait şemaların, sınıfların bağımlılıklarını gösteren şemaların bu araç ile doğru kolay bir şekilde oluşturulabilmektedir. Bu sayede geliştirme yaparken bir yandan da dokümantasyon işlemleri için destek sağlanmış olmaktadır.

Visual studio kullanılarak birçok farklı dil ile geliştirme yapılabilir. Mezun sisteminde C# dili kullanılarak geliştirme yapılmıştır. Visual Studio masaüstü uygulamalar, web uygulamaları, mobil uygulamalar geliştirmek için uygun araçlara sahiptir.

Visual Studio ortamı yazılım geliştiricilere kolaylık sağlayan birçok eklentini yüklenmesine de destek verir. Mezun sistemi geliştirilirken visual svn eklentisi kullanılarak sürüm takip sistemi uyumu sağlanmıştır. Visual svn eklentisi Tourtoise SVN sistemi, ile entegre şekilde çalışmaktadır. Tourtoise SVN sistemi ile birlikte çalışan, redmine isimli sistem de iş takibini kolaylaştırmakta, proje görevlendirmelerini ve işlerini yönetilmesini sağlamaktadır.

Visual studio takım halinde yazılım geliştirmeye yönelik iyileştirmeler barındırmaktadır. Microsoft firmasının sürüm takip sistemi olan Team Foundadion Server ile tümleşik şekilde çalışmaktadır. Bu sayede birçok geliştiricinin ortak projelerde etkin şekilde çalışmasına imkân sağlamaktadır.¹

3.1.3 Redmine

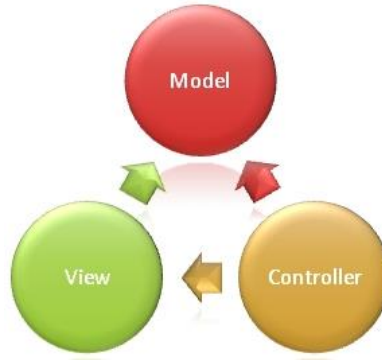
Redmine açık kaynak kodlu, web tabanlı, ücretsiz bir proje yönetim ve hata izleme aracıdır. Redmine ile projedeki hataların takibi, görevlendirmeler, proje dosyalarının sürüm takipleri, projede çalışan personellerin performansları takip

¹ <http://www.visualstudio.com/tr-tr/explore/app-lifecycle-management-vs>

edilebilir ve yönetilebilir. Sisteme takvimli olarak girilen hata ve düzeltmelerle ilgili bir gant diyagramı çıkarılabilmesine imkan tanır. Proje sürümleri arasındaki yenilikleri ve düzeltmeleri canlı sistem üzerinden yönetmeyi sağlar. Redmine sistemi üzerinde birden fazla proje tanımlanabilir ve yönetilebilir. Her projede görev yapan yazılımcı, yönetici ve raporlayıcı rolleri tanımlanabilir.¹

3.1.4 MVC (Model View Controller)

MVC 1979 yılında tanımlanmış olan yazılım mimarisi desenidir. MVC deseninin tanımlanması eskiye dayanır. Bu desenin daha fazla tanınması ve kullanılmaya başlanması ise son yıllarda Microsoft firmasının geliştirdiği MVC framework sayesinde mümkün olmuştur. Yazılım uygulamalarında veri tabanı ve kullanıcı arayüzü katmanlarının birbirinden soyutlanması esasına dayanır. Böylece veriler (model) ve arayüzler (view) birbirinden bağımsız olarak tasarlanabilir.²



Şekil 3.10. Asp.Net MVC Modeli.³

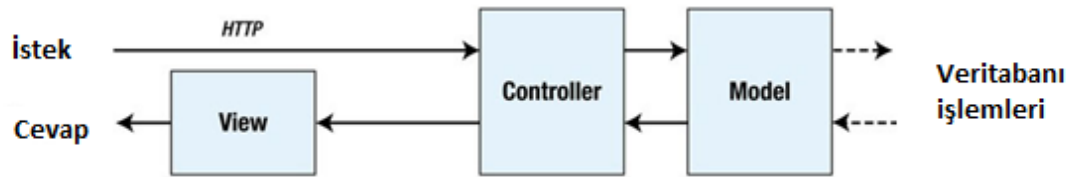
İstekler doğrudan controller katmanına yapılır. Her view mutlaka bir controller'a bağlı çalışır ancak her controller için bir view olmak zorunda değildir. Model katmanı entity data model gibi veri tabanı bağlantı nesnelerini barındırır. Model

¹ <http://www.ozguryazilim.com.tr/bt-altyapi-cozumleri/proje-yonetimi-redmine>

² <http://mcansozeri.wordpress.com/2011/11/27/mvc-nedir-asp-net-mvc-nedir/>

³ <http://www.asp.net/mvc/tutorials/older-versions/overview/asp-net-mvc-overview>

katmanında view sayfalarında kullanılmak istenen veri modelleri sınıf olarak tanımlanabilir. Bu sayede model binding işlemi ile metotlara nesnelere aracılığıyla formlardaki veriler aktarılabilir. Formdaki veriler yine nesnelere aracılığıyla controller katmanına aktarılabilir. Controller katmanındaki metotlar hangi view elemanını görüntülemek istiyorlarsa view'in kullandığı nesne modeline uygun verilerle ilgili view'in görüntülenmesini sağlayabilirler.



Şekil 3.11. Asp.Net MVC mimarisinin çalışması.¹

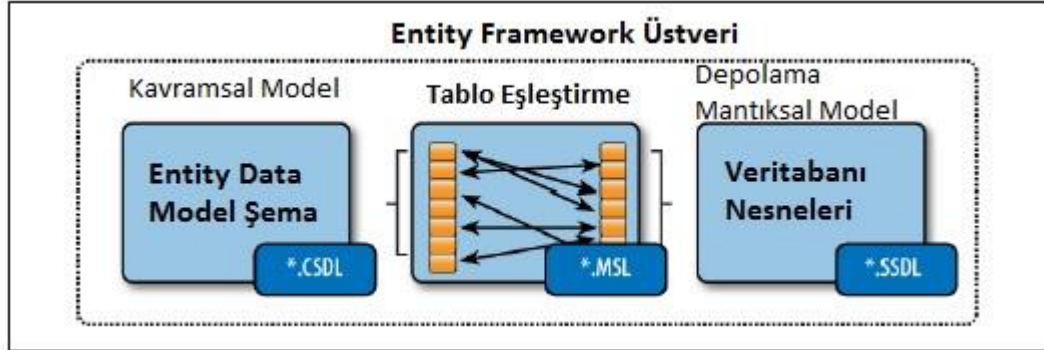
3.1.5 Entity framework

Mezun sistemi geliştirilirken Microsoft Firmanının orm aracı olan Entity Framework kullanılmıştır. Entity framework veri tabanındaki tablaları programlama tarafında class'lara dönüştürerek veri tabanına veri yazma okuma işlemlerinin nesnelere üzerinden yapılmasını sağlar. Entity framework özellikleri kullanılarak veri tabanı işlemleri kolaylaşır. Veri tabanına yazma okuma işlemleri için harcadığımız zaman azalır. Entity framework'ün repository pattern gibi projelere uygun desenlerle birlikte kullanılması yazılımın geliştirilme sürecini hızlandırır, yeni özellikler ekleme, proje ekibine yeni üyeler dahil etme sürecini kolaylaştırır.

Entity framewor kullanımı için üç farklı yöntem mevcuttur.

- Model first
- Code first,
- Database first

¹ <http://www.ahmetcigsar.com/yazilim/aspnet-mvc/aspnet-mvc-nedir/>



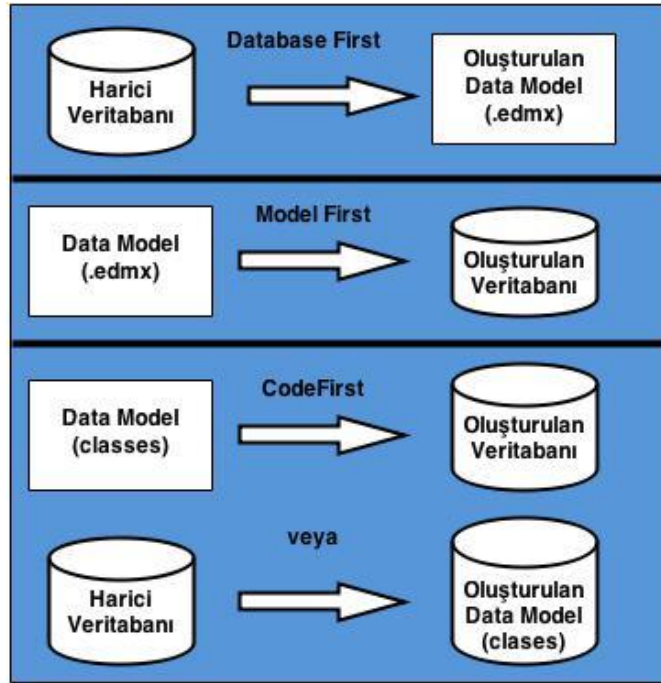
Şekil 3.12. Entity framework yapısı. (Lerman J. , 2010)

Model first yaklaşımında Visual Studio aracı kullanılarak sistem nesnelерinin modeli oluşturulur. Kullanılan arayüz ile nesnelер arası ilişkiler tanımlanabilir, nesnelere ait özellikler, kısıtlamalar tanımlanabilir. Arayüzde tanımlanan model tasarımı bittikten sonra bu modele uygun veri tabanını otomatik oluşturulması sağlanır. Kod tarafında kullanılacak sınıflar Visual Studio programının sağladığı bir arayüz ile hazırlanır.

Code first yaklaşımında sınıflar C# kodları ile oluşturulur. Kodlama yoluyla modele ait nesnelер ve veri tabanının oluşturulması sağlanabilir. Hazır bir veri tabanı yoksa ve geliştirme için zaman sorunu da söz konusu değilse tercih edilen bir yöntemdir. Bu yöntem, veri tabanında tablo olarak var olacak sınıflardan farklı olarak yeni sınıflara da ihtiyaç duyulabilen projelerde tercih edilir. Bu yöntemde sistemdeki tüm veri yapısı programcı tarafından teker teker yazılır. Hazırlanan model tasarımı bitirildiğinde veri tabanının bu modele göre oluşturulması sağlanır. Sistem tanımlanan sınıfların özelliklerine ve sınıflar arasındaki bağımlılıklara bakarak uygun ilişkisel veri tabanını oluşturur.

Hazır bir veri tabanı varsa ve hızlıca data model oluşturulmak isteniyorsa, kısıtlı zamanlarda geliştirilmesi gereken uygulamalar söz konusu ise database first yaklaşımı ile veri tabanındaki tablolardan entity nesnelерinin oluşturulması sağlanabilir. Entity framework aracı tablolar arasındaki ilişkileri de dikkate alarak tüm tabloları programlama tarafında kullanılmak üzere nesnelere dönüştürür. Veritabanında tanımlanmış prosedür, fonksiyon, view gibi nesnelерin de entity model tarafına

aktarılması sağlanabilir. Bu sayede hazır bir veritabanı üzerinde yapılan çalışmalar geliştiricinin kullanımına sunulabilir.



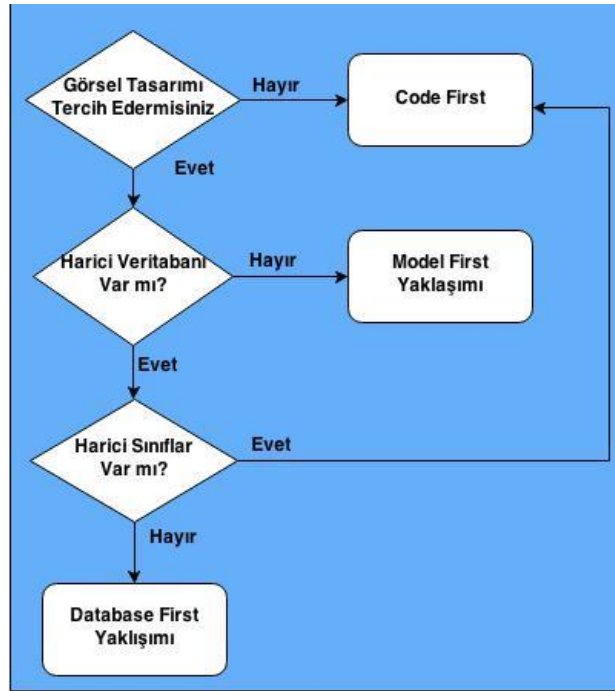
Şekil 3.13. Entity framework yaklaşımları.¹

Hangi yaklaşımın ne durumda kullanılacağıyla ilgili Julia Lerman'ın akış diyagramı açıklayıcı bir kaynaktır (Şekil 3.14).

Hangi yaklaşım kullanılırsa kullanılsın temelde yapılan işlem programlama dilini kullanarak doğrudan veriler üzerinde işlem yapılabilecek bir yapı oluşturmaktır. Bu sayede programcı sql dilinin yazım kuralları ile uğraşmadan, doğrudan kullandığı yazılım dilinin kuralları ile verileri sorgulayabilir, yazma ve okuma işlemleri gerçekleştirebilir. Entity framework aracının yeni sürümleri çıktıkça yazılan kodların sql diline daha performanslı dönüştürülmesi, sınıfların daha esnek oluşturulması, loglama gibi işlemler için araya girilerek işlemler yapılması gibi farklı konularda iyileştirmeler yapılmaktadır. Yeni sürümlerde grafik tasarım aracında da iyileştirmeler ve yeni

¹ <http://www.erenguvercin.com/2013/05/entity-framework-yaklasmlar-ve.html>

özellikler eklenmektedir.

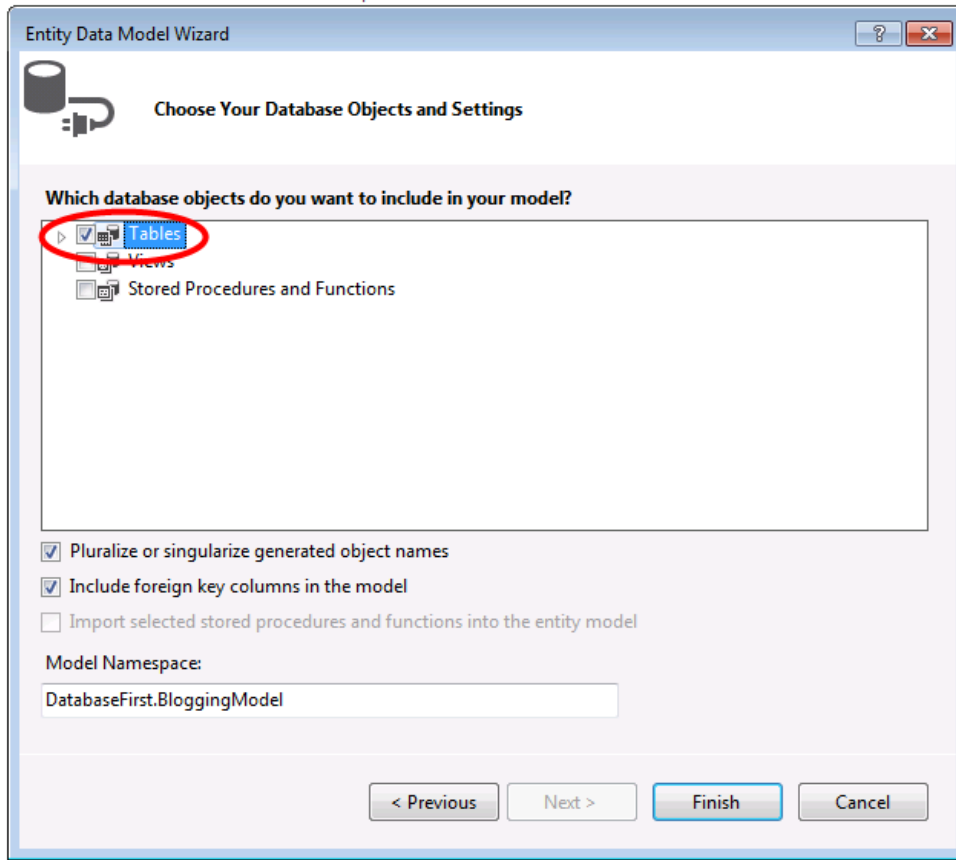


Şekil 3.14. Hangi yaklaşım ne zaman kullanılabilir.¹

Bu çalışmada entity framework geliştirme yöntemlerinden database first yaklaşımı tercih edilmiştir. Zaten var olan öğrenci bilgi sistemi veri tabanı, mezun sistemi veri tabanı kullanılarak data model oluşturulmuştur. Data model oluşturulurken entity framework aracından yararlanılmıştır. Oluşturulan data model class library olarak projeye dahil edilmiştir. İş katmanına referans verilerek oluşturulan data model sınıfları kullanılabilir.

Data model katmanı ayrı bir kütüphane projesi olarak sisteme dahil edilmiştir. Bu sayede data model sınıflarına özel işlemler, genişletme metotları gibi çalışmalar iş katmanından bağımsız olarak tasarlanabilir. İş katmanında kullanılan tüm sınıfların data modelden bağımsız olmasını sağlayacak bir katman da tasarlanabilir. Bu çalışmada böyle bir katman tercih edilmemiştir.

¹ <http://www.erenguvercin.com/2013/05/entity-framework-yaklasmlar-ve.html>



Şekil 3.15. Visual Studio kullanılarak veri tabanından model oluşturma.

Entity model oluşturulurken sadece veri tabanındaki tablo elemanları entity modele dahil edilmiştir. Bu sayede veri tabanındaki tüm tablolar nesnelere olarak uygulamaya aktarılmıştır.

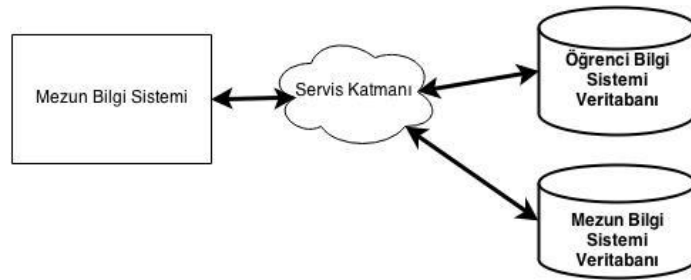
3.1.6 MSSQL server 2012

Mezun sisteminde MSSQL Server 2012 veri tabanı yönetim sistemi kullanılmıştır. Bu veri tabanı yönetim sisteminin ücretsiz bir sürümünü firma sitesinden indirip kurmak mümkündür.¹ Microsoft firmasının araçları ile uyumlu çalıştığı, kullanım kolaylığı sağladığı için bu araç tercih edilmiştir. Entity framework oluşturulurken bu veri tabanında bulunan tablo, view ve prosedürlerden de yararlanılabilmektedir.

¹ <http://www.microsoft.com/en-us/download/details.aspx?id=29062>

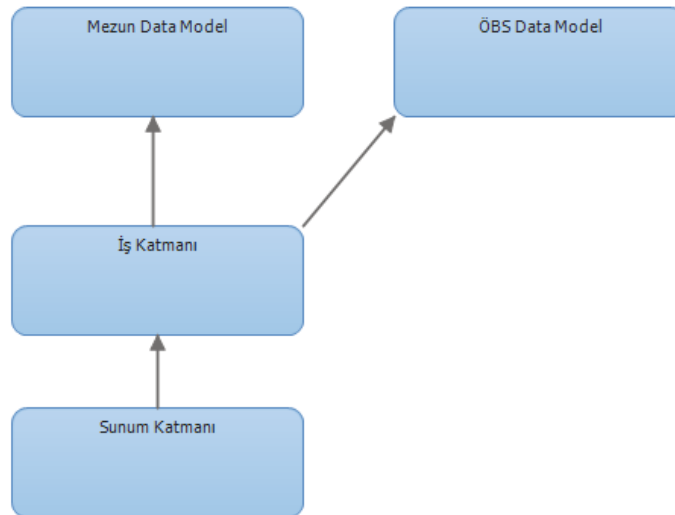
3.2 Mezun Sistemi Mimarisi

Sistemin genel mimarisini gösteren şekil aşağıda görülmektedir. Sistem uygulama katmanı, uygulama katmanına verileri sunan ve çeşitli algoritmaları çalıştıran servis katmanı ve veri tabanı katmanından oluşmaktadır. Şekil 3.16 sistemin genel mimarisini göstermektedir.



