

T.C.  
YÜZÜNCÜ YIL ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ  
ANABİLİM DALI

**ANDROİD VE WEB TABANLI BİR UYGULAMA İLE İŞİTME  
ENGELLİLERE YÖNELİK SESLİ DERSİ YAZILI TAKİP SİSTEMİ**

YÜKSEK LİSANS TEZİ

HAZIRLAYAN: Emre BİÇEK  
DANIŞMAN : Yrd. Doç. Dr. M. Nuri ALMALI

VAN-2016

T.C.  
YÜZÜNCÜ YIL ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ  
ANABİLİM DALI

**ANDROİD VE WEB TABANLI BİR UYGULAMA İLE İŞİTME  
ENGELLİLERE YÖNELİK SESLİ DERSİ YAZILI TAKİP SİSTEMİ**

YÜKSEK LİSANS TEZİ

HAZIRLAYAN: Emre BİÇEK

Bu çalışma Yüzüncü Yıl Üniversitesi Bilimsel Araştırma Projeleri Koordinasyon Birimi tarafından **2015-FBE-YL059** no'lu Yüksek Lisans Tez Projesi olarak desteklenmiştir.

VAN-2016

## KABUL VE ONAY SAYFASI

Elektrik-Elektronik Mühendisliği Anabilim Dalı'nda Yrd. Doç. Dr. M. Nuri ALMALI danışmanlığında, Emre BİÇEK tarafından sunulan “**Android ve Web Tabanlı Bir Uygulama ile İşitme Engellilere Yönelik Sesli Dersi Yazılı Takip Sistemi**” isimli bu çalışma Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili hükümleri gereğince 05/02/2016 tarihinde aşağıdaki jüri tarafından oy birliği ile başarılı bulunmuş ve Yüksek Lisans tezi olarak kabul edilmiştir.

Başkan:

Prof. Dr. Sabir RÜSTEMLİ

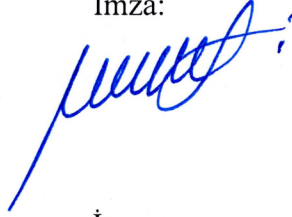
İmza:



Üye:

Doç. Dr. Naci GENÇ

İmza:



Üye:

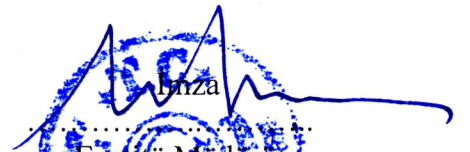
Yrd. Doç. Dr. M. Nuri ALMALI

İmza:



Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 13.02/2016 tarih ve 2016/3-1 sayılı kararı ile onaylanmıştır.

İmza



Enstitü Müdürü



## TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.



Emre BİÇEK

## ÖZET

### ANDROID VE WEB TABANLI BİR UYGULAMA İLE İŞİTME ENGELLİLERE YÖNELİK SESLİ DERSİ YAZILI TAKİP SİSTEMİ

BİÇEK, Emre

Yüksek Lisans Tezi, Elektrik-Elektronik Mühendisliği Anabilim Dalı

Tez Danışmanı: Yrd. Doç. Dr. M. Nuri ALMALI

Şubat 2016, 78 sayfa

Gelişen teknolojiler, insanlara her alanda destekleyici çözümler sunmaktadır. Bu tezde, işitme engellilere yönelik Android ve Web tabanlı bir uygulama geliştirilmiştir. İşitme engelliler, birbirleriyle iletişim kurmak için işaret dili kullanmaktadırlar. İşaret dili bilmeyen bireylerin, işitme engellilerle iletişim kurmaları güçtür. Bu çalışmada gelişen yazılım teknolojileri kullanılarak bu soruna çözüm sunulmaya çalışılmıştır. Sesi yazıya çeviren teknolojiler ve buna ek çeşitli yazılımlar kullanılarak bütünlük bir sistem geliştirilmiştir.

Sesli Dersi Yazılı Takip adı verilen sistem kapsamında, Android işletim sistemi üzerinde çalışan mobil bir uygulama ile kullanıcıların konuşmaları yazıya çevrilmekte ve anlık olarak uzak bir sunucuya gönderilmektedir. Uzak sunucudaki veri tabanına kaydedilen konuşma metinleri, internet üzerinden asenkron veri alışverişi yapabilen (AJAX) bir yazılım ile Web sayfaları ve Android telefonlar üzerinden anlık olarak takip edilebilmektedir. Tez çalışması kapsamında Web tabanlı açık bir ders yönetim sistemi geliştirilmiş ve anlatılan tüm derslerin daha sonra ulaşılabilir olması sağlanmıştır.

**Anahtar kelimeler:** İşitme engelli, ANDROID, AJAX, Konuşmanın yazıya çevrilmesi



## **ABSTRACT**

### **REAL-TIME INSCRIPTIVE FOLLOW UP SYSTEM OF AUDIBLE LECTURE WITH AN ANDROID BASED AND WEB BASED APPLICATION**

BiÇEK, Emre

M. Sc., Electrical-Electronics Engineering

Supervisor : Asst. Prof. Dr. M. Nuri ALMALI

February 2016, 78 pages

Current development in technologies offers people supporting solutions in all areas. In this study, it is aimed to develop an Android and Web-based applications for the hearing impaired. Hearing impaired use sign language to communicate with each other. People who do not know sign language it is difficult to communicate with hearing impaired. In this study, it is tried to solve this problem with developed software technologies. An integrated system has been developed using with speech to text technologies and additional software.

The scope of the system which named as "Sesli Dersi Yazılı Takip" with an application running on the Android operating system, user's speeches are converted into text (Speech-to-Text) and instantly transferred to remote server. Spoken text which recorded on database of the remote server could be followed up in real time on Android mobile phones and Web pages by the software that has capable of asynchronous data exchange over the Internet (AJAX). Under the thesis, an open Web-based course management system has been developed and all courses are provided that can be accessed later.

**Key words:** Hearing impaired, ANDROID, AJAX, Speech-to-Text





## **ÖN SÖZ**

Bu tez çalışmasına verdiği destekten dolayı Yüzüncü Yıl Üniversitesi Bilimsel Araştırma Projeleri Koordinasyon Birimi'ne, her türlü ilgi ve yardımlarını esirgemeyen danışmanım Sayın Yrd. Doç. Dr. M. Nuri ALMALI'ya, ayrıca çalışmalarımda bana her zaman destek olan aileme, eşime ve değerli dostlarıma en içten teşekkürlerimi sunarım.

2016

Emre BİÇEK



## İÇİNDEKİLER

|                                                                         | Sayfa |
|-------------------------------------------------------------------------|-------|
| ÖZET .....                                                              | i     |
| ABSTRACT .....                                                          | iii   |
| ÖN SÖZ.....                                                             | v     |
| İÇİNDEKİLER.....                                                        | vii   |
| ÇİZELGELER LİSTESİ .....                                                | ix    |
| ŞEKİLLER LİSTESİ.....                                                   | xi    |
| SİMGELER VE KISALTMALAR .....                                           | xiii  |
| EKLER DİZİNİ .....                                                      | xv    |
| 1. GİRİŞ.....                                                           | 1     |
| 1.1. İşitme Engeli.....                                                 | 3     |
| 1.2. İşitme Engelli Bireylerin İletişim Yöntemleri .....                | 5     |
| 1.3. İşitme Engellilere Yönelik Cihaz Teknolojileri.....                | 6     |
| 2. LİTERATÜR BİLDİRİŞLERİ .....                                         | 10    |
| 3. MATERYAL ve YÖNTEM .....                                             | 14    |
| 3.1. Kullanılan Yazılım Teknolojileri ve Ortamları .....                | 14    |
| 3.1.1. Android işletim sistemi.....                                     | 15    |
| 3.1.2. Sistemin veri tabanı .....                                       | 18    |
| 3.1.3. Konuşmanın yazıya çevrilmesi (Speech-to-Text) entegrasyonu ..... | 21    |
| 3.1.4. AJAX entegrasyonu.....                                           | 28    |
| 3.2. SDYT (Sesli Dersi Yazılı Takip) Uygulaması .....                   | 32    |
| 3.2.1. Android uygulama .....                                           | 33    |
| 3.2.1.1 Giriş ekranı .....                                              | 34    |
| 3.2.1.2 Giriş paneli .....                                              | 35    |
| 3.2.1.3 Ders seçim ekranı .....                                         | 36    |
| 3.2.1.4 Sesi yazıya dönüştürme ekranı .....                             | 37    |
| 3.2.1.5 Canlı ders takip ekranı.....                                    | 38    |

|                                       |    |
|---------------------------------------|----|
| 3.2.2. Web uygulama .....             | 39 |
| 3.2.2.1 Ana sayfa .....               | 40 |
| 3.2.2.2 Hakkında .....                | 41 |
| 3.2.2.3 Kayıt paneli .....            | 41 |
| 3.2.2.4 Canlı takip .....             | 43 |
| 3.2.2.5 Anlatıcı yönetim paneli ..... | 44 |
| 3.2.2.6 Takipçi yönetim paneli .....  | 47 |
| 4. TARTIŞMA VE SONUÇ .....            | 49 |
| 5. KAYNAKLAR .....                    | 52 |
| EKLER .....                           | 55 |
| ÖZGEÇMİŞ .....                        | 78 |

## ÇİZELGELER LİSTESİ

| Çizelge                                                                     | Sayfa |
|-----------------------------------------------------------------------------|-------|
| Çizelge 1.1. İşitme kaybı dereceleri (Ataş ve ark., 2003) .....             | 4     |
| Çizelge 3.1. RecognizerIntent sınıfının projede kullanılan özellikleri..... | 25    |



## ŞEKİLLER LİSTESİ

| Şekil                                                                                            | Sayfa |
|--------------------------------------------------------------------------------------------------|-------|
| Şekil 3.1. Android zaman çizelgesi. ....                                                         | 16    |
| Şekil 3.2. Android mimarisi. ....                                                                | 16    |
| Şekil 3.3. Tablo nitelikleri ve ilişki şeması. ....                                              | 19    |
| Şekil 3.4. Standart ses tanıma sistemi yapısı (Tebelskis, 1995).....                             | 21    |
| Şekil 3.5. İki durumlu ve iki çıkışlı basit bir Hidden Markov Modeli<br>(Tebelskis, 1995).....   | 22    |
| Şekil 3.6. İnternet izinlerinin verilmesi.....                                                   | 23    |
| Şekil 3.7. RecognizerIntent sınıfının import edilmesi. ....                                      | 23    |
| Şekil 3.8. Kod kanalının tanımlanması. ....                                                      | 23    |
| Şekil 3.9. Konuşma ekranındaki işlev butonlarının ara yüzde tanımlanması.....                    | 24    |
| Şekil 3.10. Speech-to-Text ekranının tanımlanması ve ayarlarının yapılması. ....                 | 25    |
| Şekil 3.11. onActivityResult metodu ile sunucusundan verilerin alınması.....                     | 26    |
| Şekil 3.12. Konuşmayı veri tabanına kaydetme fonksiyonu. ....                                    | 27    |
| Şekil 3.13. Klasik web uygulamaları ile Ajax modelinin karşılaştırılması<br>(Garrett, 2005)..... | 30    |
| Şekil 3.14. Konuşmayı anlık olarak çeken AJAX kodu.....                                          | 31    |
| Şekil 3.15. ajax_islemler.php sayfası.....                                                       | 31    |
| Şekil 3.16. İçerik getir fonksiyonu. ....                                                        | 32    |
| Şekil 3.17. Açılış ekranı. ....                                                                  | 33    |
| Şekil 3.18. Giriş ekranı.....                                                                    | 34    |
| Şekil 3.19. Konuşmacı giriş ekranı.....                                                          | 35    |
| Şekil 3.20. Canlı takibe giriş ekranı. ....                                                      | 36    |

| <b>Şekil</b>                                                    | <b>Sayfa</b> |
|-----------------------------------------------------------------|--------------|
| Şekil 3.21. Ders seçim ekranı. ....                             | 37           |
| Şekil 3.22. Sesi yazıya dönüştürme ekranı. ....                 | 38           |
| Şekil 3.23. Canlı ders takip ekranı. ....                       | 39           |
| Şekil 3.24. Ana sayfa. ....                                     | 40           |
| Şekil 3.25. Hakkında ekranı. ....                               | 41           |
| Şekil 3.26. Kayıt paneli. ....                                  | 42           |
| Şekil 3.27. Canlı ders giriş ekranı. ....                       | 43           |
| Şekil 3.28. Canlı ders takip ekranı. ....                       | 44           |
| Şekil 3.29. Anlatıcı yönetim paneli ana sayfa. ....             | 44           |
| Şekil 3.30. Anlatıcı yönetim paneli ders ekleme ekranı. ....    | 46           |
| Şekil 3.31. Anlatıcı yönetim paneli ders düzenleme ekranı. .... | 46           |
| Şekil 3.32. Anlatıcı yönetim paneli üye bilgileri ekranı. ....  | 47           |
| Şekil 3.33. Takipçi yönetim paneli ana sayfa. ....              | 48           |



## SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

### Simgeler

### Açıklama

|           |            |
|-----------|------------|
| <b>ms</b> | Milisaniye |
| <b>Hz</b> | Hertz      |
| <b>dB</b> | Desibel    |

### Kısaltmalar

### Açıklama

|             |                                   |
|-------------|-----------------------------------|
| <b>AJAX</b> | Asynchronous JavaScript and XML   |
| <b>API</b>  | Application Programming Interface |
| <b>AVD</b>  | Android Virtual Device            |
| <b>BBC</b>  | British Broadcasting Corporation  |
| <b>CSS</b>  | Cascading Style Sheets            |
| <b>DSP</b>  | Digital Signal Processing         |
| <b>GPS</b>  | Global Positioning System         |
| <b>HMM</b>  | Hidden Markov Model               |
| <b>HTML</b> | Hypertext Markup Language         |
| <b>HTTP</b> | Hypertext Transfer Protocol       |
| <b>HTS</b>  | Hosted Transcription Service      |
| <b>IBM</b>  | International Business Machine    |
| <b>IOS</b>  | iPhone Operating System           |
| <b>JDK</b>  | Java Development Kit              |
| <b>NCI</b>  | National Captioning Institute     |
| <b>OHA</b>  | Open Handset Alliance             |
| <b>PHP</b>  | PHP: Hypertext Preprocessor       |
| <b>SDK</b>  | Software Development Kit          |
| <b>SDYT</b> | Sesli Dersi Yazılı Takip          |

|             |                            |
|-------------|----------------------------|
| <b>SMS</b>  | Short Message Service      |
| <b>SQL</b>  | Structured Query Language  |
| <b>TSI</b>  | Translation System Inc.    |
| <b>TTYs</b> | Teletypewriters            |
| <b>WHO</b>  | World Health Organization  |
| <b>XML</b>  | Extensible Markup Language |

## EKLER DİZİNİ

| Ek                                 | Sayfa |
|------------------------------------|-------|
| Ek-1: Mobil uygulama kodları. .... | 55    |

## 1. GİRİŞ

Gelişen teknolojiler her geçen gün insanlığa birçok alanda kolaylıklar sunmaktadır. Değişen yaşam standartları ve gereksinimler teknolojinin hızla kendini yenilemesi ihtiyacını doğurmaktadır. Cep telefonları önceleri basit özellikleri olan ve sadece iletişim sağlayan araçlar iken zamanla iletişim hızının artması ve teknolojinin çok hızlı bir şekilde tüketilmeye başlanmasıyla iletişim ihtiyaçları da değişmeye başlamış ve zamanın daha ekonomik kullanılabilmesi amacıyla yeni teknolojiler üretilmeye başlanmıştır. Cep telefonları artık iletişim aracı olmaktan çıkıp adeta insanların yaşam arkadaşı olup, vücudun bir uzvu haline gelmiştir.

Son yıllarda cep telefonları yazılımlarında devrim olmuştur. Önceleri sadece ses ve metin iletimi sağlayan cihazların yerini günümüzde akıllı telefon diye tabir edilen ve bir bilgisayarın neredeyse tüm özelliklerini içiren yüksek seviyeli dillerle geliştirilmiş cihazlar almıştır. Bu cihazlar sahip oldukları özelliklerle insanlığa teknolojik çözümler sunarak yaşamlarını kolaylaştırmaktadırlar. Bu telefonlar sadece günlük yaşamı kolaylaştırmanın yanı sıra bilim dünyasına da çeşitli uygulamalarla çözümler sunmakta ve faydalı olmaktadır. Akıllı telefon üreten firmalar (Apple, Samsung, Sony, Nokia vs.) cihazlarında mobil işletim sistemi olarak çeşitli yazılımlar kullanılmaktadır (Android, IOS, Windows Mobile vs.) ve insanlara bu yazılımları kullanarak uygulama geliştirme imkânı sunmaktadırlar. Her firmanın genel olarak market diye tabir ettikleri bir uygulama yükleme platformu vardır ve bu platformlar üzerinden akıllı telefon kullanıcıları için geliştirilmiş milyonlarca uygulamayı kullanıcıların hizmetine sunmaktadırlar. Uygulama çeşitliliği oldukça zengin olan uygulama marketleri; eğitim, sağlık, teknoloji, bilim, sanat, vs. birçok alanda geliştirilmiş ücretli veya ücretsiz uygulamaları barındırmaktadırlar.

Geçmişten günümüze yazılım ihtiyaçları her zaman değişime uğramıştır. Genel olarak Masaüstü uygulama geliştiriciliği, internetin yayılması ile yerini daha esnek ve daha ulaşılabilir olan web tabanlı uygulamalara bırakmıştır. İnsanların teknolojiye her yerden ulaşabilme isteği günümüzde mobil uygulama geliştiriciliği kavramını doğurmuştur. Mobil uygulamalarla insanlar hemen her yerden internete ulaşabilmekte ve ihtiyaçlarına anında çözüm bulabilmektedirler. Mobil uygulamaların kolay

ulařılabilirlięi ve eřitlilięi hem kullanıcılara hem de uygulama geliřtiricilere byk kolaylıklar sunmaktadır, bu nedenle gnmzde minyonlarca uygulama ve milyarlarca uygulama indirme sayılarından bahsedilmektedir.

İřitme engellilerinin iřaret dilinin kullanılmadıęı bir dersi takip etmelerinin ne denli zor olduęu bilinmektedir. ęrenciler, kendilerine zel okullarda iřaret dili bilen eęitmenler tarafından eřitli zel ęretim metotları kullanılarak verilen derslerle eęitim almaktadırlar. İřaret dilini bilmeyen bir eęitmenin bu okullarda ders verebilmesi veya bir seminer kapsamında ęrencilere bilgilerini aktarması ok gtr. Bu durum ęrencilerin geliřimi aısından olumsuz bir durumdur. Buradan hareketle bahsini ettięimiz geliřen mobil yazılım teknolojileri yardımıyla bu probleme bir katkıda bulunabilmek adına bu proje arařtırmaya konu edilmiřtir.