Şekil 3.16. Sistemin genel mimari yapısı.

Mezun sistemi geliştirilirken MVC deseni kullanılmıştır. Projede üç farklı katmana görevlerin dağıtılması sağlanmıştır. MVC deseni sayesinde web uygulamasında zaten veri katmanı view tarafından soyutlanmış durumdadır. Mezun bilgi sisteminin mimari yapısı Şekil 3.17’de görülmektedir.



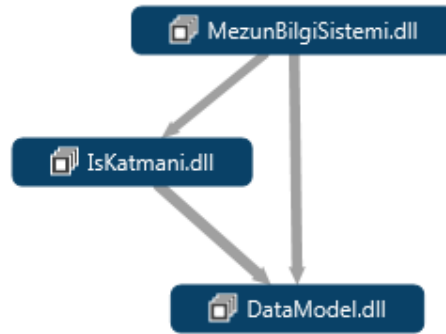
Şekil 3.17. Projenin genel mimari yapısı.

Bu desenin yanında proje geliştirilirken entity modeli barındıracak bir katman

planlanmıştır. Veri tabanındaki tabloları program tarafından linq sorgularıyla erişilmesini sağlayan entity model bu katmanda oluşturulmuştur. İkinci katman data model üzerinde veri tabanı işlemlerini gerçekleştirmek üzere yazılmış sınıflardan oluşmaktadır. Bu sayede web uygulamasının kullandığı tüm metotların iş kurallarının uygulayıcısı olarak bu katman planlanmıştır. Web uygulamasının bulunduğu katman ise belirtilen modeller ile çalışan view ve controller metotlarını barındırmaktadır. Mezun sisteminin visual studio kullanılarak proje üzerinden alınan sınıf bağımlılıkları görünümü Şekil 3.18’de görülmektedir.

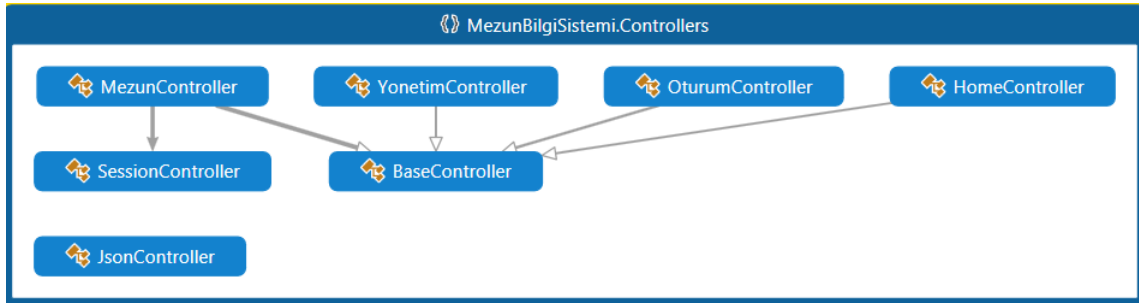
Katmanların birbirinden bağımsız planlanması güncellemeler için sistemi esnek ve yönetilebilir bir yapıya kavuşturmuştur. Görevler katmanlar arasında paylaştırılarak kod karmaşasının engellenmesi amaçlanmıştır.

Her bir katman içerisinde bulunan sınıfların detaylı görünümünün inceleyecek olursak data model katmanını entity sınıflarını barındırdığını, IsKatmani isimli katmanını projede ihtiyaç duyulan listeleme ve veri ekleme işlemlerini yapan sınıfları barındırdığını görebiliriz.



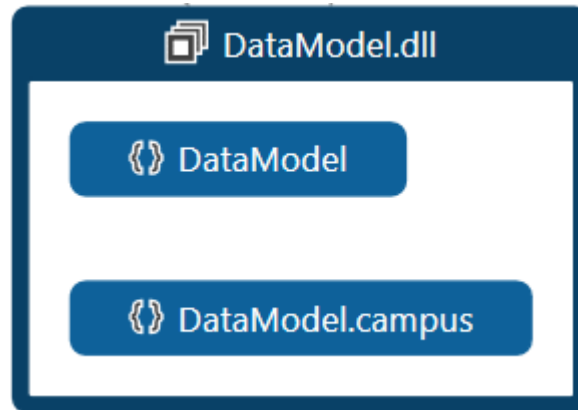
Şekil 3.18. Mezun bilgi sistemi proje bağımlılıkları.

Mezunbilgisistemi isimli dll dosyası da web uygulamasına ait sınıfları barındırmaktadır. Mezun bilgi sistemi projesindeki controller katmanındaki bağımlılıklar Şekil 3.19’da görülmektedir. Sınıf bağımlılıklarında görüldüğü üzere controller katmanındaki sınıflar güvenlik, loglama gibi işlemler için kullanılan temel bir sınıftan türetilmişlerdir.



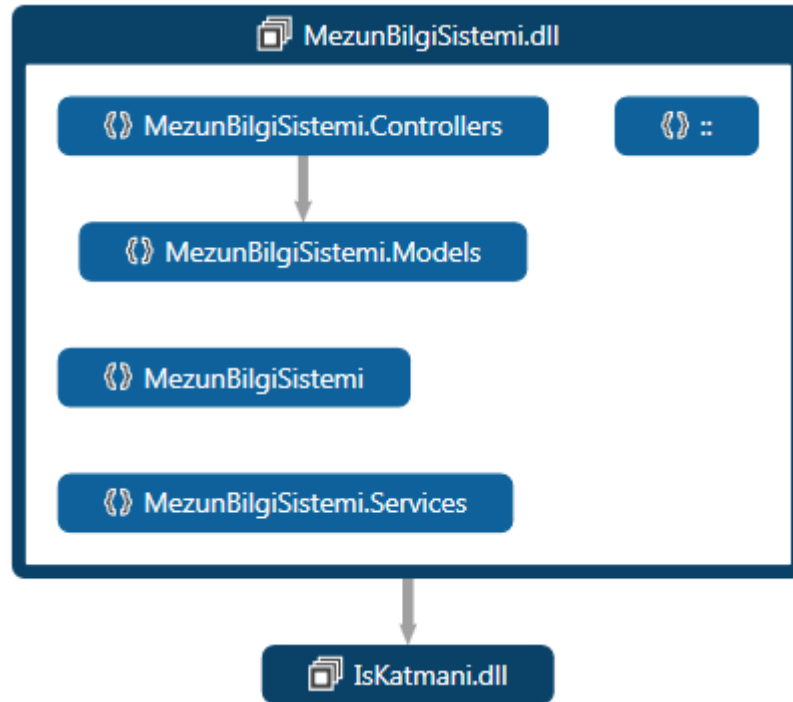
Şekil 3.19. Controller bağımlılıkları.

Data model katmanında bulunan iki farklı temel model tanımlanmıştır. Bu modellerden birinin öğrenci bilgi sisteminden veri okuma ihtiyacı için kullanılan model diğeri de mezun sistemi veri tabanından veri okumak için kullanılan modeldir. Entity framework kullanılarak öğrenci bilgi sisteminden yalnızca ihtiyaç duyulan tablolar sisteme dahil edilmiştir. Data model oluşturulurken iki ayrı veri tabanından gelen veriler için ayrı isim uzayı oluşturulması tercih edilmiştir. (Şekil 3.20)



Şekil 3.20. Data model katmanı sınıf görünümü.

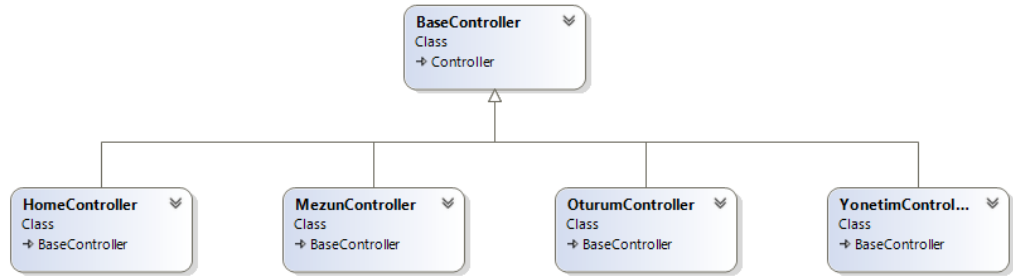
Mezun sisteminin web uygulaması kısmındaki sınıflara göz atacak olursak MVC mimarisine uygun olarak controller sınıflarının olduğunu görebiliriz. Sunum katmanı olarak planlanan MVC web uygulamasının sınıf bağımlılıkları Şekil 3.21’de görülmektedir.



Şekil 3.21. Mezun sistemi web uygulamasına ait sınıflar.

Web uygulamasında controller sınıfı, iş katmanındaki metotları kullanmaktadır. Controller sınıfının detay görünümü aşağıda görünmektedir. Controller sınıfının içinde güvenlik işlemleri, oturum işlemleri, yönetim işlemlerini sağlayan sınıflar bulunmaktadır. Projedeki her bir controller temel bir controller nesnesinden türetilerek oluşturulmuştur. Bu sayede her bir controller içinde ulaşılmak istenen ortak nesnelere ulaşılması sağlanmış, her bir controller sınıfının temel bir takım özelliklere sahip olması sağlanmıştır.

Sistem mimarisinde temel olarak dört farklı controller sınıfı tanımlanmıştır. Oturum işlemleri, mezun işlemleri, yönetim işlemleri ve ana sayfa işlemleri için ayrı ayrı controller tanımlanmıştır. Her bir controller tanımlanırken güvenlik amacıyla oluşturulmuş niteliklerle işaretlenmiş ve temel bir controller sınıfından türetilmiştir. Temel controller sınıfı sayesinde oturum açan kullanıcı bilgilerine her bir controller üzerinden ulaşılabilmesi kolaylaştırılmıştır.



Şekil 3.22. Controller sınıflarının bağımlılıkları.

3.3 Mezun Sisteminde Güvenlik

Mezun sisteminde güvenliği sağlamak amacıyla MVC framework'ünün yeteneklerinden yararlanılmıştır. Controller katmanındaki her bir metod çalıştırılmadan önce kullanıcı yetkisi olup olmadığı kontrol edilmiştir. Bunu sağlamak için niteliklerden faydalanılmıştır. Güvenlik amacıyla yazılmış olan bir nitelik `ActionFilterAttribute` sınıfından miras alınarak oluşturulmuştur. `ActionFilterAttribute` sınıfı filtreleme amacıyla kullanılan bir sınıftır. Bu sınıftan türetilerek oluşturulan sınıflar nitelik olarak controller metodlarında belirtilebilir. Bu sayede bu niteliğin belirtildiği metod veya sınıf projenin herhangi bir yerinden çağrıldığında sistem istenen güvenlik kontrollerini yapabilecektir. Güvenlik amacıyla tanımlanmış örnek bir nitelik Şekil 3.23 numaralı şekilde görülebilir.

```

public class SessionController : ActionFilterAttribute
{
    public override void OnActionExecuting(ActionExecutingContext filterContext)
    {
        if (!HttpContext.Current.User.Identity.IsAuthenticated)
        {
            if (!HttpContext.Current.Response.IsRequestBeingRedirected)
            {
                filterContext.Result = new RedirectToRouteResult(new RouteValueDictionary { { "action", "Giris" }, { "controller", "Oturum" } });
            }
        }
        FacebookService fb1 = new FacebookService();
        Facebook.FacebookClient c1 = new Facebook.FacebookClient();
    }
}

```

Şekil 3.23. Nitelik tanımlama.

Nitelikler uygun şekilde tanımlanarak belli rollere sahip kullanıcılara yetkiler tanımlanabilir. Metotlara istekler geldiğinde nitelikler sayesinde kullanıcının rolüne ait

yetkileri kontrol edilebilir.

Kullanıcı doğrulama yöntemi olarak form authentication yöntemi kullanılmıştır. Doğrulama verileri öğrenci bilgi sistemi veri tabanından sorgulanmaktadır. Kullanıcılar facebook hesaplarıyla sisteme kayıt olarak sosyal medya hesaplarıyla da oturum açabilirler.

Kullanıcılara ait farklı rollere göre sistemde farklı sayfalara erişebilecekleri alt yapı sağlanmıştır. Yönetim sayfalarına sadece sistem yönetici rolünde olanlar bağlanabilirken, öğrenciler sadece mezunlarla ilgili sayfalara bağlanabilmektedir. Güvenlik önlemini sağlamak için oluşturulan her bir sınıf veya metot bir nitelik ile işaretlenmiştir. Böylece metot bazlı bir yetkilendirme sağlanmıştır.

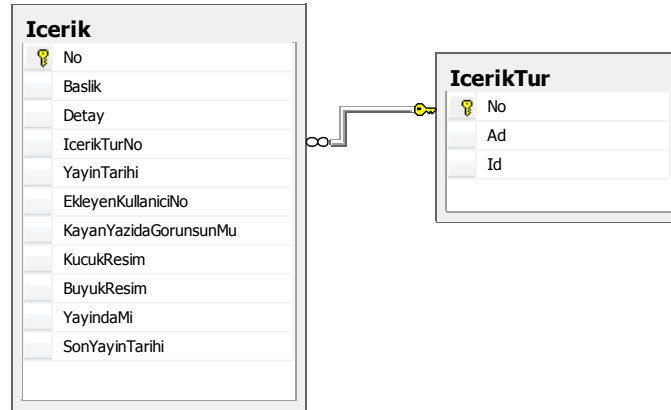
Facebook işlemlerini kolaylaştırmak amacıyla bir sınıf oluşturulmuştur. Bu sınıf kullanılarak facebook oturum bilgilerinin yönetimi sağlanmıştır. Facebook oturum açma, oturum kapatma, kullanıcı id değerinin elde edilmesi gibi işlemler için bu sınıf kullanılmaktadır.

3.4 Mezun Sistemi Veri tabanı

Mezun sistemi için MS SQL Server veri tabanı yönetim sistemi seçilmiştir. Veri tabanı tasarımı yapılırken normalizasyon kurallarına dikkat edilmiş verit tekrarı olmayan bir veri tabanı tasarlanmıştır. İlişkisel veri tabanı yönetim sistemlerinin sağladığı özellikler sayesinde veri tekrarı olmadan tek bir yerden verilerin yönetilebilmesi sağlanmıştır. Veri tabanını incelerken kullanılan verilere göre gruplandırarak incelemek faydalı olacaktır.

3.4.1 Sisteme eklenen içerik bilgilerini tutulduğu tablolar

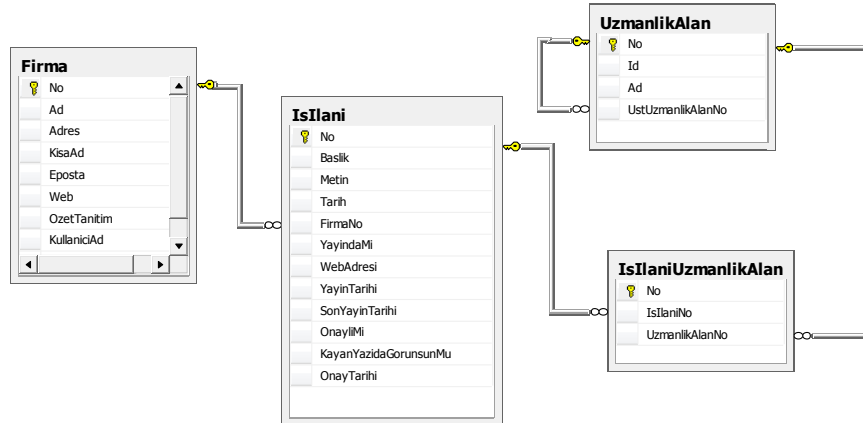
İçerik tablolarında sisteme girilen, etkinlik, duyuru, iş ilanı gibi içerikler tutulmaktadır. Tüm bu türler bir tür numarasıyla birbirinden ayrılmıştır. İçeriklerin türü değişse de veri alanları aynı kaldığından her içerik türü için ayrı tablo planlanmamış, içerikler tür numaraları ile birbirinden ayrılmıştır. Sisteme tanımlanmış içeriklerin kullanıcı oturum açtığı anda kullanıcının ana sayfasında da görünmesi sağlanmıştır.



Şekil 3.24. İçerik tabloları.

3.4.2 Mezun sistemi iş ilanı tabloları

İş ilanları uzmanlık alanlarıyla ilişkilendirilerek ilana uygun mezunlara daha kolay ulaşılabilmesi sağlanmıştır. İş ilanları firmalarla bağlantılı olarak sisteme kayıt edilebilmektedir. İş ilanı modülü etkin kullanılabilirse mezunlar için bir kariyer merkezi görevi görebilir. İş ilanları ile ilgili sayfaların firma ihtiyaçlarına göre tanzim edilmesi ile de firmaların sisteme ilgi göstermeleri ve nitelikli elemana ulaşmalarını sağlayabilir. Şekil 3.25 iş ilanı konusuyla alakalı verilerin ilişkilerini göstermektedir.



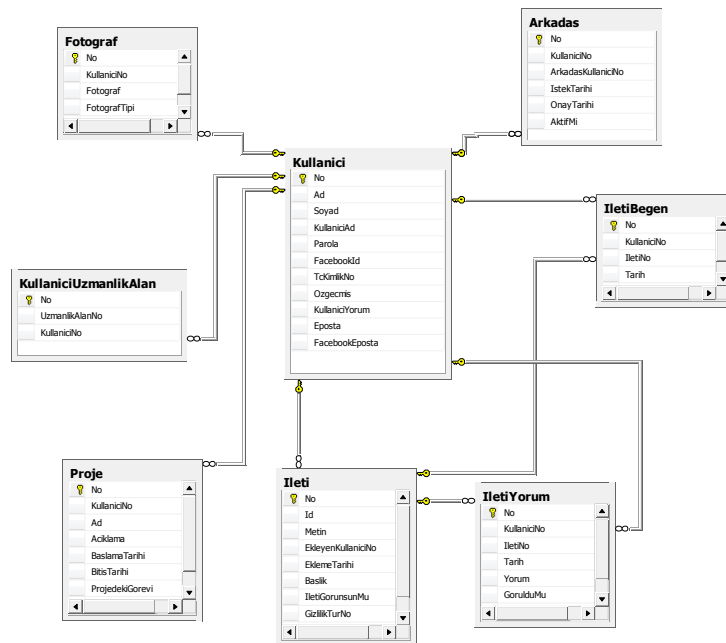
Şekil 3.25. İş ilanı ilişkileri.

3.4.3 Mezun sisteminde mezunlara ait veriler

Mezunlara ait temel veriler öğrenci bilgi sisteminden sağlanmaktadır. Mezunların iletileri, iletilere yaptıkları yorumlar, mezunlara gelen mesajlar, arkadaşlar

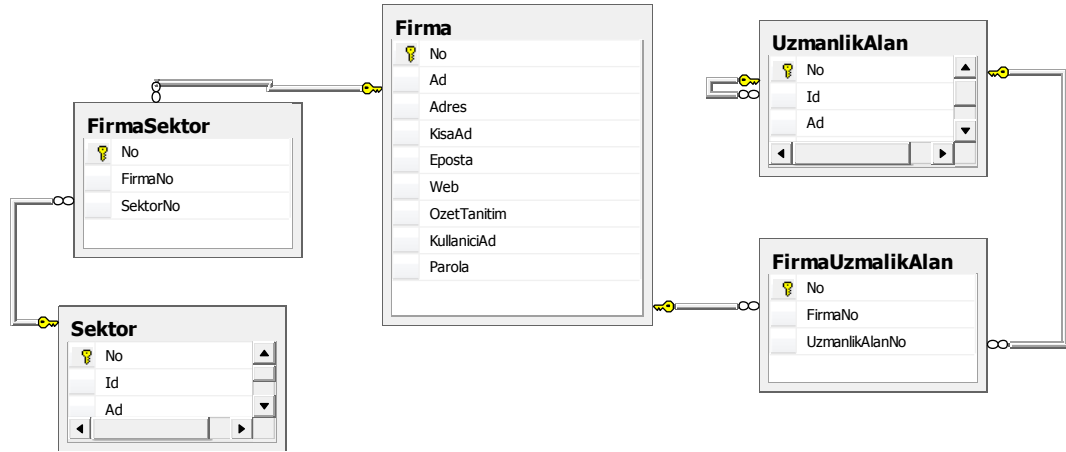
gibi veri alanları planlanmıştır. Kullanıcıya ait verilerin ilişkilerini gösteren şema şekilde görülmektedir. (Şekil 3.26) Kullanıcıların facebook hesap bilgilerinin kayıt altına alınması için doğrulama işleminden sonra facebook sisteminin sağladığı facebookId verisi kullanılmıştır. Bu verinin yanında kullanıcının facebook hesabında kullandığı eposta adresi de kayıt altına alınmıştır. Facebook hesaplarının aktif olarak kullanıldığı düşünülürse mezunların güncel eposta bilgilerinin bu sayede kayıt altına alınması sağlanmıştır.