Bu tezde iřitme engellilere ynelik bir ANDROID ve Web tabanlı bir yazılım projesi olan SDYT (Sesli Dersi Yazılı Takip) uygulaması gerekleřtirilmiřtir. Geliřtirilmiř olan yazılımla iřitme engellilerin herhangi bir dersi grme duyularını kullanarak yazılı bir řekilde takip etmeleri saęlanmıřtır. Bu baęlamda dersi anlatan hocanın konuřması senkronize bir řekilde yazıya dnřtrlmř ve mobil telefonlar zerinden ders takibi saęlandıęı gibi ayrıca; bir web sayfasına anlık olarak aktarılan konuřma metinleri bir projektr veya akıllı tahta aracılıęıyla da ekrana aktarılıp ęrencilerin dersi yazılı olarak takip etmeleri saęlanmıřtır. Geliřtirilmiř yazılımda birbirine btnleřtirilmiř sistemlerle sonuca ulařılmıřtır. Temel olarak ANDROID iřletim sistemi zerinde geliřtirilmiř bir yazılım ile ses yazıya dnřtrlp uzak bir sunucuya aktarılmıř ve son olarak da web tabanlı bir uygulama ile AJAX (Asynchronous JavaScript and XML) kullanılarak anlık olarak ekrana yazdırılmıřtır. Ayrıca geliřtirilmiř bir aık ders ynetim sistemi ile uygulama zerinde yapılan tm derslerin daha sonra ulařılabilir ve dzenlenebilir olması saęlanmıřtır.

Grboęa ve Kargın (2003), iřitme engelli yetiřkinlerin farklı ortamlarda kullandıkları iletiřim yntemlerinin ve becerilerinin incelenmesi adına yaptıkları bir arařtırmada 69'u erkek, 31'i kadın toplam 100 iřitme engelli bireye bir takım sorular yneltilmiř ve bu sorulara verilen cevaplara gre onların farklı ortamlarda en sık kullandıkları iletiřim yntemleri tespit edilmeye alıřılmıřtır. Arařtırmaya gre; İřitme engellilere yneltilen “Hi tanımadıęınız birisinden bir konu hakkında bilgi almak

istediğinizde anlatılanları hangi iletişim yöntemiyle anlarsınız?” şeklindeki yedi seçenekli soruya cevap olarak; en fazla işaret dili ile (60 kişi), ikinci olarak ise yazı ile (54 kişi) iletişime geçtiklerinde daha iyi anladıklarını belirtmişlerdir. Bu veriler ışığında gerçekleştirilmiş olan SDYT uygulamasının işitme engelli bireylerin iletişim, eğitim gibi ihtiyaçlarına getireceği kolaylık ve sağlayacağı katkı ortaya çıkmaktadır.

### 1.1. İşitme Engeli

(WHO,2013)’e göre Dünya genelinde 360 milyon insan işitme problemi yaşamaktadır, bu da dünya nüfusunun yaklaşık %5’i ne karşılık gelmektedir. İşitme problemi yaşayan insanların 32 milyonunu 15 yaşın altındaki çocuklar oluşturmaktadır. İşitme kaybına neden olan faktörler; genetik, doğum anında gelişebilecek hasarlar, ciddi bulaşıcı hastalıklar, kronik kulak enfeksiyonları, bazı ilaçlar, aşırı derecede sese maruz kalma ve yaşlanma olarak gösterilmektedir. İşitme testleri sonucu 0-25 dB arasındaki sesleri duyabilen insanlar normal bir işitmeye sahiptir denir.

İşitme bozukluğu, diğer sağlık sorunlarına oranla daha yaygın olarak görülmektedir. Genel olarak odyometre ve konuşma ayırt etme testleri ile işitme bozukluğu teşhis edilir ancak; bunların dışında da özelleştirilmiş bazı diğer testlerde kullanılmaktadır. İşitme bozukluğu nedenleri yaygın olarak iletim tipi ve sinirsel işitme bozukluğu adı altında iki farklı grupta incelenebilir. İletim tipi işitme bozukluğu, dış ve orta kulaktan iç kulağa giden iletim yollarındaki bozuklukların ses seviyesini azaltması hatta tamamen engellemesinden kaynaklanır. Sinirsel İşitme bozukluğu ise iç kulaktaki algılayıcı hücrelerin (tüylü hücreler) zarar görmesi ya da hiç olmamasından kaynaklanır, genelde kalıcıdır, hafif, orta ve ağır olarak nitelendirilmektedir (Moller, 2006).

İşitme kaybının belirlenmesi, işitme kaybına uygun rehabilitasyon çalışmalarının yapılması, işitme kaybında ihtiyaç duyulması durumunda işitme cihazlarının uygulanması ve kontrollerinin yapılması alanında çalışan bilime Odyoloji denir. Erken dönemde teşhis edilen işitme kayıplarında, doğru tedavi ile hastaların iletişim sorunu önemli ölçüde iyileştirilebilmektedir. Bu yüzden sağlık alanında çalışan diğer personellerinde hastalarında işitme sorunu olduğunu fark etmeleri durumunda hemen ilgili bölümlere yönlendirmeleri erken teşhis ve tedavi için çok önemlidir.

İşitme kaybının tespit edilmesi amacıyla kullanılan ve saf ses çıkarabilen araçlara Odyometre denir. Odyometre cihazlarının çıkardıkları ses şiddeti işitme kaybının teşhisinde belirleyicidir. Günümüzde kullanılan dijital odyometreler ile 125 Hz ile 1800 Hz arasında değerlendirme yapılabilmektedir. İnsan kulağı tarafından hissedilebilen farklı frekanslardaki minimum ses şiddeti işitme düzeyi olarak kabul edilmiştir. Bu değer 0 dB olarak belirlenmiş ve odyometre ile ölçümler buna göre yapılmıştır. Çizelge 1.1.'de bu cihazlarla yapılan ölçümler ile elde edilen saf ses ortalamasına göre işitme kaybı dereceleri gösterilmektedir (Ataş ve ark., 2003).

Çizelge 1.1. İşitme kaybı dereceleri (Ataş ve ark., 2003)

| Saf ses ortalaması (dB) | İşitme kaybı derecesi           |
|-------------------------|---------------------------------|
| 0-15 dB                 | Normal işitme                   |
| 16-40 dB                | Çok hafif derecede işitme kaybı |
| 41-55 dB                | Hafif derecede işitme kaybı     |
| 56-70 dB                | Orta derecede işitme kaybı      |
| 71-90 dB                | İleri derecede işitme kaybı     |
| 91 dB ve üzeri          | Çok ileri derecede işitme kaybı |

İşitme engelliler yüksek seviyede işitme kaybı olan dolayısıyla çok az duyan veya hiç duymayan kişilerdir. İşitme Engelinin bireyler üzerindeki en önemli etkisi diğer insanlarla bireysel iletişim yeteneklerinin gelişmemesi ve dil gelişimlerinin normal bireylere göre gecikmeye uğramasıdır. İşitme Engelli bireyler kendileriyle aynı engele sahip bireylerle aynı sınıflarda aynı şartlarda eğitim almaktadırlar. Bu eğitimler sözlü ve yazılı verilebildiği gibi özelleştirilmiş bir dil olan işitme dili ile de verilmektedir (WHO, 2013).

## 1.2. İşitme Engelli Bireylerin İletişim Yöntemleri

İşitme engelli bireylerin en fazla etkilendiği konulardan biri dil gelişimidir. Bebekler doğum anından itibaren çevreleri ile iletişime geçerler, nesneler ve sesler onların ilgilerini çeker ve sürekli olarak bir iletişim ve gelişim sürecine girerler, duyduklarına tepkiler verir, kendi seslerini tanıyarak kontrol edebilirler. İleri seviyede işitme engeline sahip bebeklerde bu durum daha farklıdır, onlar sadece görüş alanlarındaki nesne ve olaylarla ilgilenirler.

Çocuğu engelli olan aileler, çocuklarının tepkilerinin az olması nedeni ile çocuklarıyla iletişimlerini büyük ölçüde azaltırlar. Ailelerin kullandıkları sözel ifadeler değişir, daha kısa veya abartılı olmaya başlar. Dolayısıyla işitme engelli bebeklerde ilk aylara oranla sesli ifadeler 9 ay gibi bir süre sonra yok olur, sesin geldiği noktaya odaklanma hareketi görülmez. İşiten çocuklar gibi kendi kendilerine öğrenmeleri gerçekleşmez ve bu gelişimlerini tamamlamak için destek eğitime ihtiyaçları vardır.

İşitme engelli çocuklar isteklerini dile getirmede zorluk çekerler. Yaşadıkları ortamdaki insanlar onları anlamakta güçlük yaşarlar. Bu da onların gelişimini etkilediği gibi özgüvenlerini de olumsuz etkiler. En başından beri bu bireylere işiten çocuklardan farklı davranmayarak, onları toplumda etkin hale getirebilmek için okul ve aile ortak hareket etmeli ve en uygun eğitim ortamları oluşturulmalıdır (Akçamete, 2003).

İşitme engellilerin eğitiminden sorumlu öğretmenin karşılaştığı başlıca problem, öğrencileri ile nasıl iletişime geçeceği ve onlara da başkaları ile nasıl iletişim kurmaları gerektiğini öğreteceğidir. Tüfekçioğlu (1998), işitme engelli bireylere iletişim yeteneğinin kazandırılmasında izlenen iki temel yöntemin, sözlü dil kullanan iletişim yöntemleri ve işaret dili kullanan iletişim yöntemleri olarak bildirmiş ve bu yöntemleri şu şekilde açıklamıştır;

### **Sözlü dil kullanan iletişim yöntemleri:**

- Yapısal Sözel/Oral Yöntem, dilin öğretilmesinin mümkün olduğu ve öğretilmesi gerektiği inancıyla kurgulanmış dil, belirli kalıplar ve belirli bir düzende bireylere öğretilmektedir. Bu yapılandırma, eklemeleme çalışmaları, konuşma eğitimi,



dudak okuma, işitme eğitimi, okuma ve dil eğitimi gibi bir takım çalışmaların bütünüdür. Çalışmalar, dilin öğretilmesi esassından yola çıkılarak yapılır.