Mezunlara ait verilerin normalize yapıda kayıt altına alındığı veri tabanı tasarımından görülebilir. Burada veri tabanındaki tüm tablolar yerine temel bir takım tablolar normalize yapıyı göstermek amacıyla belirtilmiştir. Mezunlar çalıştıkları projelerini, iş ve eğitim tecrübelerini sisteme girebilirler. Özgeçmişe ait bilgilerin tamamlanmasıyla hazır bir özgeçmiş formatını sistemden alabilirler. Kullanıcılar uzmanlık alanlarını sisteme tanımlayabilir bu sayede işverenlerin aradıkları vasıflara göre mezunlarla iş ilanları sistem tarafından firmaya veya mezunlara tavsiye edilebilir. Uzmanlık alanlarının güncel olarak takip edilebilmesi sayesinde ders içerikleri yeni uzmanlık alanı ihtiyaçlarına göre yenilebilir ve günceli takip etmesi için mezun sisteminden veri elde edilebilir.



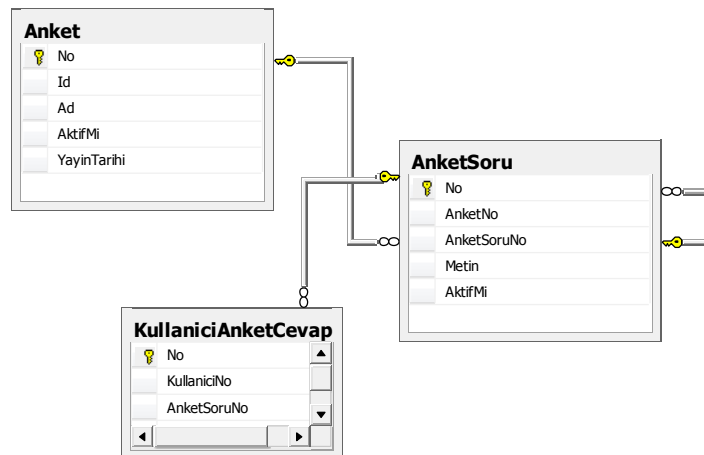
Şekil 3.26. Temel kullanıcı verileri ilişkileri.

Firmalar faaliyet gösterdikleri sektörler göre tutulabilecek şekilde tasarlanmıştır. Böylece hangi sektörden ne kadar iş ilanı geldiği bilgisi elde edilebilecektir. Firmalarla etkin bir işbirliği oluşturularak mezunlar için kariyer imkanları sistem üzerinden sağlanabilir. Şekil 3.27 firmaya ait verilerin ilişkilerini göstermektedir.



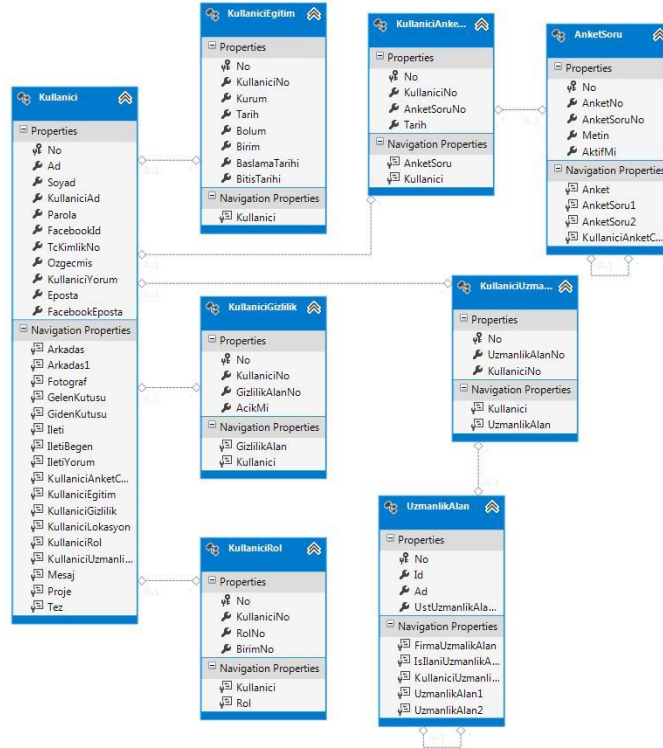
Şekil 3.27. Firma verileri arasındaki ilişkiler.

Anket verileri tek bir tabloda tanımlanabilecek şekilde tasarlanmıştır. Hiyerarşik veriler tek bir tabloda rekürsif yapıda planlanmıştır. Bu sayede sorular ve soruların cevapları tek bir tabloda tanımlanabilmiştir. Alt sorular tanımlanabilmesine alt yapı hazırlanmıştır. Anket sorularına ait hiyerarşik veri yapısının tanımlanması Şekil 3.28’de görülmektedir.



Şekil 3.28. Anket sorularının tanımlanması ve cevaplanması veri ilişkileri.

Kullanıcı verilerine ait sınıfların ilişkilerini gösteren şema Şekil 3.29’da sunulmuştur. Kullanıcı nesnesine bağlı olan pek çok veri alanı bulunduğu için burada sadece bir kısım alanlar belirtilmiştir.



Şekil 3.29 Kullanıcı verilerine ait entity model görünümü. projeler, hazırladıkları tez bilgileri sistemde tutulabilmektedir. Kullanıcıların harita üzerinde hangi konumda oldukları bilgisinin tutularak Google Maps api kullanılarak harita üzerinde kullanıcıların nerede oldukları bilgisine ulaşılabilir. Bu işlem için kullanıcının harita üzerindeki koordinat verilerinin veri tabanında tutulmasına ihtiyaç vardır. Kullanıcılar buldukları konum bilgilerini mobil uygulamayı kullanarak bildirebilmektedirler.

3.5 Mezun Bilgi Sistemi Web Uygulaması

Bu bölümde gerçekleştirilen uygulamanın kullanımına yönelik bilgilendirmeler yapılmıştır. Mezun bilgi sistemi Sistem Yönetimi İşlemleri, Mezun Kullanıcı İşlemleri ve Ana sayfa olmak üzere üç farklı modül olarak incelenebilir. Sistem MVC desenine

sahip web uygulaması olarak geliştirilmiştir.

Bu bölümde geliştirilen mezun sistemindeki özelliklerin açıklaması yapılmıştır. Geliştirilen uygulamadaki sayfalar tanıtılmıştır.

3.6 Sistem Yöneticisi İşlemleri

Sistem yöneticisi işlemleri sisteme duyuru ve etkinlik ekleme, mezun iletişim bilgilerine erişme, anket tanımlama gibi işlemlerden oluşmaktadır. Her sistem yöneticisi sadece tanımlı olduğu öğrencilerin bilgilerini görebilmektedir.

Mezunlara yönelik duyurular sisteme eklenebilecektir. Sisteme eklenen duyurular facebook oturumu açmış olan kullanıcılar tarafından paylaşılabilir. Duyuruların yayın tarihi belirlenebilecek ve sadece belirtilen tarih aralığındaki duyurular sistemde aktif olarak görüntülenebilecektir. Duyuru yönetimi sayfasında ajax metotları kullanılmış ve kullanım kolaylığı sağlanmasına çalışılmıştır. Duyuru yönetimi ekranının görünümü şekilde görülmektedir. (Şekil 3.30)

Duyuru yönetimi ekranı kullanılarak yeni duyuru eklenebilir. Duyuruların yayın tarihi belirlenebilir. Duyurular sadece belirtilen tarih aralığında görünecek şekilde ayarlanabilir. Duyuru yönetim ekranında herhangi bir duyuru yayından kaldırılabilir. Ana sayfada hareketli fotoğraflar ile duyuruya eklenen resmin görüntülenmesi sağlanabilir.

Duyurular istendiğinde mezunlara eposta olarak gönderilmek istenebilir. Bilgilendirme istediğini belirten mezunlara duyurular otomatik olarak eposta olarak gönderilir.

Mezun sistemi ana sayfasında istenirse üniversite ana sayfa duyurularının görüntülediği bir alan eklenebilir. Mezunların üniversitenin tüm etkinliklerinden haberdar olması sağlanabilir.

Tüm Duyurular

[Yeni Duyuru](#)

Duyurularda aramak istediğiniz kelimeyi girebilirsiniz. [Duyurularda Ara](#)

Başlık	Yayın Tarihi	No	İşlem	Kayan Yazıda Görünme	Yayın Durumu
Mezun Bilgi Sistemi Açıldı	14.01.2014 00:00:00 ?? "	1	Düzenle Duyuru Sil	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Mezun Derneğinden Seminer	18.01.2014 00:00:00 ?? "	3	Düzenle Duyuru Sil	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Duyuru modülü çalışması bitiyor	02.04.2014 00:00:00 ?? "	4	Düzenle Duyuru Sil	<input type="checkbox"/> Kapalı	<input checked="" type="checkbox"/>
Duyur ekleme başlık	10.12.2014 00:00:00 ?? "	70	Düzenle Duyuru Sil	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Şekil 3.30. Duyuru yönetimi ekranı.

Yayınlanan duyurular sosyal medyada paylaşılabilmesi için paylaş butonları sisteme eklenmiştir. Facebook oturumunu açmış olan kullanıcılar içerikleri paylaşarak duyuruların sosyal medyada yayılmasını sağlayabilmektedir.

Sisteme etkinlikler eklenebilecek ve takvim olarak etkinlikler görüntülenmektedir. Ana sayfada görüntülenen etkinlikler sosyal medyada paylaş butonu ile paylaşılabilir.

Etkinlik yönetimi sayfası ile sistemde görünecek yeni etkinlikler oluşturulabilir. Var olan etkinlikler düzenlenebilir. Etkinliklerin başlama ve bitiş tarihleri sisteme tanımlanabilir. Tanımlanan etkinlikler bir takvim görünümü ile sunulabilir. Etkinlikler mezunlara eposta veya sms yoluyla bildirilebilir. Şekil 3.31’de görülen etkinlik yönetim ekranındaki butonlar sayfa mobil cihazlarda açıldığında sayfa boyutlarına uyum sağlamaktadır.

Tanımlanan etkinliklerle ilgili bilgiler mezunlara eposta olarak gönderilebilir. Bilgilendirme istediğini belirten kullanıcılara, etkinlik bilgileri eposta ile gönderilir. Kullanıcılar etkinliğe katılıp katılmayacaklarını işaretleyebilirler. Sistem üzerinde etkinliğe katılım bilgileri diğer mezunlarla paylaşarak katılım konusunda mezunlar

bilgilendirilebilir.

Tüm Etkinlikler

[Yeni Etkinlik](#)

Etkinliklerde aramak istediğiniz kelimeyi girebilirsiniz. [Etkinliklerde Ara](#)

No	Başlık	Yayın Tarihi	İşlem	Kayan Yazıda Görünme	Yayın Durumu
36	Sony bazı çalışanları ile yollarını ayıracak	02.05.2014 00:00:00	Düzenle Sistemden Sil	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
37	Kardeşim hafta sonu buraya geliyor.	02.08.2014 00:00:00	Düzenle Sistemden Sil	<input type="checkbox"/>	<input checked="" type="checkbox"/>
38	İş adamı olan mezunları topluyor	02.05.2014 00:00:00	Düzenle Sistemden Sil	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Şekil 3.31. Etkinlik Yönetimi ekranı.

Yayınlanan etkinliklerin facebook sisteminde paylaşılması için gerekli eklentiler sisteme dahil edilmiştir. Tanımlanan etkinliklerin takvim tasarımında görüntülenmesi sağlanmıştır.

İş ilanları sisteme girildiğinde yönetici onayı ile ana sayfada yayınlanmaktadır. Firmalara kullanıcı tanımlanarak gerekirse firmaların da iş ilanı ekleyebilmeleri sağlanmıştır. Mezunlar da iş ilanı ekleyebilmektedirler. İş ilanları uzmanlık alanlarına bağlı olarak tanımlanabilmektedir. İş ilanı veren firmanın çalıştığı sektör bilgisi sistemde tutulmaktadır.

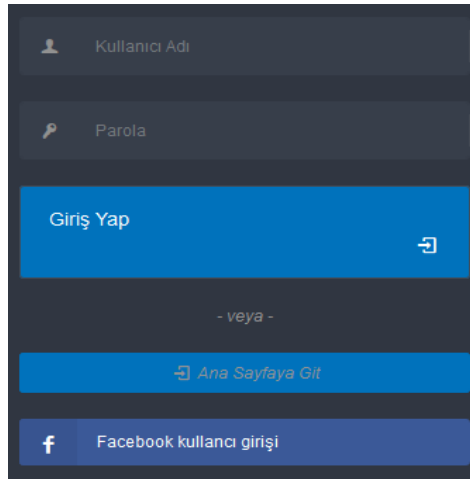
Mezunlara ait bilgiler yönetici tarafından görüntülenebilmektedir. Telefon, eposta ve adres bilgileri yönetici tarafından güncellenebilmektedir. Mezunlar bu bilgilerini kendileri de güncelleyebilmektedirler. Mezunlara ait adres bilgileri nüfus idaresinin sistemi ile öğrenci bilgi sistemine aktarılmaktadır. Aktarılan güncel veriler mezun sistemine kayıt edilmektedir.

Mezunlar istedikleri takdirde sisteme eklenene iş ilanı ve duyurulardan haberdar olmak istediklerini işaretleyebilmekte ve işaretli olan mezunlara duyuru eposta ile gönderilebilmektedir. Yönetici mezunlara toplu eposta gönderebilmektedir.

Mezun sistemi mevcut öğrenci bilgi sistemi ile uyumlu şekilde çalışmaktadır.

Sistem mezunları öğrenci bilgi sistemi üzerinden doğrulamakta ve kayıt işlemini bu şekilde gerçekleştirmektedir. Kayıt işlemi istenildiğinde facebook hesabı kullanılarak da yapılabilmektedir. Kullanıcılar ilk kayıt esnasında facebook hesaplarını, kimlik numaralarını, diploma numaralarını belirterek sisteme kayıt olabilmektedir. Facebook sistemiyle uyum konusunda ilk olarak doğrulama işlemi gerçekleştirilmiştir. Kullanıcıların facebook id bilgisi ve facebook eposta bilgisi sisteme kayıt edilmekte ve mezun sistemindeki kullanıcı bilgileri ile eşleştirilmektedir.

Mezunlar oturum açmak için öğrenci bilgi sistemindeki kullanıcı bilgilerini kullanabilirler. Öğrenci bilgi sisteminin yanında facebook hesaplarıyla da oturum açabilirler. Mezunlar sisteme ilk girişlerinde, öğrenci bilgi sistemindeki bilgileri mezun sistemine aktarırlar. Sonraki doğrulama işlemleri mezun sistemi üzerinden yapılmaktadır. Şekil 3.32 kullanıcı girişi ekran görünümünü göstermektedir.



Şekil 3.32. Oturum açma ekranı.

Kullanıcıların facebook hesaplarının doğrulanmasının ardından kimlik numaraları ve diploma numaraları istenir. Bu bilgileri de doğru giren kullanıcı sisteme kayıt olmuş olur. Şekil 3.33 facebook ile ilk kez kayıt olan mezunların bilgilerini doğrulama ekranını göstermektedir.

Sistem yöneticisi mezunlara ait istatistiklere sistem üzerinden ulaşabilmektedir. Yapılan anketlere ait verileri inceleyebilir ve grafik sonuçları görebilir. Bazı istatistikler

herkese açık olarak yayınlanmaktadır. Mezun sayıları Google Maps api kullanılarak harita üzerinde görüntülenebilmektedir.

Şekil 3.33. Facebook kullanıcı kayıt ekranı.

Mezun sisteminde anket tanımlanabilmekte, mezunlara anket uygulanabilmektedir. Sisteme tanımlanan anketler aktif olarak işaretlendiğinde mezun sayfasında mezunların önüne gelmekte ve mezunlar anketlere cevap verebilmektedirler. Anketlere ait sorular ve cevap seçenekleri yönetici tarafından tanımlanabilmektedir.

Anket yönetimi ile sisteme yeni anketler eklenebilir. Var olan anketler düzenlenebilir. Sorulara verilmesi istene cevaplar düzenlenebilir. Anketlerin yayın tarihi aralığı belirtilebilir. Anketlerin cevaplanması işlemi öğrencilerin oturum sayfasında yapılabilmektedir. Oturum açmış olan kullanıcıya sistemde mevcut olan anket soruları gösterilmektedir. Kullanıcı her cevapladığı anket sorusunun ardından sistemde olan diğer anket sorusunu görmektedir. Anket cevaplama işlemi ajax metotları kullanılarak gerçekleştirilmiş olduğundan tüm sayfa yenilenmeden hızlıca anketlerin cevaplanması sağlanmıştır. Şekil 3.34, Şekil 3.35 ve Şekil 3.36 anket tanımlanmasını sağlayan sayfaları göstermektedir.

Sistemde yöneticisi rolündeki kullanıcılar sadece sorumlu oldukları bölümlere ait öğrenciler üzerinde işlem yapabilmektedir. Kullanıcıların rollerini ve sorumlu oldukları birimleri yönetici tanımlayabilmektedir. Sistem yöneticisi sistemde tanımlı rolleri ve kullanıcıları belirleyebilmektedir. Akademik birim bazlı yetkilendirme sağlanabilmektedir.

+ Yeni Anket Ekleme			
Anket Adı	Yayın Tarihi	Actions	Yayın Durumu
Mezun çalışma durumu anketi	23.04.2014 00:00:00	+ Anket Soruları Yönetimi ✎ Düzenle 🗑 Sil	<input checked="" type="checkbox"/> ON
Mezunlara son sene anketi	25.06.2014 00:00:00	+ Anket Soruları Yönetimi ✎ Düzenle 🗑 Sil	<input type="checkbox"/> OFF

Şekil 3.34. Anket yönetimi ana ekranı.

Mezun çalışma durumu anketi anketine ait bilgiler.

Yayın Tarihi	23.04.2014 00:00:00	Yayın Durumu	<input checked="" type="checkbox"/>
✕ Yeni Soru Ekle ➡ Anket Ana Sayfasına Dön Mezun çalışma durumu anketi Anketinin soruları			
Soru Metni	İşlemler		
Alanınızla ilgili bir işte çalışıyor musunuz?	+ Cevap Ekle	✎ Düzenle	🗑 Sil
Çalıştığınız yerde bilecik üniversitesi mezunu var mı?	+ Cevap Ekle	✎ Düzenle	🗑 Sil

Şekil 3.35. Anket soruları yönetimi ekranı.

Anket sorusuna seçenek ekleyebilirsiniz

+ Cevap Ekle ✕ Geri Dön

Sorunun görünümü

Çalıştığınız yerde bilecik üniversitesi mezunu var mı?	+ Cevap Ekle	✎ Düzenle	🗑 Sil
<input checked="" type="radio"/> hayır yok	+ Cevap Ekle	✎ Düzenle	🗑 Sil
<input type="radio"/> Evet var	+ Cevap Ekle	✎ Düzenle	🗑 Sil

Şekil 3.36. Anket soruları için cevap yönetimi ekranı.

Sistemde yöneticisi rolündeki kullanıcılar sadece sorumlu oldukları bölümlere ait öğrenciler üzerinde işlem yapabilmektedir. Kullanıcıların rollerini ve sorumlu oldukları birimleri yönetici tanımlayabilmektedir. Sistem yöneticisi sistemde tanımlı rolleri ve kullanıcıları belirleyebilmektedir. Akademik birim bazlı yetkilendirme sağlanabilmektedir.

3.7 Sisteme Kayıt Olan Kullanıcılara Ait İşlemler

Bu bölümde oturum açan mezunlara yönelik olarak tasarlanan özellikler

açıklanmıştır.

Mezunlar kendi sayfalarına mesaj yazabilmekte yazdıkları mesajlar arkadaş oldukları kişiler tarafından görünmektedir. Mezunlar arası mesajlar istenildiğinde facebook ile paylaşılabilir. Mezunlar sistemde arkadaş ekleme, arkadaş arama özelliklerini kullanabilecektir. Birbirlerine arkadaşlık isteği gönderebileceklerdir. Mezunlar iletilerine istedikleri takdirde fotoğraf da ekleyebileceklerdir.

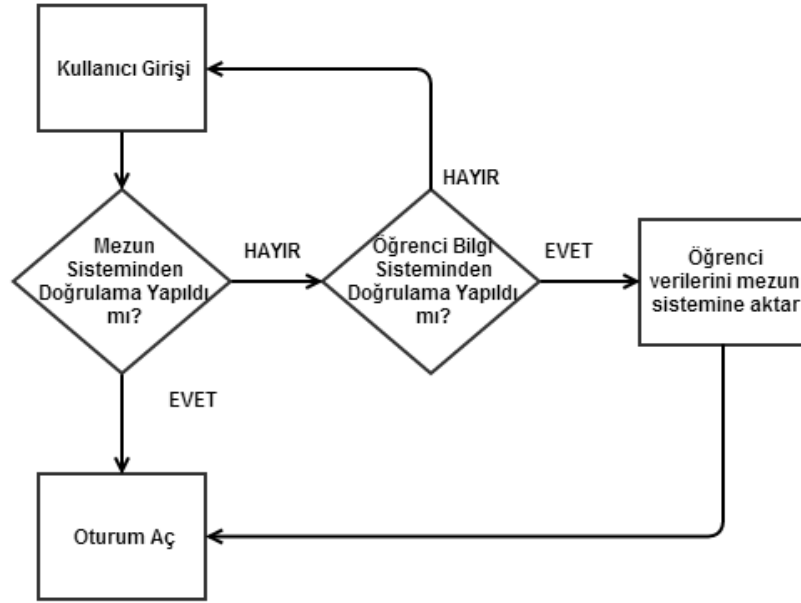
Mezunlar uzmanlık alanlarını sisteme tanımlayabilmektedir. Her bir mezun uzmanlık alanı olan konuları sisteme girebilmektedir. Mezunlar arasında uzmanlık alanlarına göre arama yapılabilir. Sisteme giriş yapan her mezun daha önce eklediği uzmanlık alanlarını sistemde görebilmektedir.

Mezunlar uzmanlık alanlarına, ad soyad bilgisine göre aranabilmektedir. Bu sayede istenen bir alanda bilgi ve tecrübe sahibi mezunlara daha kolay ulaşılabilir. Mezunlar sistemden arkadaş arama işlemi için bu arama sonuçlarını kullanabilmektedirler. Bilgilerine ulaştıkları arkadaşlarını arkadaş olarak ekleyebilmektedirler.

Kullanıcılar mevcut öğrenci bilgi sistemi parola bilgileri ile sisteme giriş yapabilmektedirler. Parolasını bilmeyen kullanıcılar mevcut facebook hesaplarını kullanarak kimlik numarası ve diploma numarası belirterek sisteme kayıt olabilmekte ve giriş yaparken facebook hesap bilgilerini kullanabilmektedirler. Sisteme giriş yapan kullanıcıların bilgileri öğrenci bilgi sisteminden mezun sistemine aktarılmaktadır.

Şekil 3.37 mezun sisteminde oturum açılması ve öğrenci verilerinin aktarılması algoritmasını göstermektedir. Sistemde ilk defa oturum açan mezunların bilgileri mezun sistemine aktarılmakta ve öğrencilerin kayıt olması sağlanmaktadır. Mezun sistemine ilk girişten sonra sistem doğrulama işlemini mezun sistemi veri tabanından yapmaktadır.

Kullanıcılar facebook sistemindeki kullanıcı bilgileri ile de oturum açabilmektedirler. Bu işlem sayesinde zaten sürekli kullanılmakta olan sosyal medya hesap bilgileri ile mezun sistemine daha kolay ulaşılabilmesi sağlanmıştır.

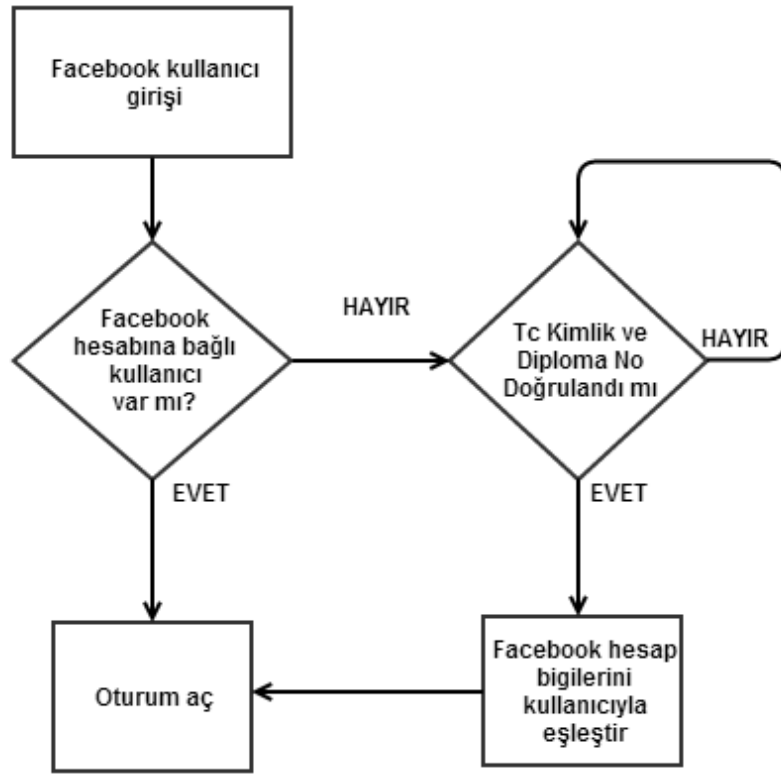


Şekil 3.37 Mezun oturum açma algoritması.

Facebook hesap bilgileriyle sisteme giriş algoritması Şekil 3.38’de görülmektedir. Şekilde görüldüğü üzere kullanıcının facebook id değeri sistemden kontrol edilmekte daha önce sisteme kayıtlı olup olmadığına bakılmaktadır. Daha önce sisteme facebook ile kayıt olmayan kullanıcılara doğrulama amaçlı bilgiler sorulmaktadır. Oturum açılmış olan facebook hesabıyla bilgileri doğrulanan kullanıcı facebook id değeri üzerinden eşleştirilmektedir.

Kullanıcı kayıt işlemleri sırasında doğrulama amaçlı birçok farklı soru sorulabilmektedir. Birçok üniversite mezuniyet tarihi ve diploma bilgilerini kullanırken bazı üniversiteler sadece kimlik numarası ile doğrulamayı tercih edebilmektedir. Doğulamada hatırlanması daha kolay olan kimlik bilgilerinin kullanılması mezunların kayıt işlemlerini kolaylaştıracaktır.

Zaman tüneli mezunların okul hayatı boyunca sistemlere bıraktıkları izlerin tarih sıralamasıyla görüntülenmesidir. İsteyen mezun bu işlemi sistem üzerindeki zaman tüneli oluşturma butonuna tıklayarak gerçekleştirebilir. Zaman tüneli oluşturulduğunda mezuna ait ders bilgileri, not bilgileri, kayıt işlemleri tarih sıralı olarak sayfasında görüntülenmektedir.



Şekil 3.38. Facebook bilgilerinin kullanıcıyla eşleştirilmesi

Mezun sistemi mobil cihazlarda kullanılacak bir tasarıma sahiptir. Sistem responsive tema özelliğinde olduğundan mobil aygıtlarla bağlanıldığında mobil tasarıma uygun şekilde görüntülenmektedir.

Mezunlar kendi özgeçmişlerini sisteme ekleyebilmektedirler. Eğitim bilgileri, iş bilgileri, çalışılan proje bilgileri alanları doldurulduğunda sistem bu verilerden yola çıkarak bir özgeçmiş hazırlayabilmektedir.

Mezunlar görüşlerini bildirmek amacıyla yorum alanını kullanabilirler. Görüşlerini bildiren mezunların görüşleri ana sayfada yayınlanmaktadır. Mezun görüşleri alanı mezun olan öğrencilerin okul ve bölüm hakkındaki kısa yorumlarıdır. Bu alanda öğrencileri başarı hikayelerini de paylaşabilirler.

Oturum açan kullanıcılar ekranın sağ tarafında aktif ankete ait soruları görüp cevaplayabilirler. Her cevaplanan anket sorusunun ardından sistemde var olan aktif

anketlerden yeni sorularla bu alan güncellenir. Şekil 3.39 mezunların anket cevaplama işlemini yaptıkları sayfa bölümünü göstermektedir. Sistemde aktif anket yoksa bu alan boş olarak görünür. Kullanıcılar cevapladıkları soru sayısını bu bölümde görebilir ve toplam kaç soru olduğu bilgisine ulaşabilirler.

Anket - 8 tane soru var. 4 soruyu cevaplandın.

Mezun bilgi sistemini nereden duydunuz?

İnternet

Arkadaşım söyledi

Bölümüm bilgiledirdi

Diğer

Cevapla

Şekil 3.39. Anket sorusu cevaplama.

Oturum açan kullanıcılar istedikleri takdirde sisteme uzmanlık alanlarını ekleyebilirler. Sisteme uzmanlık alanı ekleme ve daha önce eklenen uzmanlık alanlarını görmek için ana ekranın sağında kullanılan alan kullanılır. Şekil 3.40 kullanıcıların sisteme tanımladıkları uzmanlık alanlarının sayfada nasıl görüntülendiğini göstermektedir. Uzmanlık alanları profil bilgilerinden yeniden güncellenebilir. Bu alana girilen veriler özgeçmiş oluşturulurken kullanılabilir.

Uzmanlık konuların - 9 farklı alanın alanın var.

PHP PLC

Gömülü Sistemler

Autocat Uzmanı

Net programlama Visual basic

Gıda Hijyeni Uluslararası İktisat

vedat

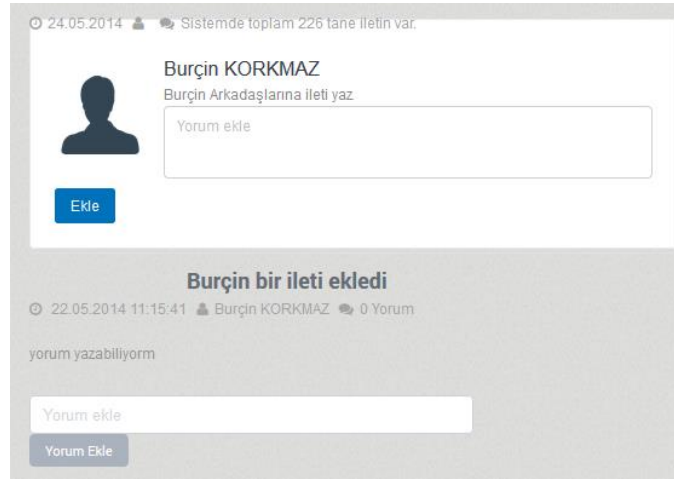
Yeni uzmanlık alanı ekle

Alan Ekle

Şekil 3.40. Kullanıcı uzmanlık alanı ekleme.

Mezunlar sisteme giriş yaptıklarında arkadaş oldukları kişilerin iletilerini ve kendi iletilerini gördükleri ekranla karşılaşılır. Bu ekranı kullanarak sisteme yeni ileti ekleyebilirler. Daha önce eklenmiş olan iletilere yorum yazabilirler. İleti ekleme işlemi ajax yöntemi kullanılarak yapılmaktadır. Bu sayede sayfa yenilenmeden hızlıca eklenen iletiler sisteme kayıt edilebilmekte ve kullanıcı dostu bir kullanım sağlanmaktadır. Şekil 3.41’de mezunların ileti ekleme işlemi yaptığı form görülmektedir.

Kullanıcılar ana ekranda sistem yöneticisi tarafından eklenmiş olan duyuru ve iş ilanlarını da görebilmektedir.



Şekil 3.41. İleti ekleme ekranı.

Mezunlar profil ayarları ekranında başkalarıyla paylaşmak istedikleri bilgilerini seçebilirler. Fotoğraf, özgeçmiş, iletiler gibi içeriklerini paylaşmak isteyip istemediklerini seçebilirler. Profil sayfasında mezun sistemindeki parolayı değiştirme, eğitim ve iş bilgisi ekleme, uzmanlık alan ekleme işlemleri de yapılabilmektedir. Mezunlar profil ayarlarında buldukları konum bilgilerini de sisteme girerek harita üzerinde tam olarak nerede olduklarını belirtebilirler. Mezunlara ait konum bilgileri Google Maps java script api kullanılarak harita üzerinde görüntülenebilmektedir.



Şekil 3.42. Mezun profil sayfasındaki gizlilik ayarları.

Mezun arama işlemi ad, soyad ve mezunların uzmanlık alanlarına göre yapılabilmektedir. Uzmanlık alanına göre arama sayesinde istenen alanda ihtisas yapmış mezunlara hızlı olarak ulaşılabilir. İş ilanları da bu uzmanlık alanlarıyla eşleştirilerek sistemdeki iş ilanlarıyla ilgilenebilecek öğrencilere bilgi gönderilmesi sağlanabilmektedir. Mezunların uzmanlık alanlarıyla ilgili iş ilanları mezun ana sayfasında görüntülenerek mezunların haberdar edilmesi sağlanabilmektedir. Şekil 3.43 mezun arama ekranını göstermektedir.



Şekil 3.43. Mezun arama ekranı.

Diploma doğrulama sayfası diploma bilgileri ile doğrulama yapmak amacıyla kullanılmaktadır. Herhangi bir kurum belirtilen bir diplomanın sahte olup olmadığını diplomadaki bilgileri girerek doğrulayabilir. Şekil 3.44 diploma sorgulama sayfasında doldurulması gereken alanları göstermektedir. Doğrulama işlemi sonucunda girilen bilgilerin doğruluğu sistemden onaylanmakta ve sayfada kullanıcıya bilgi verilmektedir. Çalışmanın ikinci aşamasında sonucun grafik olarak diploma şeklinde oluşturulması

sağlanabilir.

Diploma Sorgulama

Tc Kimlik No	<input type="text" value="Tc Kimlik No"/>
Diploma No	<input type="text" value="Diploma No"/>
Mezuniyet Yılı	<input type="text" value="Mezuniyet Yılı"/>

[Diploma Ara](#)

Şekil 3.44. Diploma sorgulama.

Mezun sistemi ana sayfasında sisteme girilen duyurular, son mezunlara ait bilgiler, iş ilanları ve etkinlik bilgileri bulunmaktadır. Sayfa her seferinde son mezunlara ait bilgileri ekrana getirmektedir. Başarı hikâyesini yazan, yorum yapan, bilgilerini görüntülenmesine izin veren mezunlara ait bilgiler ekranda listelenmektedir. Sayfa her yenilendiğinde uygun şartları taşıyan mezunların bilgileri ekranda listelenmektedir. Ana sayfada görülen içeriklerden biri olan etkinliklerin görünümü Şekil 3.45’de görülmektedir.

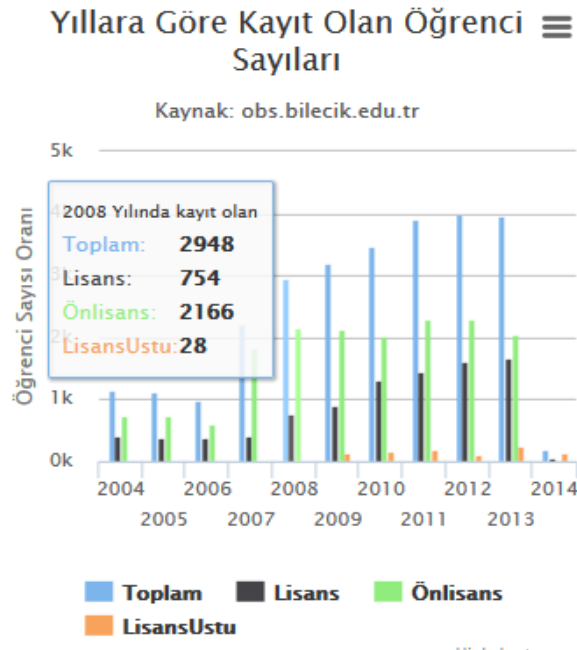


Şekil 3.45. Ana sayfada etkinliklerin görünümü.

İstatistik sayfasında öğrenci bilgi sisteminden beslenen bir takım veri grafiklerine ulaşabilmektedir. Grafikler ihtiyaca göre zenginleştirilebilmektedir.

Grafiklerin oluşturulması için HighChart isimli java script kütüphanesi kullanılmıştır. Highchart isimli api ile çok çeşitli grafikler özelleştirilerek oluşturulabilmektedir.¹ Bu kütüphaneye jquery fonksiyonları ile veri sağlanarak grafikler oluşturulmuştur. Veriler öğrenci bilgi sistemi veri tabanından ve mezun sistemi veri tabanından alınmaktadır. Şekil 3.46 yıllara göre yeni kayıt olan öğrenci sayılarını gösteren bir grafik örneğidir. Grafikte öğretim aşamasına göre yıllara göre yeni kayıt olan öğrenci sayıları gösterilmiştir.

İstatistikler ilgili birimlerin ihtiyacına göre genişletilebilir yapıdadır. Mezunlarla ilgili öğrenci bilgi sisteminden elde edilebilen verilerin yanında anketlerle elde edilen verilerin de grafikleri elde edilebilir. Nüfusa kayıtlı olunan iller, mezunların uzmanlık alanları, etkinliklerin sayısı ve katılımları gibi birçok konuda veriler grafik olarak sunulabilir.



Şekil 3.46. Kayıt olan öğrenci sayılarının gösteren grafik.

Çalışmanın ilk aşamasında temel grafikler elde edilerek kullanıma açılmıştır.

¹ <http://www.highcharts.com/>

Sisteme veri girişi sağlandıkça istatistiki grafikler çeşitlenecektir. Özellikle anketlerle ilgili bilgiler ve istatistikler ilerleyen zamanlarda önem kazanacaktır.

Mezun sisteminde mezunların nüfus kayıt bilgilerine göre harita üzerinde mezun sayıları görülebilmektedir. Yabancı uyruklu öğrencilerin de buldukları şehir ve enlem boylam bilgileri sisteme girildiğinde hangi şehirde kaç mezun öğrencimizin olduğunu harita üzerinde görülebilmektedir. Harita üzerinde mezun sayılarının görünümü Şekil 3.47 numaralı şekilde görülmektedir.

Harita üzerinde mezunların gösterilmesi konusunda ilerleyen zamanlarda mezunların konum bilgilerine göre harita oluşturulabilir. Mobil uygulama yaygınlaştıkça bu uygulamadan gelen konum bilgileriyle haritalar oluşturulabilir. Harita entegrasyonu iş ilanlarıyla ilişkilendirilebilir ve hangi bölgede kaç iş ilanı olduğu bilgisi elde edilebilir.



Şekil 3.47. Haritada mezun sayıları.

Mezun haritasında mezunların sayılarıyla orantılı olarak ilgili konumda daire şeklinde bir işaret oluşturulur. Mezun sayıları nüfus kayıt yerine göre görülebildiği gibi adres sistemine göre de oluşturulabilir. Mobil uygulamadan gelen konum bilgileri de kullanılarak mezunların hangi bölgelerde yoğunlaştıkları bilgisi harita üzerinden görülebilir.

4. MEZUN BİLGİ SİSTEMİ MOBİL UYGULAMA

Mezun bilgi sisteminin kullanım oranını arttırmak ve mezunlara daha etkin ulaşılabilmesini sağlamak üzere android uygulaması geliştirilmiştir. Geliştirilen android uygulamasında sisteme eklenen yeni duyuru ve etkinlikler görüntülenebilir. Kullanıcıların konum bilgileri uygulama ile elde edilebilir ve her bir kullanıcı konumunu bildiren diğer arkadaşlarının nerede olduğunu görebilir. Google bulut mesaj servisi sayesinde sisteme eklenen duyuruların anında kullanıcıların mobil cihazlarında uyarı ile bildirilmesi sağlanabilir. Duyur ve etkinlik gibi içerikler sisteme girildiğinde android uygulaması yüklü olan mezunlara yeni içerik bilgilendirmesini sistem uyarısı ile gösterir. Bu bölümde uygulamada kullanılan yöntemler tanıtılmış ve uygulamanın ekran görüntülerine yer verilmiştir.