- Doğal İşitsel/Sözel Yaklaşım, işitme engeline sahip bireylerinde aynı işiten çocuklar gibi dili öğrenebileceğini ve onların etkileşime geçebileceği bir ortam sağlandığında işiten çocuklara oranla daha yavaş bir hızda fakat onlarla eşit seviyede dili öğrenebileceklerini savunmuştur. Bu yöntemde işitme cihazlarının kullanımı, ön şarttır. Eğitim gören bireylerin işitme kalıntısının tam anlamıyla en yüksek seviyede olması gerektiğini savunur.

- İşitsel Yaklaşım/Tek Duyulu Yöntem, sözel iletişim eğitiminin sadece duyma ile gerçekleşebileceğini öngören bu yönetim amacı, işitmeyi çocuk yaşlarda işitme engellinin kişiliği ile birleştirmektir. Bu sebeple işitme kaydı çok erken yaşlardan itibaren tespit edilip çocuk hızlıca cihazlandırılmalıdır. Doğal İşitsel/Sözel Yöntem yaklaşımdan farkı, yalnızca cihazlandırmaya odaklanması ve dudak okumaya yer vermemesidir.

### **İşaret kullanan başlıca iletişim yöntemleri:**

- İşaret Dili, el ile oluşturulan yerel ya da ulusal bazı işaretlerin kullanıldığı görsel bir iletişim sistemidir. Konuşma dilinden farklı olarak kendi sistemi olan, farklı söz dizim özelliklerine sahip bir dildir.

- Parmak Alfabesi, parmakların değişik şekillerde konumlandırılması ile cümlelerin ve sözcüklerin ifade edilmesidir.

- Tüm İletişim, yukarıda bahsedilen iletişim yöntemlerinin tümünün birlikte kullanıldığı eğitim yöntemidir. Ancak işaret dili kullanımı daha yoğundur.

### **1.3. İşitme Engellilere Yönelik Cihaz Teknolojileri**

İleri düzeyde işitme kaybına sahip olan bireyler, küçük yaşlardan itibaren çevreleri ile iletişim kurmakta güçlük çekerler ve yaşlılarına oranla dil gelişimleri büyük oranda geri kalır. Normal bireylerin eğitim gördükleri alanlarda onlarla birlikte eğitim görmeleri, onların sahip olduğu hassasiyetler ve kısıtlı algılama özelliklerinden dolayı mümkün olmamaktadır. İşitme engelli bireyler kendileri ile aynı engele sahip akranları

ile birlikte onların teknik gereksinimlerini karşılayabilecek donanımlı eğitim ortamlarında eğitim almalıdırlar.

İşitme engelliler için üretilen ilk yüksek teknoloji cihaz telefondur. Alexander Graham Bell işitme engeline sahip kız kardeşine yardım etmek amacıyla çıktığı yolun sonunda 1876 yılında telefonu icat etmiştir. O tarihten günümüze gelişen bilgisayar teknolojileri ve tıp alanındaki ilerlemelerle birlikte işitme cihazı teknolojileri iyi bir seviyeye ulaşmıştır. Bu cihazlar işitme zorluğu çeken insanlara her alanda büyük faydalar sağlamaktadırlar (Hasselbring ve Ark., 2000).

Duymaya yardımcı cihazlardaki çeşitlilik ve işlevsellik her geçen gün artmaktadır. Bu cihazları, çözüm sunduğu problemlere ve yapılarına göre çeşitli sınıflara ayırabiliriz. Ancak, protez olarak düşünüldüğünde temel olarak iki gruba ayırmak mümkündür. Bunlar, akustik iletişim gerçekleştiren protezler, diğer bir tabir ile işitme cihazları ve elektriksel uyarım oluşturan protezler, diğer bir tabir ile iç kulak protezleridir (Tüfekçioğlu, 1998).

Akustik uyarım oluşturan protezler sesin yükseltilmesini sağlayan işitme cihazlarıdır. Cep tipi, gözlük tipi, kulak arkası tipi ve kanal içi tipleri bulunmaktadır. Bu cihazlar var olan işitme bozukluğuna kalıcı bir çözüm getirmez, mikrofon yardımı ile dışardan aldığı sesleri yükselterek bireyde var olan işitme kalıntısının en verimli şekilde kullanılmasına yardımcı olur ve yükseltilmiş bu sesleri bir alıcı yardımı ile kullanıcının kulağına gönderir. Genel yapısı itibarıyla yükseltici, mikrofon ve alıcı olmak üzere üç parçadan oluşur (Akçamete, 2003).

İşitme kaybı sebebi ile kullanılan işitme cihazları, dışarıdan gelen sesleri yükselterek kokleadaki tüylü hücreleri uyarır ve sesin kulaktaki gerekli merkezi duyma sistemlerine ulaşmasını sağlar. Çok ileri derecedeki işitme kayıplı bireylerde ise kokleadaki tüylü hücrelerin büyük bölümünün yapısı bozuktur veya hiç hücre oluşmamıştır. Bu durumdaki bireylerde normal işitme cihazları yerine koklear implantlar kullanılmaktadır. Koklear implantlar normal işitme cihazlarından farklı olarak kulaktaki tüylü hücreleri devreden çıkarmakta ve doğrudan işitme sinirindeki canlı nöronları uyatarak ses iletimini sağlamaktadır. Koklear implantlar cerrahi yöntemler ile kulağa yerleştirilmektedir (Wilson ve Dorman, 2009).

İşitme cihazlarının kullanım amacı yalın bir ifade ile ortamdaki sesleri yükselterek işitme engellilerin duyabileceği seviyeye getirmektir. Bu amaç için geliştirilen cihazlar çalışma prensiplerine göre genel olarak iki grupta incelenebilir. Cihazlardaki ayırım sesi işleme ve iletim özelliklerine göre belirlenir.

İşitme engelliler tarafından kullanılan ilk cihazlar analog işitme cihazlarıydı. Analog cihazlar ortamdaki akustik sesi herhangi bir işleminden geçirmeden (frekans işleme, gürültü engelleme vb.) doğrudan doğruya bireyin kulağına iletmesi ile gerçekleşir. İletilen bu sesler herhangi bir işleme tabi tutulmadığından kullanıcı açısından bazı dezavantajları da beraberinde getirmektedir. Bunlar, ortamdaki var olan gürültünün kullanıcıyı rahatsız etmesi ve konuşulanların anlaşılmasının güçlüğü, ses ayarının manuel olarak yapılması dolayısıyla her ortamda ayar yapma ihtiyacı, sürekli olarak yüksek ses işitilmesinden kaynaklı işitme bozukluğunda artış olarak söylenebilir.

1996'dan önce yaygın olarak analog işitme cihazları kullanılmakta idi. 1996'dan sonra DSP (Digital Signal Processing – Dijital Sinyal İşleme) algoritmaları konuşulmaya ve uygulanmaya başlandı. Bu algoritmaya sahip cihazlar konforlu kullanımları ve sinyal kaybı problemlerini en aza indirmesi ile kullanıcıları tarafından oldukça popüler hale gelmiştir. 2005'li yıllara gelindiğinde bu cihazların kullanım oranı Amerika'da %93'ü bulmuştur. DSP teknolojisine sahip cihazlar gürültü azaltma, çevre sınıflandırması, daha net ses iletimi, bilgisayardan cihazı programlama gibi özellikleriyle analog cihazlardan ayrılmaktadırlar.

Günümüzde teknolojinin hızla gelişmesiyle insanların beklentileri de buna paralel olarak artmaktadır. Son on yılda işitme cihaz teknolojilerindeki gelişmeler bu alanda daha yenilikçi neler yapılabilir sorusunu akıllara getirmektedir. Şimdilerde konuşulan teknoloji ise şüphesiz wireless teknolojisidir. Wireless teknolojisi yıllardır işitme cihazı endüstrisinde kullanılmaktadır. Bu cihazlar basit haliyle ses kaynağına bağlı vericiden oluşmaktadır. Bu vericiler sesi işitme cihazlarındaki alıcılara wireless sinyalleri ile gönderirler. Bu sistemler sınıf ortamında kullanım için uygundur ancak başlı başına bireysel kullanım için uygun değildirler. Sistemin maliyeti, diğer kullanıcılar için ayrıca aksesuar gerektirmesi sinyal kayıpları gibi dezavantajları bulunmaktadır (Edwards, 2007).

İşitme cihazı endüstrisinin şimdilerde konuştuğu ve uyguladığı yeni teknoloji dijital wireless teknolojisidir. Wireless teknolojilerinin elektronik cihazlarda yaygınlaşması ile bu teknolojinin önemi de artmaktadır. Bu teknoloji digital sinyal işleme cihazlarının tüm özelliklerini kapsadığı gibi, daha gelişmiş birçok özellik de sağlamaktadır. Dijital wireless işitme cihazları doğrudan elektronik cihazlardaki wireless sistemlerine bağlanırlar. Televizyon, cep telefonu, müzik sistemlerinde kullanılabilirler. Cihaz ile doğrudan etkileşime geçtiği için dışarıdan gelebilecek gürültüleri tamamen giderir ve kaliteli bir duyum sağlar. Sistem otomatik olarak diğer wireless cihazlara geçiş yapabilmektedir. Örneğin televizyon izlerken cep telefonunuza arama geldiğinde sistem otomatik olarak hali hazırdaki bağlantıyı keserek cep telefonuna bağlantı sağlar. Dünyada birçok firma dijital wireless işitme cihazı üretmektedir. Cihazlar gün geçtikçe daha da küçülmekte ve güç tüketimleri de azalmaktadır. Cihazların kullanımı ve üretimi arttıkça fiyatının da düşmesi ve ulaşılabilirliğinin artması beklenmektedir (Anonim, 2015a).

## 2. LİTERATÜR BİLDİRİŞLERİ

Bu bölümde işitme engeline veya işitme yetersizliğine sahip bireyleri teknolojik olarak destekleyen çözümlerden bahsedilmiş, geçmişten günümüze kullanılan teknolojiler ve yapılmış çalışmalar hakkında bilgiler derlenip sunulmuştur. Fiziksel olarak işitmeyi iyileştiren cihazlar veya uygulamalardan ziyade, işitme engeline sahip bireylerin sosyal ve eğitim hayatlarında onları destekleyici ve hayatlarını kolaylaştırıcı, yardımcı teknolojilere yer verilmiştir.

1960'lı yıllarda TTYs (Teletypewriters) cihazlarının keşfiyle, iletişim dünyasında işitme engelliler açısından bir devrim yaşanmıştı. TTYs diye adlandırılan bu telekomünikasyon cihazları, işitme engellilerin birbirleriyle ve işiten bireylerle iletişim kurmasına imkân tanımaktaydı. Bir klavye ve bir görüntüleme ekranından oluşan bu cihazlar aranan kişiyle mesaj yoluyla iletişim kurulması yöntemiyle çalışmaktaydılar. TTYs cihazları daha sonraları sabit telefonlara bağlanarak daha da geliştirilmiş ve işitme engellilerin işiten bireyleri arayabilmesine de olanak sağlamışlardı. Bu teknoloji işitme engelliler açısından günümüz teknolojilerine giden yolda atılmış en yenilikçi adım olmasının yanında, sadece telekomünikasyon değil, eğitim ve iş yaşamı gibi hayatın diğer alanlarında da işitme engellilere büyük katkılar sağlamıştı (Lang, 2000).

Amerika'da 1900'lü yıllarda popülerleşmeye başlayan film endüstrisinde, sessiz filmler işitme engeli olsun olmasın tüm insanlar tarafından ilgiyle izlenmekteydi. Özellikle işitme engeline sahip bireylerin toplumla bütünleşmelerinde önemli bir unsurdu. Sessiz filmler altyazıdan ziyade oyuncunun vücut dilini ve yüz mimiklerini kullandığı kısa hikâyelerden oluşan ve daha çok komediye yönelik filmlerdi. Teknolojinin gelişmesi ve sesli filmlerin üretilmeye başlaması ile sessiz filmlerin sonu gelmiş oldu. İşitme engelliler, hareketten çok diyalog üzerine kurulu bu yeni nesil film anlayışıyla birlikte Amerikan toplumunun en popüler sektörü haline gelen film sektöründen dışlanmış oldu (Kovalik, 1992).

İlk olarak 1947'de Kübalı aktör Emerson Romero tarafından ilk alt yazılı filmler diyebileceğimiz, diyalog kartlarından oluşan filmler yayınlandı. Bu filmlerde sahne geçişlerinde diyalog kartları gösterilmekteydi ancak buda film akışını durdurduğu için tam olarak bir izleyici memnuniyeti sağlayamamaktaydı. Ancak bu yöntem Amerikalı

işitme engellilerin modern film endüstrisi ile yeniden bağ kurmasını sağlamış oldu. 1950'lerde gerçek manada kesintisiz altyazı sunulmaya başlandı. Bu gelişmeler üzerine 1958'de Amerikan hükümeti tarafından işitme engellilere yönelik altyazılı filmler oluşturulmasına yönelik bir yasa çıkarılması ile bu uygulamalar yaygınlaşmaya başladı (Kovalik, 1992).

1979 yılında altyazı teknolojisinin gelişimi ile birlikte dünyada ilk defa BBC tarafından bir televizyon kanalında Ceefax uygulaması ile altyazı hizmeti sunulmaya başlandı. Bu gelişme ile birlikte birçok televizyon kanalı altyazı hizmeti sunmaya başladı. İlerleyen yıllarda Avrupa'da birçok ülkede altyazılı televizyon programları oldukça yaygınlaştı (Zárate, 2010).

NCI (National Captioning Institute) ve TSI (Translation System Inc.) 1980 yılında InstaText adını verdikleri sistem ile stenografi makinesini kullanan stenograflarla canlı yayını altyazıya dönüştürmekteydiler. Stenografi makineleri kullanıcısının tecrübesine bağlı olarak konuşulanları dakikada 180 ve üzerinde kelimeye kadar yazabilen, kendine ait kısaltılmış ifadeleri ve basit işaretleri ile söylenenleri söylendiği kadar hızlı yazmaya yarayan makinelerdir. Bu sistemde çevirisi yapılan konuşmaların televizyona aktarılması 4 ila 7 saniye sürmekteydi. 1990'lara gelindiğinde NCI haftada 60 saatlik canlı altyazı sunabilir haldeydi (Downey, 2008).

Pražák ve ark. (2012), Çek Cumhuriyetinde bir televizyon kanalı için geliştirdikleri ve test aşamasında olan canlı yayınlara yönelik anlık altyazı oluşturma sistemden bahsetmişlerdir. Bu kapsamda; kelime doğrulama sistemi, sesi yazıya çevirici sistemler ve tekrar konuşma diye tabir edilen, videolardaki diyalogların bir konuşmacı tarafından tekrar edilip sisteme kaydedilmesinden oluşan bir sistem hakkında detaylı bilgiler verilmiştir. Proje kapsamında canlı yayınlara yönelik otomatik altyazı sistemi uygulaması bir ekonomi programı için test edilmiş ve sonuçları incelenmiştir. Buna göre; sistemin dezavantajlarının çeviride yaşanan gecikmeler, sistemin kelime haznesinin yetersizliğinden kaynaklanan sorunlar, oluşan hataların düzeltilememesi olduğu gözlenmiştir. Yapılan çalışmada videoların tekrardan konuşmacılar tarafından bir stüdyo ortamında seslendirildiği sistem en makul çözüm olarak gösterilmektedir.

Aran ve Akarun (2007), işitme engellilere yönelik etkileşimli bir işaret dili eğitim sistemi geliştirmişlerdir. Sistem bilgisayar ortamında işaret dilinin tanımlanabilmesine yönelik çalışmaktadır. Sistemin veri tabanına önceden birkaç el hareketi kayıt edilmiş durumdadır. Ele takılan renkli bir eldiven ile kamera karşısına geçilir ve sisteme daha önceden işaret dili ile kaydedilmiş kelime el hareketleri ile yapılmaya çalışılır. Kullanıcı hareketi yapmadan önce hareketin nasıl yapıldığı izleyebilir ve hareketi yaptıktan sonra yapmış olduğu hareketin karikatürize edilmiş halini izleyerek hatalarını inceleyip düzeltebilir. Sistem yapılan hareketi analiz ederek veri tabanındaki herhangi bir kayıt ile eşleşip eşleşmediğini kontrol etmektedir. Eğer eşleşiyorsa kullanıcıya hareketin doğru yapıldığı bilgisini vermektedir. Sistem işitme engellilere işaret dilini öğrenmede destek verebileceği gibi, işaret dilini öğrenmek isteyenler içinde bir kılavuz niteliğindedir.

Uluslararası araştırma guruplarından biri olan Kanada'da bulunan Liberated Learning Consortium, 2010 ve 2012 yılları arasında Konuşma Tanıma (Speech Recognition) altyapılı sistemler üzerine çalışmalar yürütmüşlerdir. Geliştirilen uygulamalarda IBM Speech Recognition teknolojisini kullanmışlardır. Kanada'da birçok üniversite ve engelliler üzerine araştırma yapan kuruluşlar ile işbirliği yaparak işitme engellilere yönelik uygulamalar geliştirmişlerdir. HTS (Hosted Transcription Service) adını verdikleri daha önceden kaydedilmiş ses dosyasını metne çevirici bir uygulama geliştirmişlerdir. Bu uygulama kayıtlı ses dosyalarını masaüstü bir uygulama ile metne çevirmektedir (Anonim, 2015c).

Amerika'da 2014 yılında bir grup mobil uygulama geliştirici tarafından, işitme engeli veya işitme güçlüğü çeken insanlar için bir mobil uygulama geliştirilmiştir. Transcense adlı bu uygulama ile birçok insanın bir arada bulunduğu bir ortamda, konuşulanların mobil telefonlar üzerinden anlık takip edilip, karşılıklı diyalogların gerçekleştirilebilmesi sağlanmıştır. Her konuşmacı sisteme kendi mobil telefonu ile bağlanmaktadır, konuşmacıların her birine farklı renkler atanmış ve bu sayede konuşmanın kim tarafından yapıldığı daha anlaşılır hale getirilmiştir. Konuşmacılardan alınan sesler mobil uygulama üzerinde metne çevrilmekte ve konuşmaları takip etmek isteyen kullanıcıya ulaştırılmaktadır. Uygulama konuşmayı takip etmek isteyenler için ücretli olup, konuşmacılara ücretsiz olarak sunulmaktadır (Anonim, 2015d).