Mobil uygulama geliştirilirken eclipse geliştirme ortamı tercih edilmiştir. Google firmasının tavsiye ettiği geliştirme ortamı olduğu için tercih edilmiştir. Android kaynaklarının bir çoğunda da örnekler eclipse ile anlatılmaktadır. Eclipse programı <https://www.eclipse.org/downloads/> adresinden indirilebilir. Eclipse ile android uygulaması geliştirilebilmesi için Android sdk yüklenir. Android SDK yüklemek için aşağıdaki araçların sistemde var olması gerekir.

- Eclipse 3.6.2 veya daha üst bir sürüm
- JDK 6
- Android Development Toolkit (ADT) Plugin

Eclipse kurulduktan sonra JDK 6 yüklemesi yapılır. Son olarak eclipse eklentilerinden ADT eklentisi kurulur. Bu eklenti sayesinde yazılan programlar istenen bir android sürümlü telefonda çalışıyormuş gibi modellenebilir.

4.1 Android Geliştirme Ortamının Hazırlanması

Eclipse ile Android uygulaması geliştirmek için yalnızca programı yüklemek yeterli olmamaktadır. Android SDK kurulumunun da yapılması gerekmektedir. Geliştirme yapılacak bilgisayarda ayrıca Java Development Kit (JDK) kurulu olması

gerekmektedir.

4.1.1 Java uygulama geliştirme seti kurulumu (JDK)

Geliştirme yapılacak bilgisayarda Java Development Kit (java uygulama geliştirme seti) kurulu olması gerekir. JDK kurulumu <http://www.oracle.com/technetwork/java/javase/downloads/index.html> adresinden gerekli dosyalar indirilerek gerçekleştirilebilir. Kurulum dosyası indirilirken işletim sistemine uygun olan dosyanın indirilmesine dikkat edilmesi gerekir. Şekil 4.1 İndirme sayfasının görünümünü göstermektedir.

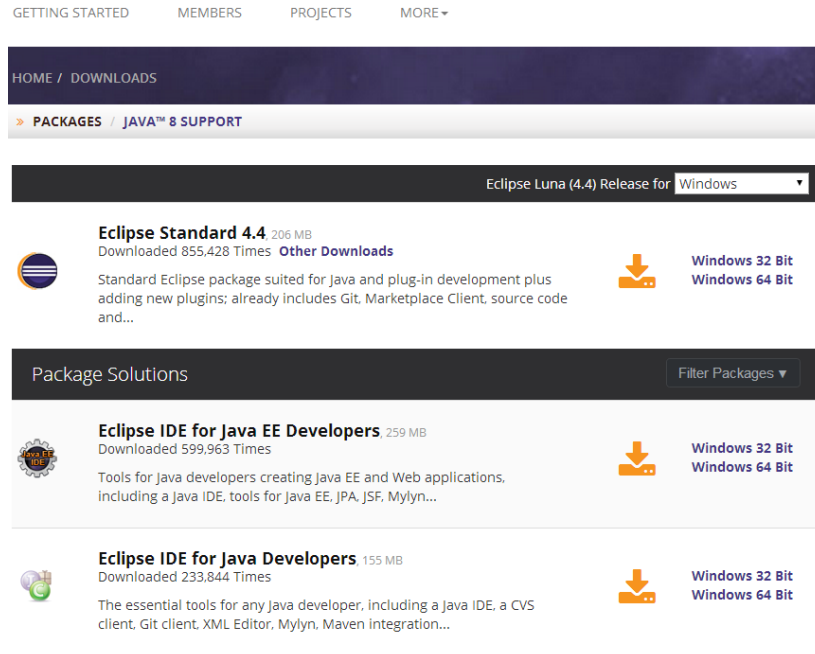


Şekil 4.1. JDK indirme sayfasının görüntüsü.

4.1.2 Eclipse kurulumu

Geliştirmede kullanılacak IDE programının kurulumu için eclipse sayfası ziyaret edilir. <http://www.eclipse.org/downloads/> bağlantısından eclipse IDE'sinin son sürümü indirilip kurulabilir. Şekil 4.2 eclipse indirme sayfasını göstermektedir. Sayfadaki en son sürümü işletim sistemine uygun olarak indirilmesi gerekir. İndirilen sıkıştırılmış dosyalar açıldığında eclipse ide ortamı kullanılmaya hazırdır. Eclipse.exe isimli dosya çalıştırılarak IDE ortamı açılır. Eclipse program dosyaları da hazırlandıktan sonra sıra

Android Development Kit yüklenesine gelmiştir. Android SDK yüklenerek geliştirme ortamı hazırlıkları tamamlanır.



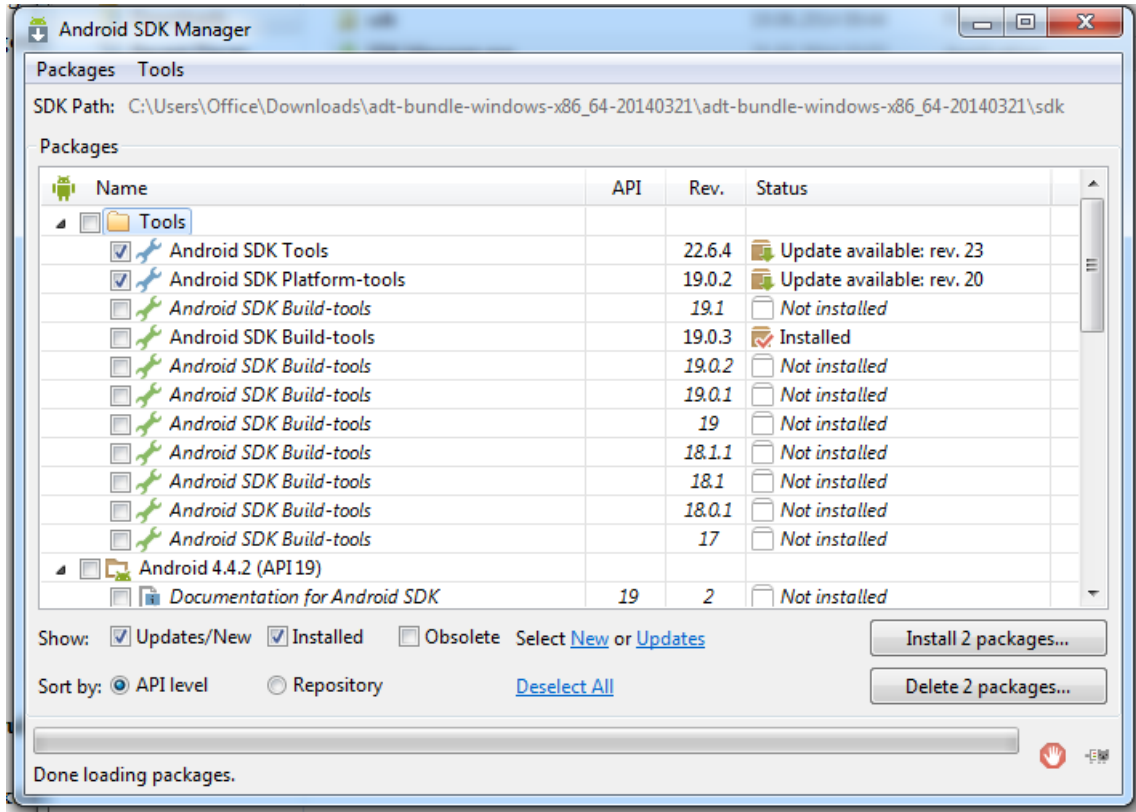
Şekil 4.2. Eclipse indirme sayfası.

4.1.3 Android uygulama geliştirme seti (Android SDK) yükleme

Eclipse ile Android uygulaması geliştirebilmek için Android SDK kurulumuna ihtiyaç vardır. Android SDK <http://developer.android.com/sdk/index.html> adresinden indirilebilir. İndirme işleminde işletim sistemine uygun olan sürümün seçilmesine dikkat edilmelidir. Ayrıca android SDK indirilirken farklı android sürümlerine ait SDK dosyaları sayfada görülmektedir. İndirme işlemi tamamlandığında sıkıştırılmış dosyalar bir klasöre çıkartılır. Çıkan dosyalar arasında SDK Manager.exe isimli dosya çalıştırılarak istenen android sürümüne ait SDK kurulumu yapılabilir.

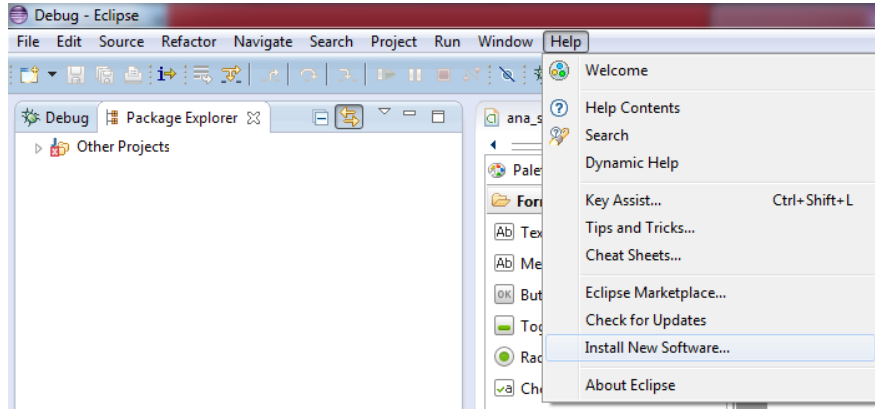
Şekil 4.3 SDM Manager yükleme ekranını göstermektedir. Şekilde görüldüğü gibi birçok farklı sürümde Android SDK yüklemek mümkündür. Android SDK Tools ve son sürüm android seçilerek kurulum yapılabilir. Geliştirilen uygulama en az hangi sürümü destekleyecekse o sürümün seçilerek yüklenmesi sağlanmalıdır. Birden fazla farklı sürüm yüklenebilir. Yükleme işleminden sonra geliştirme yapılırken uygulamanın hangi

sürüm android ile çalışacağı da seçilebilir. Tüm android sürümleri seçilerek yükleme yapılırsa yükleme işlemi uzun sürecektir.



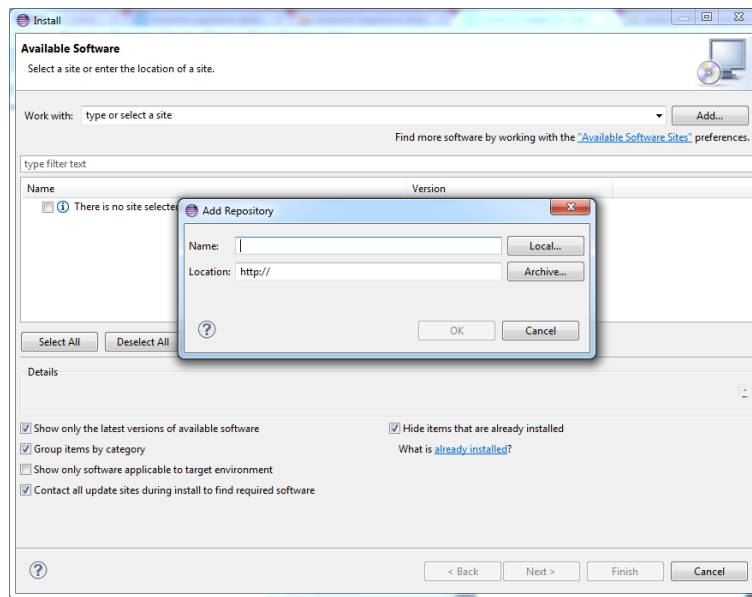
Şekil 4.3. SDK Manager yükleme ekranı.

Android SDK yüklendikten sonra Eclipse programında yapılması gereken işlemler vardır. Eclipse için Android Development Tools eklentisinin yüklenmesi gerekmektedir. Eclipse Help menüsünden Install New Software seçilerek yükleme sayfası açılır. (Şekil 4.4)



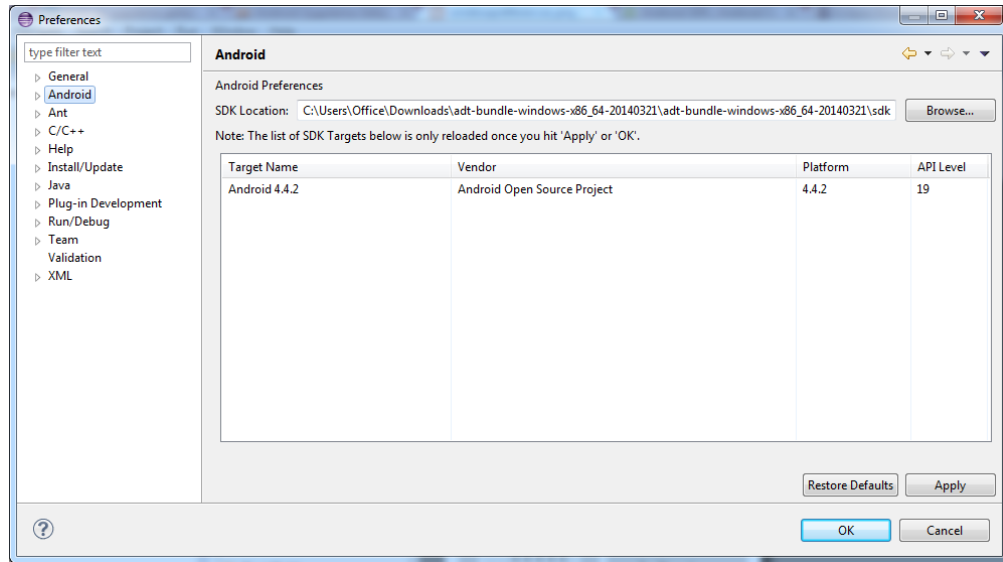
Şekil 4.4 Eclipse yardım menüsünden yeni eklenti yükleme.

Açılan pencerede sağ tarafta bulunan Add butonuna tıklanarak eklentinin yükleneceği adres seçilir. Eklenti webten yüklenecektir. Bu yüzden Android Development Tools eklentisini adresinin belirtilmesi gerekir. Penceredeki Name alanına “ADT”, location alanına da <http://dl-ssl.google.com/android/eclipse/> adresi yazılır. Bu işlemin ardından Ok butonuna basılarak pencere kapatıldığında büyük penceredeki alt kısma Developer Tools yazan bir seçenek gelir. Bu seçenek işaretlenir. İleri butonuna tıklanır, sözleşmeyi kabul ediyorum seçeneği işaretlenir ve kurulum tamamlanmış olur.(Şekil 4.5)



Şekil 4.5. Android Development Tools eklentisi yükleme.

Bu işlemlerin ardından ADT (Android Development Tools) ayarları yapılır. Bu işlem için eclipse programında window menüsünden Preferences seçeneği seçilir. Gelen pencerede Browse butonuna tıklanarak daha önce indirilmiş olan Android SDK dosyalarının çıkartıldığı klasöre gidilir. Eğer doğru dosya seçildiyse pencerede SDK yükleme yaparken seçilen sürümlerin bir listesi görülür. (Şekil 4.6) Listede yükleme yapılan SDK sürümleri görülüyorsa işlemler doğru şekilde yapılmış demektir. Artık Eclipse ile Android uygulaması geliştirmek için gerekli tüm hazırlıklar tamamlanmış olur. Bu işlemlerden sonra android projesi başlatılabilir.



Şekil 4.6. Android Development Tools ayarları.

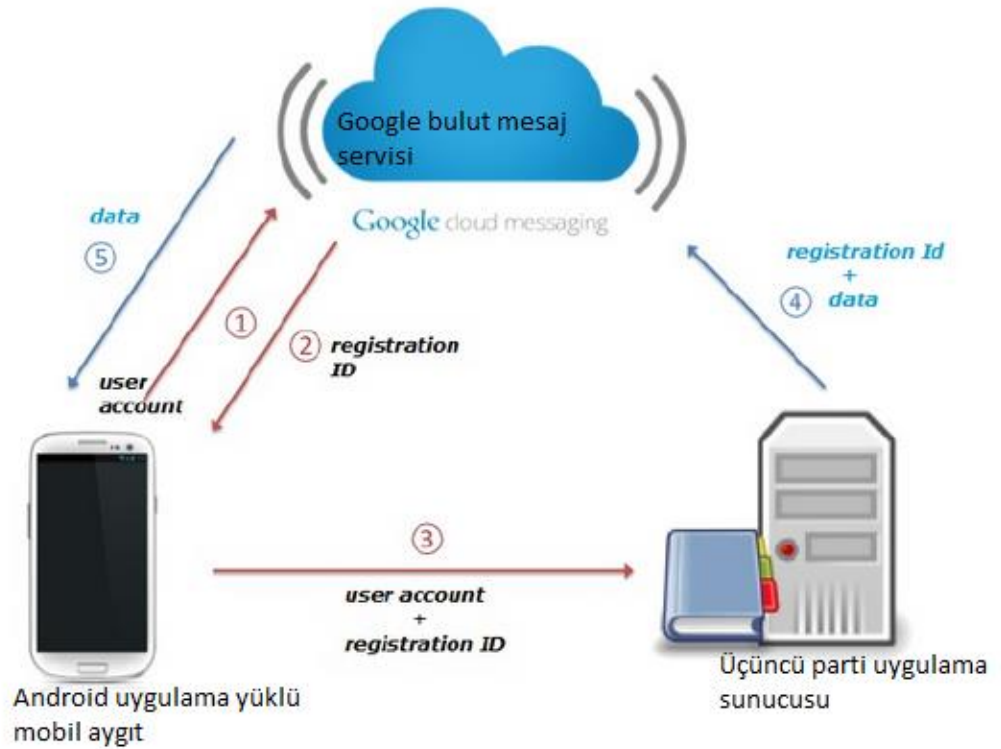
4.2 Google Bulut Mesaj Servisi

Mobil cihazlarla merkezi sunucu arasında sürekli açık kalan bir bağlantı üzerinden haberleşme sağlanması Push Notification olarak isimlendirilir. İOS ve Android cihazlarda kullanılan bu hizmet uygulama sahibi tarafından kullanıcılara çeşitli mesajlar gönderilmesini sağlar. Bu sayede uygulama sürekli canlı tutularak kullanım oranı artırılabilir ve kullanıcılar istenen verilerden anında haberdar edilebilir. (Derinözlü, 2013)

Android cihazlarda bulunan bu bildirim sistemi Google Bulut Mesajlaşma Servisi (Google Cloud Messaging Service) olarak adlandırılır. Servisin kullanılabilmesi için Google hesabıyla üye olunması gerekmektedir. Bulut mesaj servisi sayesinde

uygulama kapalıyken bile mobil cihazlara bildirimler gönderilebilir.

Bulut mesaj servisi sayesinde üçüncü parti uygulamalar android cihazlara mesaj gönderebilmektedir. Bu servis sayesinde android uygulamasının sürekli mesaj dinlemesine gerek yoktur. Yeni mesaj geldiğinde sistem android cihazlara yeni mesaj yayını yapmaktadır. Google bulut mesaj servisinin çalışmasını gösteren örnek uygulama şekilde görülmektedir. (Şekil 4.7)¹ (Derinözlü, 2013)

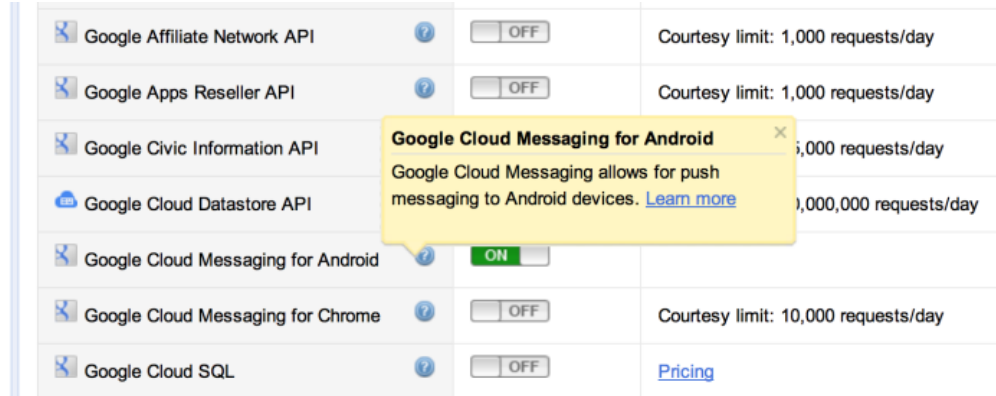


Şekil 4.7. Google bulut mesaj servisinin çalışması (Derinözlü, 2013).