Nuance firması Dragon markası ile birlikte tüm dünyada konuşma tanıma sistemleri arasında önemli bir yere sahiptir. Firma bireysel müşterilere ve şirketlere yönelik ücretli birçok hizmet sunmaktadır. Güçlü bir ses tanıma yazılımına sahip firma, bu altyapısını kullanarak sesi yazıya dönüştürmenin yanında, sesin kullanıldığı çeşitli uygulamalar geliştirmektedir. Ses komutlarını algılayan Dragon Assistant yazılımı masaüstü bilgisayarların yönetilebilmesini sağlamaktadır. Dragon Dictation sesinizi yazıya çevirerek, mobil cihazlardan SMS ve e-mail gönderme, sosyal paylaşım uygulamaları üzerinden paylaşım yapma ve kişisel notlarınızı tutmanızı sağlamaktadır. Dragon Mobil Assistant mobil telefonlarınızı sesiniz ile yönetebilmenizi sağlamaktadır. Yazılım gerçekleştirmek istediğiniz farklı işlemler için farklı komutların eklenebilmesini sağlayabilmektedir. Firma ses tanıma altyapısını tüm dünyadaki yazılım geliştiricileri ile paylaşabilmekte ve uygulama geliştiricilerine bu altyapıyı kullanarak çeşitli uygulamalar geliştirme imkânı sağlamaktadır (Anonim, 2014).



### 3. MATERYAL VE YÖNTEM

#### 3.1. Kullanılan Yazılım Teknolojileri ve Ortamları

Bu tez kapsamında geliştirilen uygulamada istenilen sonuca ulaşmak için birbirinden farklı birçok yazılım dili kullanılmıştır. Hem mobil hem de web tabanlı iki farklı uygulama geliştirilmiştir. Bu bölümde uygulama geliştirme sürecinde kullanılan donanımlar ve yazılımlar açıklanmış olup, tezde nasıl kullanıldıkları hakkında detaylı bilgiler verilmiştir.

##### **Uygulama geliştiriminde kullanılan donanımlar:**

- Windows 7 işletim sistemi yüklü bir Notebook
- Linux işletim sistemi yüklü bir sunucu
- Android işletim sistemi yüklü bir cep telefonu
- Adroid işletim sistemi yüklü bir tablet bilgisayar
- Projektör
- Mikrofon

##### **Uygulama geliştirme ortamları ve yazılımlar:**

- Android Studio: Mobil uygulama geliştirme platformu
- Adobe Dreamweaver CS6: Web uygulama geliştirme platformu
- Adobe Photoshop CS6: Uygulama ara yüz ve ikon tasarımları
- JAVA: Android uygulamanın geliştiriminde kullanılan yazılım dili
- PHP: Web uygulamasının geliştiriminde kullanılan yazılım dili
- MySQL: Uygulama veri tabanı sunucusu
- SQL: Veri tabanı işleme dili
- AJAX: Web uygulamasında kullanılan asenkron veri işlemleri
- HTML&CSS: Web sayfaları ve stil öğelerinin oluşturulması

### 3.1.1. Android işletim sistemi

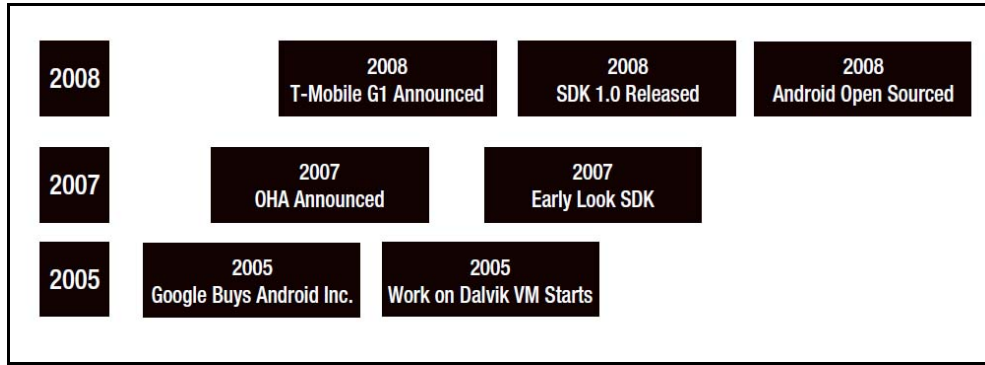
Android, yaratıcılarından biri olan Andy Rubin tarafından şöyle tanımlanmaktadır: Mobil cihazlar için gerçek anlamda ilk açık ve kapsamlı bir platformdur. İşletim sistemi, kullanıcı ara yüzleri ve uygulamalardan oluşur. Mobil değişimi engelleyen telif haklarından uzak, tamamen mobile yönelik bir yazılımdır.

Android işletim sistemi dünya üzerinde 80'den fazla firmanın katılımıyla oluşturulan ve Open Handset Alliance (OHA) adı verilen topluluk tarafından sürekli olarak geliştirilmektedir. OHA'nın amacı dünya üzerindeki tüm mobil cihaz kullanıcılarına, yazılım geliştiricilerine ve cihaz üreticilerine mümkün olan en iyi ürünü lisans bedeli olmadan sunmak ve var olan mobil işletim sistemleri arasında en hızlı ve en kaliteli ürünü oluşturmaktır (Meier, 2012).

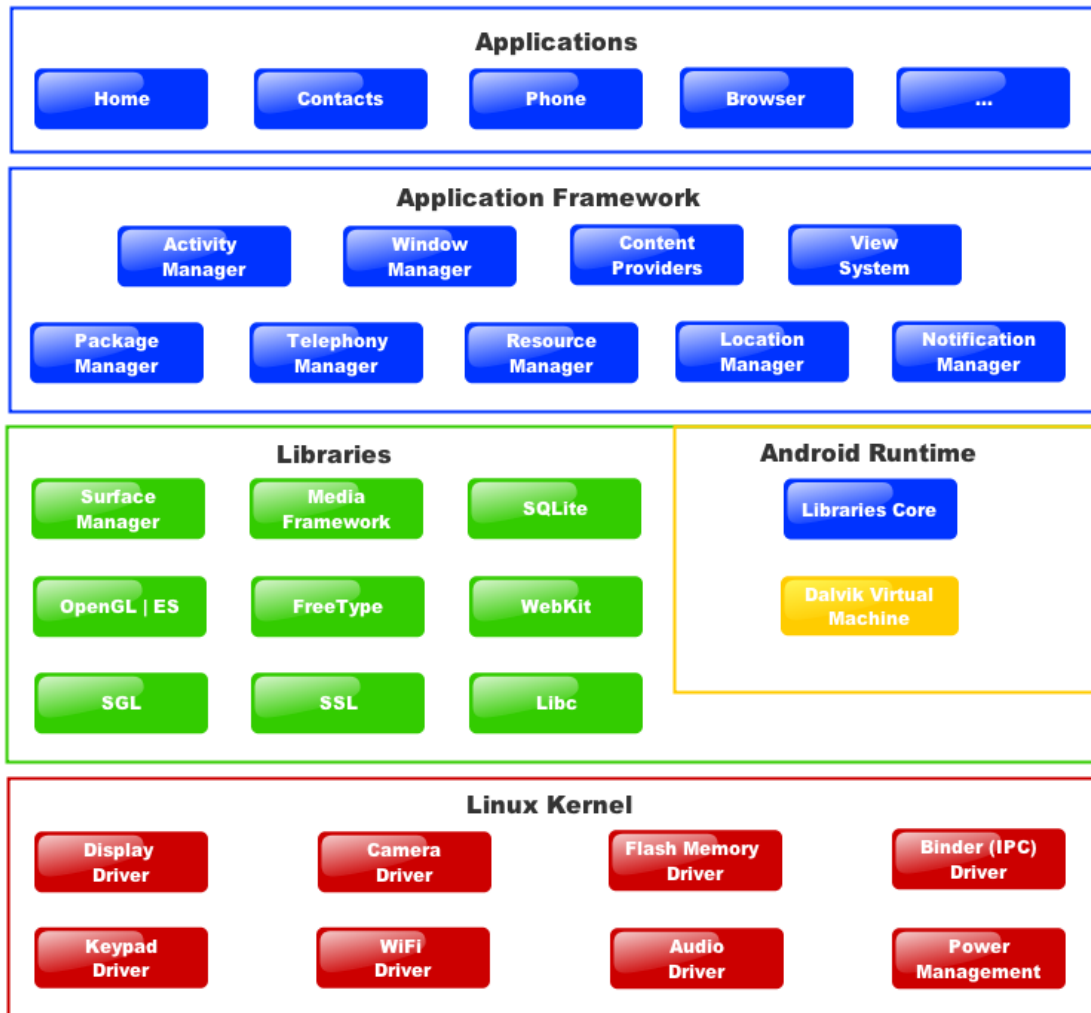
Mobil telefon üreten firmalar ürettikleri telefonlarda birçok farklı işletim sistemi kullanmaktadırlar. Symbian, Windows Mobile, Mobile Linux, iPhone, Moblin bu işletim sistemlerine örnek olarak gösterilebilir. Her işletim sisteminin kendine ait kısıtlamaları ve çalışma yapıları olduğu için mobil uygulama geliştiriciler için standart bir geliştirme platformu sunulamamaktaydı ve bu durumda mobil uygulama geliştiriciliğinde büyük sıkıntılara sebep olmaktaydı. Bu durumun aksine Android kolay, açık kaynak kodlu ve standart bir platform vadediyordu.

Andy Rubin'in önderliğindeki bir grup yazılımcı tarafından Android adlı şirket kuruldu. Ardından 2005 yılında Google bu şirketi satın aldı. 2007'nin sonlarına doğru mobil endüstri liderleri bir araya gelerek yukarıda bahsedilen OHA'yı oluşturdular. Bu topluluğa üye firmalar ürettikleri cihazlarda Android işletim sistemi kullanımı için herhangi bir lisans bedeli ödememekteydiler. İlk Android SDK (Software Development Kit) Kasım ayında ortaya çıkarıldı. Ardından Eylül 2008'de Android platformunu kullanan ilk mobil telefon T-Mobile G1 üretildi. Günler sonra Google Android SDK 1.0'ı yayınladı. Ekim 2008'de Android açık kaynak kodlu bir işletim sistemi oldu. (Hashimi ve ark., 2010). Android zaman çizelgesi Şekil 3.1'de verilmiştir.