Bulut mesaj servisini kullanabilmek için bir Google api hesabı oluşturulması gerekmektedir. Oluşturulan hesap üzerinden Android cihazlar için mesajlaşma servisinin başlatılması gerekir. Bu işlem için APIs menüsünden ilgili alanını işaretlenmesi gerekir. Şekil 4.8 bu işlemin ekran görüntüsünü göstermektedir. Şekilde (Şekil 4.7) görülen

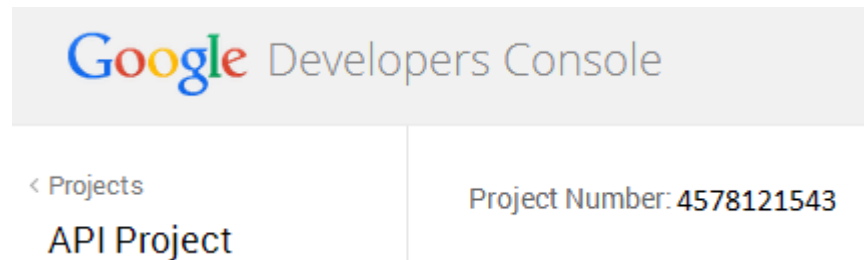
¹ <http://www.academia.edu>

üçüncü parti uygulama sunucusu kendi geliştirmiş olduğumuz uygulamanın çalıştığı sunucudur. Bu sunucu kullanılarak istenen cihazlara mesaj gönderilebilmektedir. Sunucu aynı zamanda Google bulut mesaj servisine kayıt olmuş cihazların id bilgisinin de kayıt edildiği veri tabanını barındırır. Google mesaj servisine mesaj gönderme işlemi bu id değerleri kullanılarak yapılır.



Şekil 4.8. Google API ayarlarından Android cihazlar için mesaj servisinin açılması.

Mesaj servisi açıldıktan sonra Google sistemindeki proje numarası kullanılarak mesajlaşma işlemleri yönetilmektedir. Proje numarası API konsolunun ilk ekranında görülebilir. (Şekil 4.9)



Şekil 4.9 Google API proje numarası.

Google bildirim servisi mobil cihazların ürettikleri anahtarların bir yerde saklanmasını gerektirir. Uygulama bir mobil cihaza yüklendiğinde önce Google sunucularıyla haberleşir ve itme (push) servisini kullanmak için bir anahtar değeri alır. Uygulamaya bir uyarı gönderilmek istendiğinde bu anahtar değere ihtiyaç duyulur. Bu anahtar kullanılarak cihaza özel mesajlar gönderilebilir. <https://code.google.com/p/gcm/> adresinde cihazlara ait anahtarların saklanmasını üstlenen bir servis ve bir örnek

uygulama bulunmaktadır.

4.2.1 Google bulut mesaj servisindeki bileşenler

Bu başlık altında bulut mesaj servisinin çalışması için gerekli bileşenler kısaca tanıtılacaktır. (Derinözlü, 2013)

Mobil Cihaz: GCM kullanan android uygulamasının yüklü olduğu cihazdır. Mobil Cihaz üzerinde en az android 2.2 versiyonu android işletim sistemi kurulu olmalıdır. Tasarlanan android uygulamasının kurulu olduğu cihaz uygulama çalıştırıldığında Google sunucularına kayıt olur ve benzersiz bir id değeri alır. Bu id değeri üçüncü parti sunucuda saklanır.

3rd Parti Uygulama Sunucusu: Uygulama sunucusu GCM sunucu vasıtasıyla android uygulamasına mesaj gönderir. Üçüncü parti sunucu Google mesaj servisine mesaj gönderen uygulamayı barındırır.

GCM Sunucular:Üçüncü Parti uygulama sunuculardan gelen mesajları alıp ilgili mobil cihaza ileten Google sunucularıdır.

Tanımlama Bilgileri: Google Bulut Mesajlaşma Sistemi (GCM) mimarisinde mobil cihazlar arasında mesaj gönderebilmesi için gcm mimarisini oluşturan birimler arasında kullanılan bilgilerdir.

Sender ID: Google API console üzerinden oluşturulan proje numarasıdır. Android uygulamasının gcm sunucuya kayıt edilmesinde kullanılır.

Application ID: Android uygulamasının gcm sunucuya kayıt edilmesinde kullanılır. Android uygulaması, gcm sunucusuna kayıt olabilmek için “applicationid” ve “sender id”yi gcm sunucularına gönderirler.

RegistrationID: GCM sunucular tarafından Android uygulamasına atanan id ‘dir. Her uygulama için Google Bulut Mesaj Servisi sunucuları bir benzersiz bir id belirler. Bu id değeri daha sonra uygulamaya bildirim gönderileceği zaman kullanılabilir. Uygulama Google sunucularına kayıt olduğunda bu id değerinin bir

sunucuda saklanması gerekir.

Google Kullanıcı Hesabı: GCM çalışması için android 4.0.4 den daha düşük versiyonlu mobil cihazlar için en az bir Google hesabı tanımlanmalıdır.

Sender Auth Token: Uygulama sunucusu üzerinde kayıtlı olarak tutulan API key uygulama sunucusunun Google servislerine erişimde kullanılır. Google mesaj servisine yeni bir mesaj gönderilmesini sağlayan sunucu bu API key ile sunucuyla iletişime geçer.

Android uygulamasında bildirim servisinin aktif hale getirilmesi için AndroidManifest dosyasında bir takım ayarların yapılması gerekir. Bulut mesaj servisi Google hesabı gerektirdiğinden **permission.GET_ACCOUNTS** iznini ihtiyaç duyulur. Cihazın Google sunucularıyla bağlantıda olması için C2D izinleri gereklidir. **Receiver** etiketiyle Google sunucusundan uygulamaya özel gelen mesajları ayıklarız. **GCMIntentService** bir uyarı geldiğinde harekete geçen arka plan servisini tanımlar.

Bir uyarı geldiğinde harekete geçecek bir servis tanımlamak için **GCMBaseIntetService** sınıfından türetilmiş bir sınıf tanımlanır. Bu sınıfa ait dört adet metodun bu tanımlanan sınıfta uyarlanması gerekir. Şimdi bu metodların ne işe yaradıklarını inceleyelim.¹

onError : Bulut mesaj servisi ile bağlantıda ya da kayıt sırasında bir hata oluşursa devreye girer.

onMessage : Servis üzerinden bir mesaj geldiğinde harekete geçer. Uygulamaya bir mesaj geldiğinde çalışacak kod burada tanımlanır.

onRegistered : Uygulama sunucuyla kayıt işlemi gerçekleştirdiğinde tetiklenir. Metodun içerisindeki registrationId değişkeni sunucu tarafından cihaza özel üretilen bir değerdir ve bir sunucuda saklanır. Daha sonra cihaza mesaj atılmak istendiğinde bu

¹ <https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-401/google-bulut-mesaj-servisi>

anahtarla istek yapılır.

onUnregistered : Mesaj servisi ile bağlantı koparsa (örneğin kullanıcı izinleri kaldırır) bu metod harekete geçer. Burada registeredId değeri artık geçersiz olduğundan saklandığı yerden silinir.

4.2.2 Bulut mesaj servisinde işlem adımları

Bulut mesaj sunucusuna kayıt işlemi: Android uygulamasının mesaj alabilmesi için Google bulut mesaj servisi sunucularına kayıt olunur. Kayıt olmak için uygulama SenderId ve ApplicationId olarak iki parametreye ihtiyaç duyar. SenderId ve ApplicationId parametreleri tanımlanan proje bilgilerinden görülebilir. Android uygulaması Google mesaj servisine kayıt olduktan sonra bu servisten almış olduğu registrationID değerini üçüncü parti uygulama sunucusuna kayıt eder. (Derinözlü, 2013)

Mesaj gönderme işlemi: Uygulama sunucusu android cihazdan gelen registrationID değerini veri tabanına kaydeder. Sunucu göndermek istediği mesajla birlikte registrationID değerini Google mesaj sunucusuna gönderir. Google mesaj sunucusu gelen registrationID değerine göre ilgili mobil cihaza mesaj gönderir. Cihaz aktif değilse mesajı sıraya koyar aktif olduğunda ilgili cihaza mesajı iletir.

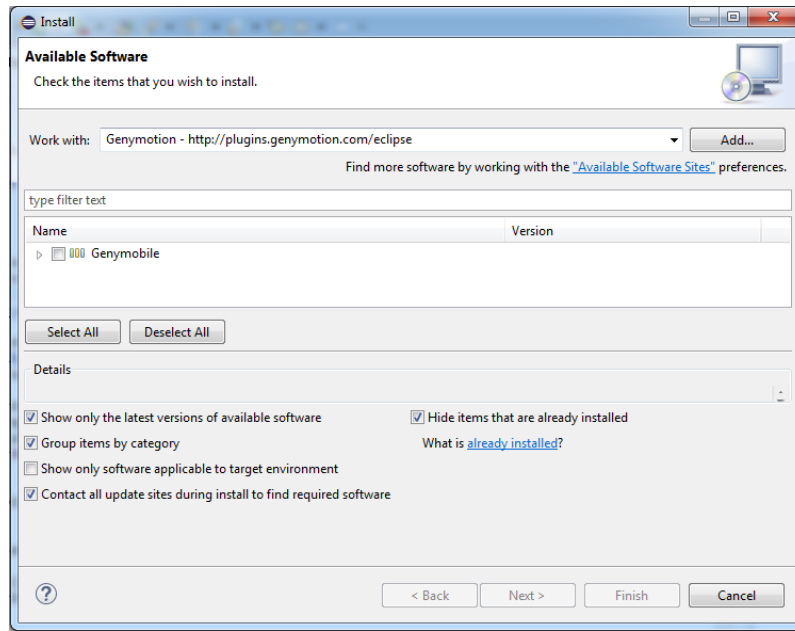
Mesajın alınması: Android uygulama Google mesaj servisinden gelen mesajı alır. GCM (Google bulut mesaj servisi) tabanlı mobil uygulama sadece yeni mesaj geldiğinde aktif olacağından çok fazla pil tüketimi olmadan etkin bir mesajlaşma sağlanabilir.

Google mesaj servisi kullanan uygulamada mobil cihaz olarak Android 4.2 sürümüne sahip cihaz kullanılmıştır. Uygulama sunucusu olarak mezun bilgi sisteminin bulunduğu sunucu ve mezun sistemi veri tabanı kullanılmıştır. Uygulamaların Google sunucularından aldıkları id değeri mezun veri tabanına kayıt edilmiştir.

GCM (Google Bulut Mesaj Servisi) servisine kayıt işlemi test etmek için emülatörde Google Play servisinin kurulu olması gerekmektedir. Eclipse ile kullanılan

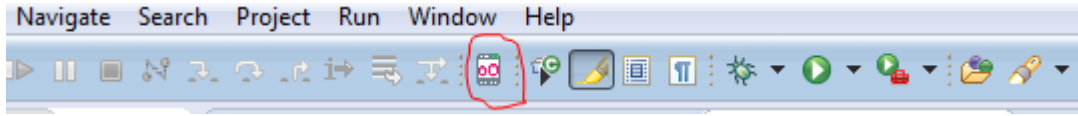
AVD manager emülatöründe Google Play servisi bulunmamaktadır. Bu sebeple Google Play servisinin kurulabildiği Genymotion emülatörü test işlemlerinde kullanılmıştır. Genymotion emülatörü varsayılan emülatöre göre çok daha hızlı çalışmakta ve Google Play servisinin kurulmasına izin vermektedir. Genymotion virtualbox üzerinden çalıştığı için normal emülatöre göre çok daha hızlı çalışmaktadır.

Genymotion emülatörünü kurmak için üreticinin sitesinden kurulum dosyaları indirilebilir. (<http://www.genymotion.com/>). Dosyaları indirmek için sisteme kullanıcı kaydı yapmak gerekmektedir. Kurulum işleminden sonra eclipse aracında bu emülatörü kullanacak bir eklenti yüklenmesi gerekmektedir. Eclipse aracının “Help” menüsünden “Install New Software” seçeneği işaretlenerek yeni eklenti yükleme ekranı açılır. Bu ekranda eklenti adresi alanına <http://plugins.genymotion.com/eclipse> adresi yazılarak genymotion emülatörünün kullanılmasını sağlayan eklenti yüklenebilir (Şekil 4.10).



Şekil 4.10. Genymotion eklentisinin eclipse aracına kurulması.

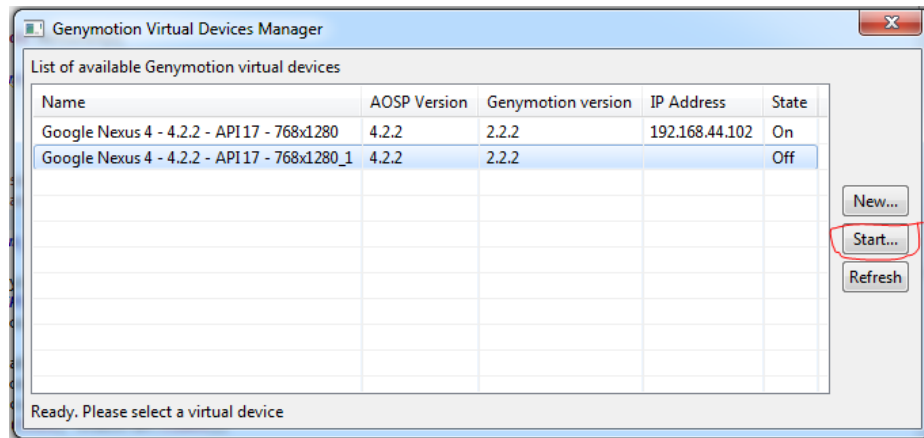
Eklenti yüklendiğinde eclipse menüsüne genymotion emülatörünün açılmasını sağlayan bir menü gelir. Bu menü kullanılarak genymotion emülatörü açılabilir (Şekil 4.11).



Şekil 4.11. Genymotion emülatör simgesinin eclipse menüsünde görünümü.

Emülatör simgesi tıklandığında daha önce sdk olarak kurulmuş olan android sürümlerinin bir listesi sunulur. Bu liste çalıştırılmak istenen android sürümü seçilerek emülatör çalıştırılır (Şekil 4.12).

Genymotion emülatörü ilk kurulduğunda Google Play servisi yüklü gelmemektedir. Bu servisi yükleyebilmek için başlatılan android sürümüne uygun Google Play uygulamasının emülatöre kurulması gerekmektedir. Bu işlem için indirilen uygulama dosyalarının Genymotion emülatörünün üzerine sürüklenip bırakılması yeterli olmaktadır. Emülatör Google Play uygulamasını kurmakta ve gerekli güncellemeler varsa internet üzerinden güncellemeleri yapmaktadır.



Şekil 4.12. Eclipse ile Genymotion emülatörünün başlatılması.

4.3 Konum Bilgisinin Elde Edilmesi

Android cihazlarda konum servisleri kullanılarak anlık olarak cihazın bulunduğu konum bilgisi elde edilebilir. Konum servisleri uydular yoluyla, telefon şebekesi yoluyla konum bilgilerini elde edebilmektedir. Bu bölümde konum servislerinin nasıl kullanılabilirliği hakkında bilgi verilecektir.

Konum servisleri üç farklı şekilde konum bilgisi elde edebilir. Bunlar GPS

(Global Positionin System), Network (telefon şebekesi), passive (pasif konum bilgisi) olarak adlandırılır.

GPS (Global Position System): Eđer cihazda GPS modülü varsa, mobil cihaz uydudan gelen bilgilerle konum bilgisini tespit edebilir. Android işletim sistemi uygulamadan konum talebi geldiğinde GPS'i devreye sokarak kullanıcının konumunu uygulamaya bildirir. GPS yöntemi kapalı alanlarda çalışmamaktadır. GPS yöntemi ile konum bilgisi elde edilmesi için mobil cihazın uyduları mutlaka görmesi gerekir. Uydu ile bağlantı varsa yüksek hassasiyette konum bilgisi GPS yöntemiyle elde edilebilir.

Network: Mobil cihazda aktif bir telefon bağlantısı varsa bu bağlantı üzerinden yaklaşık konum bilgisi elde edilebilir. GPS yöntemine göre kapalı alanlarda da konum bilgisi elde edebilir ancak hassasiyeti GPS kadar etkin değildir. Bu yöntemde baz istasyonu bilgisi merkezi bir veri tabanına gönderilerek bu istasyonların konum bilgilerinden kullanıcıların yeri tahmin edilir. Bu yöntemde baz istasyonu sayısının yoğun olduğu yerlerde biraz daha yüksek hassasiyet ile konum belirlenebilirken baz istasyonu sayısının seyrek olduğu kırsal kesimlerde sonuçların yeterliliği düşmektedir. Bu yöntem aynı zamanda bir internet bağlantısı gerektirir.

Passive: Bu yöntemde bir sensör olmamasına rağmen işletim sisteminin bellekte tuttuğu geçmiş konum bilgilerinden yararlanılarak kullanıcının konumuyla ilgili bilgi verilmesidir. Bu yöntemde elde edilen sonuçlar genel olarak tahminidir ve eski konum bilgilerini verir. Uygulama gerçek olarak kullanıcının anlık konum bilgisine ihtiyaç duyuyorsa bu yöntem tercih edilmez.

Bir android uygulamasının konum bilgilerini kullanabilmesi için öncelikli olarak androidManifest dosyasında bazı izinlerin ayarlanması gerekir. Bu dosya uygulamanın erişmek istediği servislerin bilgisinin verildiği dosyadır. AndroidManifest dosyasında açılması gereken izinler şekilde (Şekil 4.13) görülmektedir. Burada **ACCESS_FINE_LOCATION** değeri GPS gibi servislerden kesin konum bilgisi alınmasına yarar. **ACCESS_COARSE_LOCATION** ise GSM ya da WiFi kaynaklı konum bilgisi alınmasına yarar. İlçe ve şehir bilgisi bu seviyede bir izinle alınabilir.

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

Şekil 4.13. AnroidManifest dosyasında konum servisi izinlerinin açılması.

Cihazda konum bilgisinin elde edilmesi için bir LocationManager yazılır. Ayrıca konum bilgilerini anlık olarak elde etmek için LocationListener hazırlanır. LocationListener sınıfının metotları ezilerek bu metotlara konum bilgisini okumak için gerekli kodlar yazılır.

GPS bilgilerini alabilmemiz önce bir konum yöneticisi oluşturmalıyız. LocationManager sınıfı tipinde bir nesne tanımlayarak bu işlem yapılır. Bu tanımlama şu şekilde yapılır:

```
LocationManager konumYoneticisi = (LocationManager) getSystemService
(Context.LOCATION_SERVICE);
```

Tanımlanan bu sınıf üzerinden Android sisteminin konum sağlayıcı servislerine erişim sağlanır.

Konum dinleyici sınıfı Android konum servislerini dinlemeyi sağlar. Bu sınıf Android'in **LocationListener** arayüzündeki metodları kullanır. Bu işlem için ezilmesi gereken dört adet metot mevcuttur. Bu metotlar kullanılarak konum servisleri dinlenebilir.¹

onLocationChanged: Bir konum güncellemesi geldiğinde (örn. cihazın hareketi sonucu güncellenebilir) bu metot uyarılır ve güncel konum location parametresiyle istenen değişkene atılır.

onStatusChanged: Konum bilgisi veren servisin durumu değiştiğinde bu metot uyarılır. Eğer konum alınamıyorsa (örn. GPS sinyali kaybolduysa) bu metot çağırılarak

¹ <https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-301/androidde-konum-servisleri>

İlgili servisin (provider) durumu status değişkeniyle bildirilir. Buradaki değerler OUT_OF_SERVICE (hizmet dışı), TEMPORARILY_UNAVAILABLE (geçici olarak hizmet dışı) ve AVAILABLE (çalışıyor) şeklindedir. Bu değerler LocationManager içinde statik olarak saklanır.

onProviderEnabled: İlgili konum servisi kullanıcı tarafından etkinleştirilirse çalışır (örn. kullanıcının elle GPS'i açması). Konum servisi etkinleştirildiğinde işlem yapılmak isteniyorsa bu metot kullanılır.

onProviderDisabled: İlgili konum servisinin kullanıcı tarafından kapatıldığını belirtir. Konum servisi kapatıldığında yapılmak istenen işlemler bu metot içinde tanımlanır.