Android işletim sistemi, farklı katmanlardan oluşan bir yazılımdır. Her katmanda farklı servisler bulunmaktadır. Tüm bunlar bir araya gelerek işletim sistemini ve uygulamaları oluşturur. Şekil 3.2'de Android mimarisi gösterilmiştir.



Şekil 3.1. Android zaman çizelgesi (Hashimi ve ark., 2010).



Şekil 3.2. Android mimarisi (John, P. S., 2015).

**Linux Kernel:**

Bu katman işletim sistemi ile donanım arasındaki bağlantıyı sağlar. En alt katmanda bulunur. İşletim sisteminin kalbidir. Linux-Kernel, Linux diye adlandırılıyor olsa da gerçekte Linux kernelden bağımsız olarak Android için özel olarak geliştirilmiştir.

**Libraries:**

Bu katman Android'in kendi kütüphanelerinin olduğu katmandır. Farklı türdeki verilerin cihaz tarafından kullanılmasını sağlar. C veya C++ ile yazılan bu kütüphaneler belirli donanımlara özgüdür.

**Android Runtime:**

Bu katman Dalvik Sanal Makinesi ve çekirdek Java kütüphanelerinden oluşur. Android işletim sistemi üzerine dahil edilen bu kütüphane ile uygulamalar içerisinde Java kütüphanelerini kullanabilmekteyiz. Dalvik Sanal Makinesi Java ile yazılmış kodları Android işletim sisteminde çalışması için gereken formata dönüştürür.

**Application Framework:**

Bu katman uygulamalarla direkt olarak iletişimde olan katmandır. Telefonun kaynak yönetimi, sesli arama, GPS, uygulamalar arasında veri paylaşımı gibi özelliklerinin uygulamalar tarafından kullanılabilmesini sağlar.

**Applications:**

Uygulamalar Android mimarisinin en üst katmanını oluşturur. Android işletim sistemi kurulduğunda bazı uygulamalar standart olarak gelir. Bunlar; SMS uygulaması, web tarayıcı, rehber ve arama gibi uygulamalardır.

Bu çalışmada; Mobil Android uygulama geliştirmesi yapılabilmesi için, Google tarafından geliştirilen içerisinde Java Development Kit (JDK), Andoroid Virtual Device (AVD) ve mobil uygulama için gerekli ortam ayarlarının oluşturulduğu Android Studio mobil uygulama geliştirme ortamı kurulmuştur. Ayrıca, gerekli API kütüphaneleri,

geliştirme ve test araçlarının bulunduğu Android Software Development Kit (SDK) kurulmuştur.

### 3.1.2. Sistemin veri tabanı

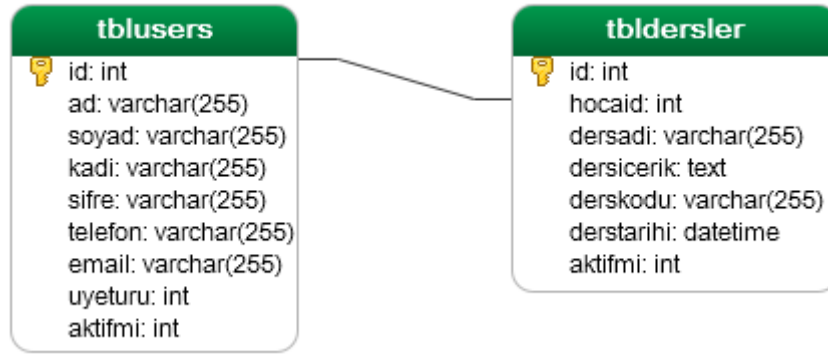
Uygulama geliştirilirken veri tabanı yönetim sistemi olarak MySQL kullanılmıştır. Ticari olmayan uygulamalar için herhangi bir lisans bedeli olmaması, açık kaynak kodlu olması, hızlı olması ve dünya genelinde yaygın olarak tercih edilmesi bu seçimde etkili olmuştur. Veri tabanı Linux işletim sistemine sahip bir sunucuya kurulmuştur. Aynı zamanda bu sunucu web uygulamasının geliştirildiği sunucudur. Linux işletim sistemi ve MySQL veri tabanı yönetim sistemi birbiri ile tam uyumlu olarak çalışmaktadır.

MySQL, altı milyondan fazla sistemde yüklü çok kullanıcılı hızlı ve sağlam bir veri tabanı yönetim sistemidir. UNIX, OS/2 ve Windows sürümleri mevcuttur ve ücretsiz dağıtılmaktadır. Linux sistemde daha hızlıdır. Web sunucularında en çok tercih edilen veri tabanıdır. Birçok veri tabanı yönetim sisteminde var olan veri tutarlılığı geliştiriciler için esnek olmayan zorlayıcı bir özellik olarak görülmektedir, ancak MySQL veri tutarlılığını sağlama işini geliştiriciye bırakmıştır. Bu da geliştiriciler tarafından tercih edilmesinin nedenlerinden biridir. Veri tabanı yönetimi için grafiksel ara yüze sahip uygulamalar mevcuttur. MySQL'i geliştiren firma tarafından sunulan MySQL GUI Tools ve PHP ile geliştirilmiş phpMyAdmin bunlardan bir kaçıdır (Anonim, 2015b).

Sesli Dersi Yazılı Takip uygulaması biri mobil ve biride web olmak üzere iki farklı yazılımdan oluşmaktadır. Uygulamada Linux bir sunucu üzerinde tutulan ortak bir veri tabanı kullanılmış ve bu iki yazılımın birbirleri ile veri alışverişi yapması sağlanmıştır. Veri tabanı oluşturulurken uygulama gereksinimleri göz önüne alınmış, işlevsel ancak minimal düzeyde bir veri tabanı oluşturulması hedeflenmiştir. Geliştirilen uygulamada anlatıcı ve takipçi olmak üzere iki farklı kullanıcı tipi bulunmaktadır. Kullanıcı tiplerine göre yetkiler değişmektedir.

Anlatıcı, Android uygulama üzerinden sistemde kayıtlı bir dersi seçip anlatabilmekte ya da yeni bir ders oluşturup anlatabilmektedir. Ders yönetim sistemi

üzerinde yeni ders ekleme, düzenleme, silme yetkilerine sahiptir. Takipçi Android işletim sistemi üzerinde ders takip etme yetkisine sahiptir. Ders yönetim sistemi üzerinde ise sitemde var olan tüm dersleri görüntüleme ve indirme yetkisine sahiptir. Sistemin veri tabanında iki adet tablo bulunmaktadır. Bunlar; Derslerin tutulduğu **tbldersler** ve kullanıcıların tutulduğu **tblusers** tablolarıdır. Tablo nitelikleri ve ilişki şeması Şekil 3.3’de verilmiştir.



Şekil 3.3. Tablo nitelikleri ve ilişki şeması.

**Kullanıcı bilgilerinin tutulduğu tblusers tablosundaki nitelikler şu şekildedir:**

- id: Benzersiz bir değerdir ve her kullanıcı için farklıdır.
- ad: Kullanıcı adının tutulduğu alandır.
- soyad: Kullanıcı soyadının tutulduğu alandır.
- kadi: Sisteme giriş yapmak için gerekli kullanıcı adının tutulduğu alandır.
- sifre: Sisteme giriş yapmak için gerekli sifrenin tutulduğu alandır.
- telefon: Kullanıcının telefonunun tutulduğu alandır.
- email: Kullanıcı e-postasının tutulduğu alandır.
- uyeturu: Kullanıcı türünün tutulduğu alandır.
- aktifmi: Kullanıcının aktif olup olmadığı bilgisinin tutulduğu alandır.

**Ders bilgilerinin tutulduğu tbldersler tablosundaki nitelikler şu şekildedir:**

- id: Benzersiz bir değerdir ve her kullanıcı için farklıdır.
- hocaid: Ders ile ilişkili hoca'nın id'sinin tutulduğu alandır.

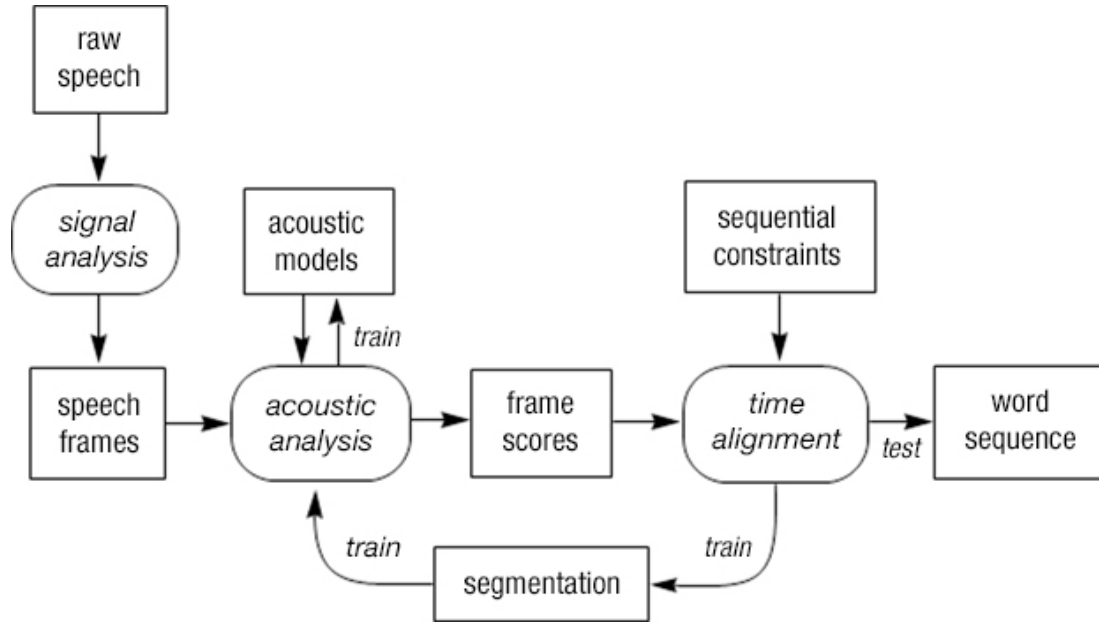
- dersadi: Ders adının tutulduğu alandır.
- dersicerik: Anlatılan dersin içeriğinin tutulduğu alandır.
- derskodu: Dersi takip etmek için gerekli kodun tutulduğu alandır.
- derstarihi: Dersin yapıldığı tarihin tutulduğu alandır.
- aktifmi: Dersin aktif olup olmadığı bilgisinin tutulduğu alandır.