Konum servisi kullanılarak bir takım kriterlere göre konum güncellenmesi sağlanabilir. Örneğin kullanıcının 1km yer değiştirmesi varsa konum güncelleme veya zamana bağlı olarak konum güncelleme işlemleri yapılabilir.

4.4 Android Uygulamasıyla Web Servisinden Veri Okuma

Android uygulamaları web servislerini kullanarak veri okuyabilir. Xml veya json formatında veri sağlayan servislere istekte bulunarak veriler üzerinde işlemler yapılmasını sağlayabilir. Web servis işlemleri asenkron olarak tanımlanmış bir metot ile yapılır. Asenkron istekler sayesinde ara yüz kilitlenmeden web servisine istekte bulunulabilir. Android sisteminin web servisini kullanımını gösteren mimari şekilde (Şekil 4.14) görülmektedir.

Mezun sistemi android uygulamasında veri tabanındaki veriler json formatında sunulmaktadır. Android uygulamasının kullanacağı metotlar için bir controller sınıfı hazırlanmıştır. Bu sınıf sayesinde android cihazlardan gelen veriler veri tabanına yazılabilir veya veri tabanından istenen veriler okunabilir. Servis ile kullanıma açılan ve kullanıcı adı ve parola gerektiren veriler için android uygulamasıyla istek yapılırken kullanıcı bilgileri de gönderilir. Kullanıcı bilgilerinin şifrelenerek gönderilmesi ve servislerin SSL kullanılarak sağlanması güvenliği artırıcı önlemlerdir.



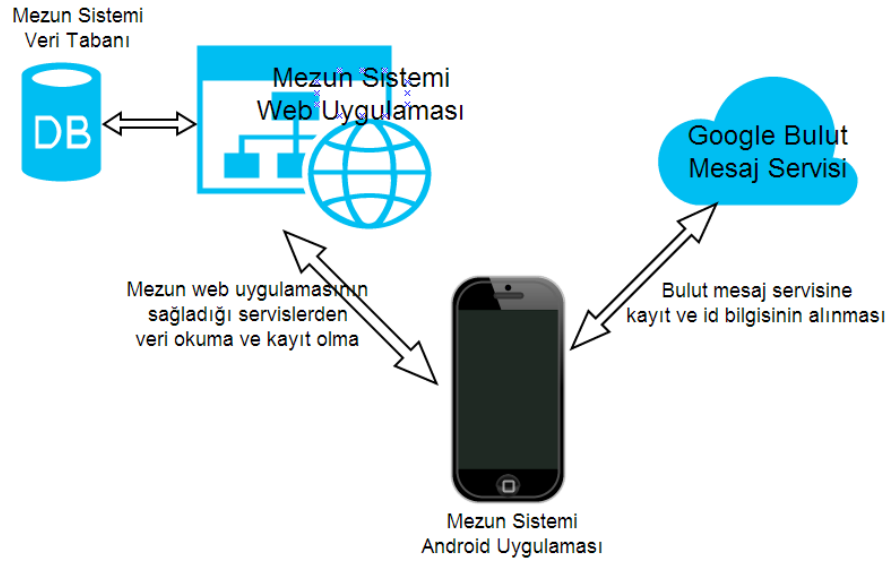
Şekil 4.14. Android uygulamasının web servisi ile haberleşmesi

4.5 Geliştirilen Android Uygulaması

Uygulamada konum servisinin kullanımı kullanıcının isteğine bırakılmıştır. Kullanıcı isterse konum bilgisini mezun sistemiyle paylaşabilir. Google bulut mesaj servisinin kullanımı ilk aşamada sisteme eklenen duyuru, etkinlik ve iş ilanlarının duyurulması amacıyla kullanılmıştır. İlerleyen sürümlerde mesajların bildirilmesi, sistem hareketlerinin bildirilmesi gibi durumlar da bulut mesaj servisi ile entegre edilebilir.

Uygulama geliştirilirken Eclipse yazılım geliştirme ortamı tercih edildi. Eclipse sayesinde android uygulamaları seçilen bir android sürümlü telefon üzerinde modellenilebilir.

Hazırlanan android uygulaması mezun sisteminin sağladığı servislerle ve Google Bulut Mesaj servisi ile haberleşmektedir. Sistemin genel mimari şeması Şekil 4.15 numaralı şekilde görülmektedir.



Şekil 4.15. Android uygulamasının haberleştiği servisler.

Android uygulaması sistemle haberleşmek için mezun sistemi uygulamasında yazılmış olan servisleri kullanır. Uygulama web servislerinden veri okur veya servisleri veri göndererek kayıt işlemlerini yapar. Servislerin kullanıcının özel bilgilerini gerektirdiği durumlarda kullanıcı bilgileri şifrelenerek servise gönderilir. Servis gelen kullanıcı bilgisini çözerek ilgili kullanıcı için işlem yapabilir.

4.5.1 Android uygulaması ana ekranı

Android uygulamasında ana ekranında kullanıcıların karşısına bir menü gelir. Menüden istenen buton seçilerek duyurulara, iş ilanlarına, etkinliklere, arkadaş listesine erişilebilir. Konum bildirme butonuyla istedikleri takdirde konum bilgisini paylaşabilirler. Konum bilgisini paylaşımı ile mezunlar arkadaşlarının konum bilgisini sistem üzerinden görebilirler.

Şekil 4.16 android uygulaması ana ekranını göstermektedir. Ana ekran mümkün olduğu kadar sade olarak içeriklere erişilmesini hedeflemektedir. Uygulama kurulduğunda Google Bulut Mesaj Servisine (GCM) cihaz kayıt olmakta ve kayıt bilgileri web uygulamasında hazırlanan metot yardımıyla mezun veri tabanına kayıt edilmektedir.



Şekil 4.16. Android uygulaması ana ekranı.

GCM kayıt ve mesaj okum işlemleri için hazırlanan sınıfın Şekil 4.17 numaralı şekilde görülmektedir. GCM işlemlerini yönetmek amacıyla hazırlanan sınıf GCMBaseIntentService isimli sınıftan türetilerek oluşturulmuştur. Bu sınıftan türetme yoluyla gelen metotlar aşırı yüklenerek mesaj alma ve kayıt işlemlerini yöneten metotlar oluşturulmuştur. GCM sistemine kayıt olduktan sonra sistemin verdiği id bilgisi mezun veri tabanına kayıt edilmektedir. Bu amaçla asenkron çalışan bir sınıf oluşturulmuştur. Google sunucularına kayıt olduktan sonra bu asenkron sınıf yoluyla mezun sistemi veri tabanına id bilgisi kayıt edilmektedir. Oturum açmamış kullanıcılar için sadece id bilgisi kayıt edilmektedir. Oturum açmış olan kullanıcılar için kayıt işlemi kullanıcı adı ve id bilgisi eşleştirilerek yapılmaktadır.

Uygulamanın eclipse ile entegre çalışan Genymotion emülatörü ile test edilmiştir. Bu emülatörün tercih edilmesinin sebebi hızlı oluşu ve GCM servislerine kayıt olmak için gerekli olan Google Play servislerinin bu emülatör üzerinde kurulabilmesidir.

```

package com.example.mezunsistemi;

import java.io.IOException;

public class GCMIntentService extends GCMBaseIntentService
{
    public GCMIntentService() {}

    protected void onError(Context arg0, String errorId) {}

    protected void onMessage(Context arg0, Intent intent) {}

    protected void onRegistered(Context context, String regId) {}

    protected void onUnregistered(Context context, String regId) {}
}

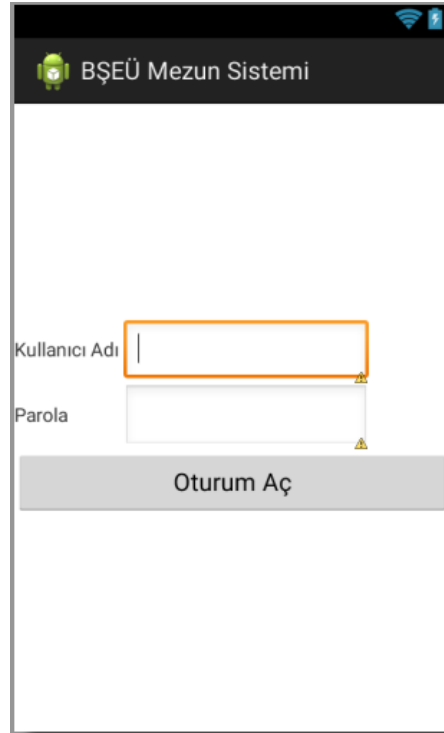
```

Şekil 4.17. GCM kayıt işlemlerini yönetmek amacıyla hazırlanan sınıf.

GCM servislerinin test üzerinde çalıştırılabilmesi için Google Play servislerine ihtiyaç duyulur. Bu servis kurulu olmayan telefonlarda GCM servisine kayıt olma işlemi yapılamaz. Genymotion isimli emülatörde Google play servisleri kurularak GCM kayıt işlemi test edilebilmiştir. Ekran görüntüleri bu emülatörden alınmıştır.

4.5.2 Oturum açma ekranı

Oturum açma ekranında kullanıcıların android uygulaması ile oturum açması sağlanmaktadır. Oturum açma işlemi yapıldığında kullanıcı kendisine gönderilen iletileri görebilir ve konum bilgisini arkadaşları ile paylaşabilir. Şekil 4.18 oturum açma ekranını göstermektedir. Oturum bilgileri uygulama tarafından kayıt edilir. Kayıt edilen oturum bilgileri konum bildirme ve GCM sunucularına kayıt olma esnasında kullanılarak konum bilgilerinin kullanıcı ile eşleştirilmesi sağlanmaktadır. Android uygulamalarında internet servisleri üzerinden yapılacak işlemlerin asenkron sınıflar üzerinden yapılması ara yüz kilitlenmelerinin önüne geçmektedir. Bu sebeple uygulamadaki web servis temelli işlemler için asenkron sınıflar tasarlanmıştır. Asenkron çalışan metotlar sayesinde arka plan işlemleri yapılırken kullanıcı ara yüzünün de cevap verebilmesine imkan sağlanabilir.



Şekil 4.18. Oturum açma ekranı.

Kullanıcı doğrulama işlemi mezun web uygulamasında tanımlanmış olan servis üzerinden doğrulama yapmaktadır. Android uygulamalarında oturum bilgilerini saklamak için diğer uygulamaların erişimine kapalı olan bir dosya oluşturulur. Temel veri tiplerini kaydetmek için shared preferences yapısı kullanılabilir. Veri tabanı ihtiyacı gerektirmeyen temel kayıt ihtiyaçları için bu yapı tercih edilebilir. Bu işlem için öncelikle bir dosya oluşturulur. Dosya oluşturulurken diğer uygulamalara erişimi kapatmak için MODE_PRIVATE özelliği ile oluşturulur. Aşağıda bir shared preferences dosyası oluşturmak için kullanılan örnek bir kod bulabilirsiniz (Şekil 4.19).

```
SharedPreferences mPaylasilanVeriler = getSharedPreferences("xmlFile",  
MODE_PRIVATE);
```

Şekil 4.19. Shared preferences dosyası oluşturma.

4.5.3 Detay ekranlar

Duyuru ve etkinlikleri görüntülemek için sekmeli ana ekrandan ilgili butonun seçilmesi gerekmektedir. Şekil 4.21 duyuruların listelendiği ekranı göstermektedir. Sisteme test amaçlı eklenen duyurular ListView elemanında görüntülenmektedir. Duyuruların listelenmesi için özelleştirilmiş bir listview kullanılmıştır. Bu amaçla BaseAdapter sınıfından türetilerek bir adapter sınıfı oluşturulmuştur (Şekil 4.20).

```

+ import java.util.List;
public class ozelAdapter extends BaseAdapter{
    private LayoutInflater mInflater;
    private List<icerikModel> mIcerikListe;
+ public ozelAdapter(Activity activity, List<icerikModel> icerikler) {}
+ public int getCount()
+ public icerikModel getItem(int position)
+ public long getItemId(int position)
+ public View getView(int position, View convertView, ViewGroup parent)
}

```

Şekil 4.20. Verileri listelemek amacıyla oluşturulan özel adapter sınıfının görünümü.

Duyuru ve etkinliklerin görüntülenmesini sağlayan servis web uygulaması üzerinde json tipinde veri döndüren metotlar olarak tasarlanmıştır. Tasarlanan bu web metotlarına uygulama tarafından istek gönderilmekte ve gelen json veriler ayrıştırılarak listview elemanından görüntülenmektedir. İstekler asenkron bir sınıf üzerinden yapılmaktadır.

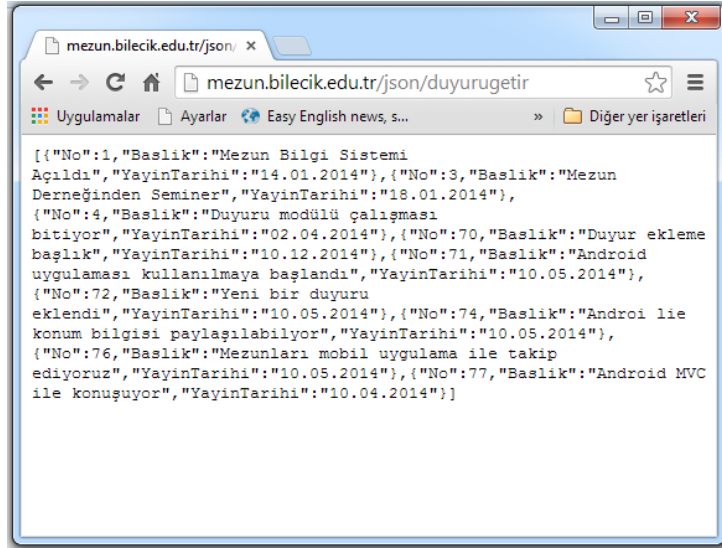
Duyuru, etkinlik ve iş ilanların görüntülenmesini sağlayan ayrı ayrı etkinlik (activity) tasarlanmıştır. Android uygulamasında kullanılan etkinliklerin (activity) eclipse ortamındaki görünümü Şekil 4.23 numaralı şekilde görülmektedir. Tanımlanan etkinliklerin yanında çeşitli servislerin kullanılmasını sağlayan bir takım sınıfların da tasarlanmış olduğu şekilden görülmektedir. Lokasyon işlemleri için LocationManager sınıfından türetilen sınıflar, Google bulut mesaj servisi (push notification) işlemleri için GCMBaseIntentService sınıfından türetilerek oluşturulmuş bir sınıf, özelleştirilmiş liste kontrolleri için BaseAdapter sınıfından türetilerek oluşturulmuş özel adapter sınıfı gibi

çeşitli ihtiyaçlara yönelik olarak sınıflar tanımlanmıştır. Android uygulamalarında çeşitli servisleri dinlemek ve servislerden veri almak için sunulan bir takım temel sınıflar temel alınarak türetilmiş sınıflar sıklıkla kullanılır. Bu temel sınıfların sağladığı metotlar aşırı yükleme yoluyla yeniden yazılarak servislerdeki değişiklikler takip edilerek işlemler gerçekleştirilebilir.

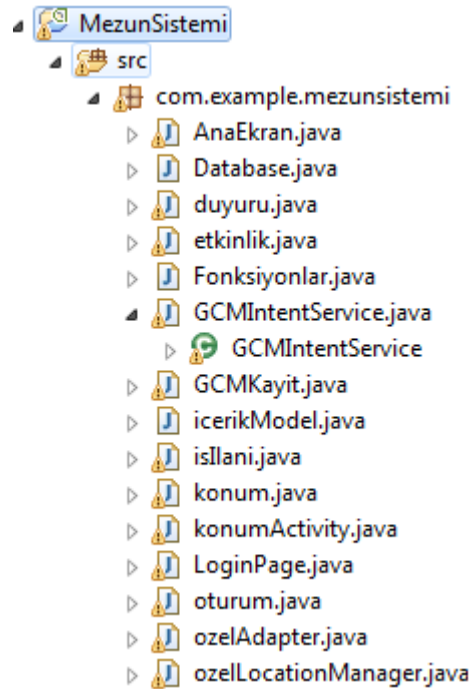


Şekil 4.21. İçeriklerin sekmelerle görünümü.

Duyuru verilerini json verisi olarak sunan metot, tarayıcı üzerinden çağrıldığında dönen veriler Şekil 4.22 numaralı şekilde görülmektedir. Json tipi, çok fazla ek işaret gerektirmeden verilerin ifade edilmesini sağlamaktadır. Mezun web uygulamasında kullanıcı adı gerektirmeden ulaşılan servisler aynı zamanda json veri yapısında da sunulmaktadır. Bu sayede farklı uygulamaların da bu servislerden veri çekebilmesi mümkün olabilmektedir.



Şekil 4.22. Duyuru verilerinin tarayıcı ile görüntülenmesi.

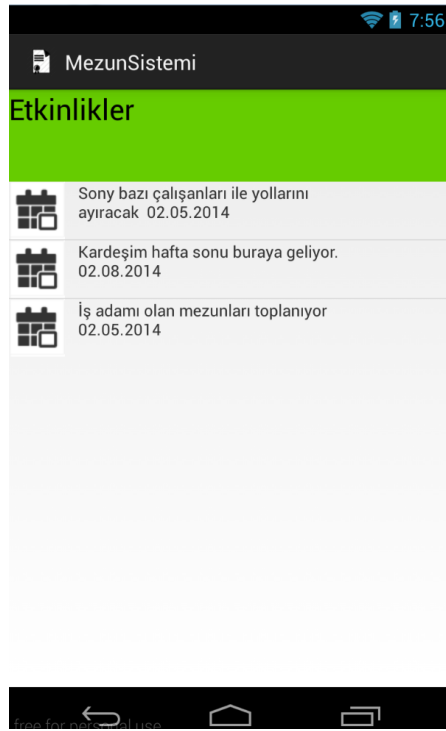


Şekil 4.23. Eclipse aracında android uygulamasında kullanılan etkinlikler (activity).

Duyurular ve etkinliklerin çağrılıp görüntülenmesi için web servise yapılan istek asenkron bir sınıf üzerinden yapılmaktadır. Asenkron sınıflar tanımlanırken AsyncTask sınıfından türetilerek tanımlanmaktadır. AsyncTask sınıfının metotları aşırı

yüklenerek arka plan işlemleri koterılmaktadır. Bu sınıfın doInBackground metodu aşırı yüklenerek servis çağırısı bu metot içinde yapılmaktadır. Servis çağırısı yapılmadan önce, yapıldığı anda ve işlem bitirildiğinde yapılabilecek işlemleri belirten metotlar bulunmaktadır. Bu metotlar tanımlanarak işlem başladığında, işlem devam ederken ve işlem bittiğinde yapılacak işlemler oluşturulabilir.

Şekil 4.25 asenkron olarak tanımlanmış bir sınıfı göstermektedir. Şekilde görüldüğü gibi aşırı yüklenmesi gereken üç metot bulunmaktadır. onPreExecute metodu arka plan işlemi başlamadan önce yapılacak işlemlerin tanımlandığı metottur. Bu metot içinde işlem devam ederken ekranda bir progressDialog penceresi çıkarılarak kullanıcının bilgilendirilmesi sağlanabilir.



Şekil 4.24. Android uygulamasında etkinliklerin görünümü.

Arka plan işlemlerinin tanımlandığı metot doInBackground isimli metottur. onPostExecute metodu arka plan işlemi bittiğinde yapılan işlemleri tanımlar. Bu metot içinde örneğin bilgilendirme penceresi kapatılabilir, başka bir etkinliği yönlendirme yapılabilir, işlem sonucu görüntülenebilir.

```

public class asenkronEtkinlikGetir extends AsyncTask<String, String, String>
{
    private ProgressDialog bilgi = new ProgressDialog(etkinlik.this);

    protected String doInBackground(String... params) {}

    protected void onPreExecute() {}

    protected void onPostExecute(String data) {}
}

```

Şekil 4.25. Eclipse aracında asenkron olarak tanımlanmış sınıfın görünümü.

4.5.4 Konum bildirme

Kullanıcılar istedikleri takdirde buldukları konum bilgisini mezun sistemine bildirebilmektedir. Konumunu bildiren bir kullanıcı mezun sistemi üzerinden arkadaşları ile konum bilgisini paylaşabilmektedir. Konum bilgisinin mezun sistemine kayıt edilmesi sırasında kullanıcı oturum açmışsa oturum bilgileriyle birlikte, oturum açmamışsa kullanıcı bilgileri olmadan konum bilgisi sisteme kayıt edilmektedir. Konum servislerinin dinlenebilmesi için LocationListener sınıfından türetilen bir sınıf tanımlanır. Tanımlanan sınıfın türetildiği sınıftan aldığı metotlar aşırı yüklenerek konum bilgisinin okunması sonucunda yapılacak işlemler tanımlanabilmektedir.

```

public class konum implements LocationListener {

    Location konum;
    konumActivity kon=new konumActivity();

    double enlem,boylam;

    public void onLocationChanged(Location loc) {}

    public void onProviderDisabled(String provider) {}

    public void onProviderEnabled(String provider) {}

    @Override
    public void onStatusChanged(String provider, int status, Bundle extras) {}
}

```

Şekil 4.26. Konum servisinin dinlenmesini sağlayan sınıf.

Android uygulamasında konum bilgilerinin sisteme kayıt edilebilmesi için

mezun uygulamasının konum kaydetme işlemini gerçekleştiren servisi çağrılmaktadır. Bu servisin çağrılması işlemi de yine asenkron sınıf üzerinden arka planda çalışacak bir işlem olarak tanımlanmıştır. Tanımlanan LocationListener sınıfı Şekil 4.26 numaralı şekilde görülmektedir.

Temel sınıftan gelen metotlar aşırı yüklenerek işlemler gerçekleştirilebilir. Örneğin kullanıcının konum bilgisi değiştiğinde tetiklenen metot onLocationChanged isimli metottur. Bu metot içinde tanımlamalar yapılarak konum bilgisi değiştiğinde sisteme kayıt edilmesi sağlanabilir. Konum servisleri açıldığında veya kapatıldığında yapılması istene işlemler de ilgili metotlar içinde tanımlanabilir. onProviderDisabled isimli metot aygıtın gps özelliği kapatıldığında devreye girer.

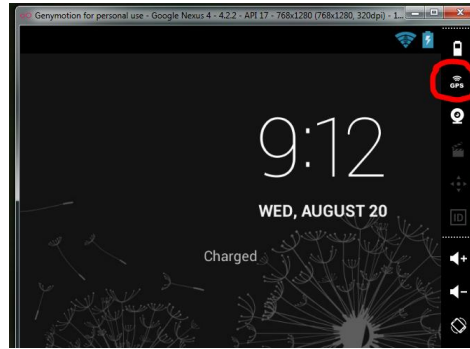
Konum bilgisinin elde edilmesi için LocationManager sınıfından yararlanılır. Bu sınıfın metotları kullanılarak konum bilgisi okunabilir. Konum bilgisinin nasıl elde edilebileceğini gösteren bir metot Şekil 4.27 numaralı şekilde görülmektedir.

Tanımlanan LocationManager tipindeki nesne üzerinden getLastKnownLocation metodu çağrılarak son konum bilgisinin elde edilmesi sağlanabilir. Bu metodun sağladığı Location nesnesi sayesinde enlem ve boylam değerleri elde edilebilir. Enlem ve boylam değerleri sisteme kayıt edilirken varsa kullanıcı bilgisi ve tarih bilgisi ile kayıt edilmektedir.

```
loc = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
provider = LocationManager.GPS_PROVIDER;
location = loc.getLastKnownLocation(provider);
if (location != null) {
double lati = location.getLatitude();
double longi = location.getLongitude();
```

Şekil 4.27. Konum bilgisinin elde edilmesi.

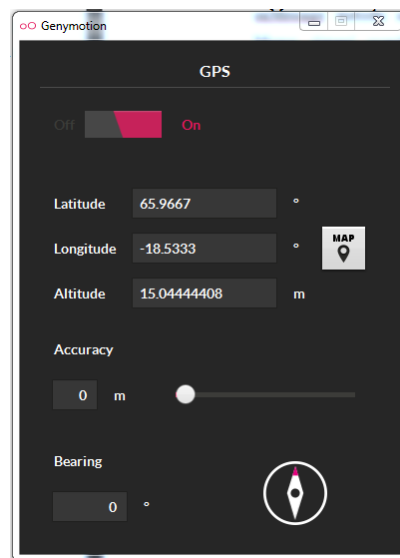
Genymotion emülatöründe konum bilgisinin test amaçlı değiştirilmesi için bir araç sunulmuştur. Ekranın sağ tarafında görülen Gps butonuna tıklanarak gps ayar ekranı açılabilir. Bu ekran üzerinden yeni konum bilgileri girilerek test yapılabilir. Şekil 4.28 genymotion emülatöründeki gps ayar butonunu göstermektedir.



Şekil 4.28. Genymotion emülatöründe gps butonu.

Gps butonuna tıklandığında açılan ekran üzerinden Gps açılarak konum bilgisi girilebilmektedir. Bu ekranda enlem boylam bilgilerini yanında rakım bilgisi de girilebilmektedir. Ekranda veri girişi yapmak için ekranın üst kısmında bulunan GPS butonun açık konumda olması gerekmektedir (Şekil 4.29). Bu ekrandaki latitüde ve logitüde değerleri değiştirilerek telefonun konum bilgisinin değişmesi modellenilebilir. Böylece değişen konum bilgisinin kayıt işlemleri test edilebilir.

Konum bilgileri kaydedilerek, kullanıcıların yakınlarda bulunan arkadaşları, buldukları konuma yakın olan kişilerin sayısı, hangi tarihte hangi konumlarda daha fazla yoğunluk yaşanmaktadır gibi birçok veriye ulaşılabilir. Konum servisleri harita servisleri ile entegre edilerek kolaylıkla harita üzerinde görüntüleme yapılabilir.



Şekil 4.29. Genymotion emülatörü gps ayarları.

Konum bilgisi sisteme kayıt edildiğinde kullanıcıyı bilgilendiren bir Toast mesajı ekranda kısa bir süre gözükür. Tost mesajı denilen bildirimler ekranın altında kısa bir süre görünüp kaybolan bilgilendirmelerdir. Konum bildirim sonucunu da kullanıcıya bu tür bir mesajla bildirilir (Şekil 4.30). Toast mesajları Toast sınıfı kullanılarak kolayca oluşturulabilir.



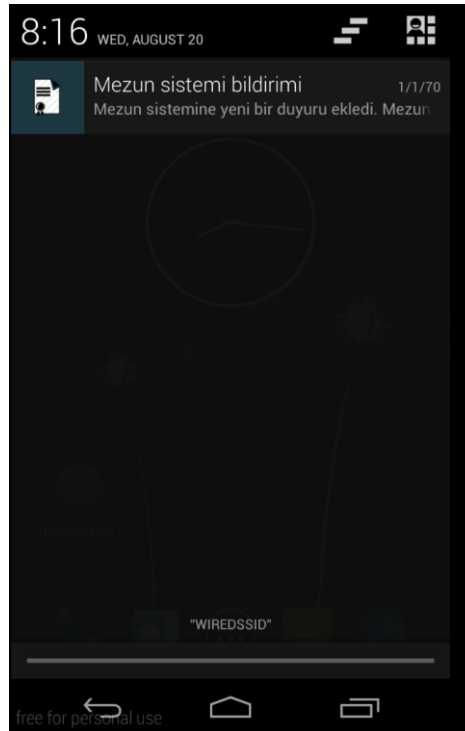
Şekil 4.30. Android uygulamasında konum bilgisinin kaydedilmesi.

4.5.5 Bildirimlerin görüntülenmesi

Sisteme yeni bir duyuru eklendiğinde Google bulut mesaj servisi sayesinde kullanıcı cihazlarında bildirim görüntülenmesi sağlanır. Bildirimlerin nasıl görüldüğüyle ilgili bir ekran görüntüsü Şekil 4.31 numaralı şekilde görülmektedir. Bildirimler çeşitli işlemler için gerçekleştirilebilir. Yeni bir arkadaş eklendiğinde, kullanıcıya arkadaşından bir mesaj geldiğinde, yeni bir etkinlik veya iş ilanı sisteme eklendiğinde kullanıcı ekranına bildirimler oluşturulabilir. Google bulut mesaj servisi sayesinde uygulamayı yükleyen kullanıcıların sistem hareketlerinden haberdar edilmesi sağlanabilir.

GCM servisinden gelen mesajların uygulama tarafından sezilerek görüntülenmesi için GCMBaseIntentService sınıfından türetilerek hazırlanan sınıfın onMessage metodu aşırı yüklenerek gelen mesajın görüntülenme şekli seçilebilir. Mezun sistemi uygulamasında gelen mesajlar bildirim olarak telefon ekranında görüntülenmektedir. Gelen mesajlar ekrana açılan bir activity yoluyla da gösterilebilir. Mezun android uygulamasında bildirim olarak görüntülenmesi tercih edilmiştir. Genel olarak uygulamalara ait mesajların bildirim olarak tanımlanması daha kullanışlı bir

yöntemdir.



Şekil 4.31. Sistem bildirimlerinin görüntülenmesi.

Hazırlanan sınıfın `onMessage` metoduna yazılan kodlar Şekil 4.32 numaralı şekilde görülmektedir. Uygulamanın gelen mesajı görüntülemesi için `Notification` nesnesi oluşturulmuş bu nesne üzerinden bildirim görüntülenmesi sağlanmıştır. Bildirim oluştururken oluşturulan bildirim başlığı ve simgesi belirtilmektedir. Bildirim metni ise gelen mesajdan alınmaktadır.

```
@SuppressWarnings("deprecation")
@Override
protected void onMessage(Context arg0, Intent intent)
{
    Intent intent2 = new Intent(this, AnaEkran.class);
    PendingIntent pnIntent = PendingIntent.getActivity(this, 0, intent, 0);

    NotificationManager notManager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
    Notification ntf = new Notification(R.drawable.ic_launcher,
        "Mezun Sisteminden Bildirim", 5000);
    ntf.setLatestEventInfo(this, "Mezun sistemi bildirim", intent
        .getExtras().getString("message"), pnIntent);

    notManager.notify(1, ntf);
}
```

Şekil 4.32. GCM servisinden gelen mesajların bildirimini sağlayan metod.

5. SONUÇ

Web tabanlı geliştirilen mezun bilgi sistemi mezunlar arasında sosyal paylaşımlar oluşturmayı ve iletişimi devam ettirmeye hedeflemiştir. Bu amaçla sosyal medya entegrasyonu ve mesajlaşma modülleri sisteme dahil edilmiştir. Sistemin MVC deseniyle geliştirilmiş olması projenin iş paylaşımını kolaylaştırmıştır. Sürüm takip sistemiyle takip edilerek geliştirilen sistem sürdürülebilir bir yapıya kavuşturulmuştur. Öğrenci bilgi sistemiyle uyumlu çalışacak bir yapıda tasarlanması da mezunlarla ilgili sayısal istatistiklere gerçek veriler üzerinden ulaşılmasını sağlamıştır. Sosyal medya uyumu sağlanarak mezunların sisteme girişleri kolaylaştırılmıştır. Sisteme tanımlanacak anketler vasıtasıyla mezunlara sorular yöneltilir. Anket sonuçları analiz edilerek mezuniyet sonrası karşılaşılan sorunlar, ihtiyaçlar, mezunların meslekleri ile ilgili uygulamadaki gelişmeler takip edilebilir.

Mezunları takip etmek, bilgilendirmek ve mezunlar arasında iletişimi sağlamak amacıyla android uygulamaları geliştirilebilir. Geliştirilen uygulamalara sosyal medya özellikleri dahil edilerek uygulamaların aktif kullanılması sağlanabilir. İlgi çekici oyunlar geliştirilerek kullanıma sunulabilir. Sisteme SMS uyumu eklenerek mezunlar SMS yoluyla bilgilendirilebilir. SMS yoluyla anketler uygulanabilir. Google mesaj servislerini farklı amaçlarla etkin şekilde kullanan uygulamalar geliştirilebilir.

Mezunların takip edilmesi, eski mezunların tecrübelerinden üniversitelerin faydalanması, yeni öğrencilerle eski mezunların buluşturulması ve bilgi paylaşımı yapılması eğitimin kalitesini arttıracaktır. Mezun sistemleri işverenler ile öğrenciler arasında bir köprü vazifesi görebilir, kariyer merkezi gibi kullanılacak şekilde planlanabilirler. Bu sayede öğrenciler henüz okul yıllarındayken meslekleriyle ilgili güncel ihtiyaçları takip edebilirler. Mezunların uzmanlık alanları sisteme güncel şekilde girilirse üniversitenin hangi yeterlikte insanlar yetiştirdiğiyle ilgili veri sağlanabilir.

Tüm üniversitelerin ortak bir mezun sistemi geliştirilmesi, yetiştirdikleri insan gücünün tek bir sistem üzerinden sorgulanması sağlanabilir. Ulusal bir kariyer merkezi oluşturularak tüm üniversite öğrencilerinin bilgi ve tecrübelerini paylaşacakları bir sistem kurgulanabilir.

6. KAYNAKÇA

Akadal E., “Mezun Takip Sistemlerinin Gerekliliği ve İstanbul Üniversitesi için Mezun Takip Sistemi Uygulama:EOK”. Yüksek Lisans Tezi, *İstanbul Üniversitesi Fen Bilimleri Enstitüsü Enformatik Ana Bilim Dalı*, İstanbul 2013.

Akdemir C., “Jquery”, *Dikeyksen Yayınları*, Ankara, (2012).

Aktaş V., Sevinç T., “MVC 4”, *Kodlab Yayınları*, İstanbul, (2013).

Algan S., “Her Yönüyle C# 4.0”, *Pusula Yayıncılık*, İstanbul (2012).

Atbaş H., “Kaliteli Yazılım Nasıl Geliştirilir”, *Pusula Yayıncılık*, İstanbul (2012)

Baykara M., “Yazılım Yaşam Döngüsü ve Yazılım Geliştirme Süreçleri”,
<http://web.firat.edu.tr/mbaykara/ym2.pdf>, 15.05.2014.

Cerrah O., “Google Maps koordinatları bilinen yerin harita üzerinde yer gösterimi.”,
<http://www.yazgelistir.com/makale/google-Maps-koordinatlari-bilinen-yerin-harita-uzerinde-yer-gosterimi> 01.05.2014.

Çelik R., “A’dan Z’ye jquery”, *Seçkin Yayıncılık*, Ankara 2013.

Bavlı U, “Sosyal Medyanın Dayanılmaz Cazibesi”, Zorlu, Sayı: 33, Nisan- Mayıs- Haziran 2011.

Derinözlü C., “Gcm Mimarisi Kullanılarak Mobil Cihazların Uzaktan Yönetilmesi”,
Android Geliştirici Günleri 2013, 2, (2013).

KAYNAKÇA (Devam Ediyor)

Erkal G., “Dördüncü kuşak yazılım geliştirme araçları ve web tabanlı mezun bilgi sistemi uygulaması”, Yüksek Lisans Tezi, *Atatürk Üniversitesi Sosyal Bilimler Enstitüsü Sayısal Yöntemler Ana Bilim Dalı*, Erzurum, 2011.

Facebook, <https://developers.facebook.com/> 15.05.2014.

Fırat A. K., “İleri Seviye Android Programlama”, *Kodlab Yayınları*, İstanbul, 2014.

Gençlik ve Spor Bakanlığı, “Gençlik ve Sosyal Medya Araştırma Raporu”, <http://www.gsb.gov.tr/HaberDetaylari/3/3816/genclik-ve-spor-bakanligi-turkiyenin-en-kapsamli-sosyal-medya-arastirmasini-yapti.aspx> 12.05.2014.

Gözüdeli Y., “Yazılımcılar için MS SQL Server 2012 Programlama”, *Seçkin Yayınları*, İstanbul, 2013.

Güçlüoğlu R. G., “Web tabanlı mezun önerme sistemi”, Yüksek Lisans Tezi, *İstanbul Aydın Üniversitesi, Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Ana Bilim Dalı*, İstanbul, 2013.

Gülbahar Y., Kalelioğlu F., Madran O., “Sosyal Ağların Eğitim Amaçlı Kullanımı” *inet-tr 2010 Türkiye’de İnternet Konferansı*, İstanbul, 2010.

Karal H., Kokoç M., “Üniversite Öğrencilerinin Sosyal Ağ Siteleri Kullanım Amaçlarını Belirlemeye Yönelik Bir Ölçek Geliştirme Çalışması”, *Turkish Journal of Computer and Mathematics Education* 1(3):251-263 (2010).

Karayel A., Bayguş Ö., Yılmaz U., “ASP.NET MVC ile E-ticarek ve İçerik Yönetimi”, *Pusula Yayıncılık*, İstanbul, (2013).

Kartal M., “Türkiye’de Sosyal Medya Raporu” <http://www.iletisimvediplomasi.com/mehmet-kartal-turkiyede-sosyal-medya-raporu/> 15.05.2014.

KAYNAKÇA (Devam Ediyor)

Lerman J., “Programming Entity Framework Second Edition”, *Safari*, 2010.

Muştu Y., “SMS ve Android tabanlı uygulamalarla sınıf içi etkileşimin artırılması”,
Yüksek Lisans Tezi, *Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü*
Bilgisayar Mühendisliği Ana Bilim Dalı, Bilecik , 2012.

Narman A. E., “Android Programlama”, *Kodlab Yayıncılık*, İstanbul 2012

Sözeri M. C., “ASP .Net MVC”, <http://mcansozeri.wordpress.com/tag/asp-net-mvc-4-e-kitap/> 10.04.2014.

Taşkın İ., “Google Maps kullanarak verilen kordinatları çizdirme”,
<http://www.yazgelistir.com/makale/google-Maps-kullanarak-verilen-kordinatları-cizdirme> 01.05.2014

Uslu B. “Android Tabanlı Mobil Uygulama Geliştirme”, *Kodlab Yayınları*, İstanbul 2013

Turkcell, “Google Bulut Mesaj Servisi”, <https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-401/google-bulut-mesaj-servisi> 15.05.2014.

<http://www.socialbakers.com> 15.05.2014.

<http://www.visualstudio.com/tr-tr/explore/app-lifecycle-management-vs> 12.05.2014.

<http://www.ozguryazilim.com.tr/bt-altyapi-cozumleri/proje-yonetimi-redmine>
17.05.2014.

<http://www.csharpnedir.com/> 01.04.2014

KAYNAKÇA (Devam Ediyor)

<http://www.erenguvercin.com/2013/05/entity-framework-yaklasmlar-ve.html>

02.05.2014

<http://www.highcharts.com/> 02.05.2014.

<http://tr-tr.facebook.com/notes/promoqube/g%C3%BCncel-facebook-t%C3%BCrkiye-istatistikleri/230108337030128> 15.05.2014.

<http://www.microsoft.com/en-us/download/details.aspx?id=29062> 10.05.2014.

<https://developers.google.com/Maps/documentation/> (22.05.2014.

<https://developers.google.com/Maps/documentation/javascript/tutorial> 25.05.2014.

<http://www.delipenguen.com/category/google-Maps-api/> 22.05.2014.

7. ÖZGEÇMİŞ

Kişisel Bilgiler

Adı Soyadı : Musa TURKAN
Doğum Yeri ve Tarihi : Akçaabat, 20.05.1979



Eğitim Durumu

Lisans Öğrenimi : Sakarya Üniversitesi, Teknik Eğitim Fakültesi,
Elektronik ve Bilgisayar Öğretmenliği
(2006)

İş Deneyimi

Çalıştığı Kurumlar : 1) Sakarya Üniversitesi Bilgi İşlem Dairesi Başkanlığı
(2004-2008)
2) Bilecik Şeyh Edebali Üniversitesi, Bilgi İşlem Dairesi Başkanlığı (2008-)

İletişim

Adres : Bilecik Şeyh Edebali Üniversitesi, Bilgi İşlem Dairesi Başkanlığı,
Gülümbe Yerleşkesi, BİLECİK
Tel: : 0228. 214 1045
E-Posta Adresi : musa.turkan @bilecik.edu.tr

Tarih: / /

İmza