Kullanıcıların tutulduğu tblusers tablosundaki uyeturu niteliği “1” yada “2” değerlerini almaktadır. “1” değeri kullanıcının anlatıcı olduğunu göstermektedir. “2” değeri ise kullanıcının takipçi olduğunu göstermektedir. Aktifmi niteliği kullanıcının sistemi kullanmaya yetkisinin olup olmadığını göstermektedir. Eğer değer “0” ise kullanıcı sistemi kullanamaz, “1” ise kullanıcı sistemi kullanmaya yetkilidir anlamına gelmektedir.

Derslerin tutulduğu tbldersler tablosundaki hocaaid niteliği tblusers tablosundaki id alanı ile ilişkilendirilmiştir ve ders-hoca ilişkisi sağlanmıştır. derskodu niteliği canlı dersi takip kodunu içerir. Bu kod canlı dersi takip etmek isteyenlerin girmesi gereken ve sistem tarafından otomatik olarak üretilen benzersiz iki haneli bir koddur. Anlatıcı yetkisine sahip kullanıcı mobil uygulama üzerinden sistemde olmayan yeni bir ders anlatacağı zaman sistem tarafından otomatik olarak üretilir. Sistem üzerinden anlatıcı yetkisine sahip kullanıcı bu kodu değiştirme yetkisine sahiptir. Aktifmi niteliği dersin paylaşımda olup olmadığı bilgisini tutar. Eğer değer “0” ise ders kullanıcılara gösterilmez, “1” ise kullanıcılar dersi görüntüleyebilir.

### 3.1.3. Konuşmanın yazıya çevrilmesi (Speech-to-Text)

Ses tanıma (Speech Recognition) çok katmanlı bir örüntü tanıma işlevidir. Akustik sinyaller belli bir hiyerarşi içerisinde işlenerek, kelimeler, ifadeler ve cümleler oluşturulur. Şekil 3.4'te Standart ses tanıma sisteminin yapısı gösterilmiştir.



Şekil 3.4. Standart ses tanıma sistemi yapısı (Tebelskis, 1995).

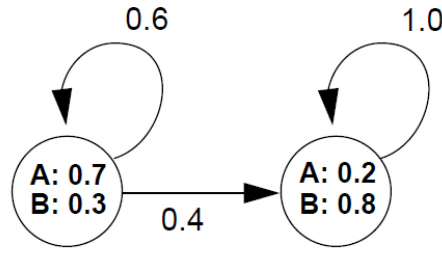
Bu katmanlardan her birinde bazı geçici kısıtlamalar oluşabilir. Telaffuz problemleri veya yanlış dilbilgisi kullanımı, bu kısıtlamalara örnek olarak verilebilir. Bu tür hatalar hiyerarşideki düşük seviyeli katmanlarda istatistiksel olarak çözümlenebilir. Üst katmanlarda ise tahmin etme üzerine kurulu algoritmalar kullanılarak ses tanıma işlemleri gerçekleştirilmektedir (Tebelskis, 1995).

Ses tanıma sistemini birkaç ana bölümde inceleyebiliriz. Bunlar; özellik çıkarımı, daha önceden test edilip doğrulamaları yapılmış verilerin tutulduğu akustik model veri tabanı, sözlük, dil modeli ve ses tanıma algoritmalarıdır. Ses tanıma sistemde ilk olarak analog ses sinyallerinin genlik ekseninde olması ya da dijital olarak örneklenmiş olması gerekmektedir. Ses sinyalleri örnekleri eşit aralıklarla analiz edilir. Bu aralık genellikler 20 ms dir çünkü bu aralıkta sinyal sabit kabul edilir. Konuşma özellik çıkarımı konuşma karakteri vektörlerinin eşit aralıklı olarak oluşmasını sağlar.



Veri tabanındaki vektörler akustik model parametrelerinin tahmininde kullanılır. Akustik model tanımlanacak temel unsurların özelliklerini açıklar. Temel unsurlardan kasıt sürekli bir konuşmadaki ses birimi veya ayrı olarak bir kelime tanımlama için gerekli ses birimi olabilir.

Sözlük kelime ile akustik modeli karşılaştırmak ve örtüştürmek için kullanılır. Dil modeli, dilin kurallarına ve diğer metinlerden elde edilen istatistiki bilgilere göre öngörülebilir kelime kombinasyonlarını azaltmak için kullanılır. Ses tanıma sisteminde kullanılan algoritmalarından olan HMM (Hidden Markov Model) algoritması modern teknolojilerde en çok kullanılan algoritmadır. Bu algoritma modelleme için sözcük ya da ses birimi kullanmaktadır. Ses tanıma sistemleri, genellikle ses sinyallerini sıralı bir ya da daha fazla sembolün bir araya gelmesinden oluşan şifrelenmiş mesajlar olarak kabul ederler. İki durumlu ve iki çıkışlı basit bir Hidden Markov Modeli Şekil 3.5’de verilmiştir.



Şekil 3.5. İki durumlu ve iki çıkışlı basit bir Hidden Markov Modeli (Tebelskis, 1995).

Hidden Markov Modeli birbirine bağlı durumlar topluluğundan oluşur. Belirli bir başlangıç durumuyla başlar. Her bir zamanda geçiş yeni bir duruma doğru yapılır. Bu yeni durumda bir çıkış sembolü oluşur. Çıkış sembolleri ve geçişler rastgeledir ve olasılık dağılımları tarafından yönetilir. HMM zamanla üretilen çıkış sembolleri dizisinin izlenebildiği bir kara kutu olarak düşünülebilir. Ancak zamanla durum dizilerinin nasıl hareket ettiğini gözlemlemek mümkün değildir. Bu nedene isminin başına Hidden (Gizli) ibaresini almıştır. HMM çeşitli uygulamalara sahiptir. Bu algoritma ses tanıma için uyarlandığında algoritmadaki her durum akustik bir model olarak yorumlanır. Bu akustik modeller her konuşma dilimi içerisindeki sesin olması muhtemel karşılıkları durumudur (Young ve Ark., 2006).

Bu tez çalışmasında Google tarafından Android uygulama geliştiricilerine sunulan Speech-to-Text servisi kullanılmaktadır. Servisin kullanılması için Google tarafından oluşturulmuş yazılım kütüphaneleri kullanılmış bir dizi işlem sırasıyla uygulanmıştır. Tez kapsamında çevirisi yapılan sesin amaca uygun olarak uzak sunucuya anlık olarak gönderilebilmesi ve ders yönetim sisteminde işlenebilmesi için bir takım ek fonksiyonlar ve sınıflar yazılmıştır (Reddy ve Mahender, 2008).

Android yazılım geliştiriminde uygulamanın cihaz üzerinde hangi izinlere sahip olacağı AndroidManifest.xml dosyası üzerinde tanımlanır. Bu tezde kullanılan Speech-to-Text servisi Google sunucuları üzerindedir bu nedenle uygulamanın bu sunucuya bağlanması için öncelikle internet izni verilmelidir. Şekil 3.6.'de bu izinlerin nasıl tanımlandığı gösterilmiştir.

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

Şekil 3.6. İnternet izinlerinin verilmesi.

Google tarafından Android geliştiricilerin kullanımına sunulan Speech-to-Text işlemleri için gerekli kütüphanelerin bulunduğu sınıf Şekil 3.7.'deki gibi tanımlanmıştır.

```
import android.speech.RecognizerIntent;
```

Şekil 3.7. RecognizerIntent sınıfının import edilmesi.

Uygulama üzerinden metne çevrilmesi için Google sunucularına gönderilecek olan ses değişmeyen statik bir kod ile gider ve bu kod üzerinden sunucu üzerinde bir aktivite açılır ve sonuç gönderilir. Şekil 3.8.'de bu kodun nasıl tanımlandığı gösterilmiştir.

```
final int KodKanali=100;
```

Şekil 3.8. Kod kanalının tanımlanması.

Konuşma ekranındaki işlev butonlarının tanımlandığı ara yüze ait konuşmaekrani.xml kodu Şekil 3.9'da verilmiştir. Bu butonlar HocaEkranı.java sınıfında setOnClickListener yapısı ile tetiklenmiş ve konuşmayı başlatmaya, konuşmayı bitirmeye ve konuşmayı temizlemeye yarayan fonksiyonlar çalıştırılmıştır. Speech-to-Text ekranının tanımlanması ve ayarlanması Şekil 3.10'da verilmiştir. Diğer fonksiyonlar Ek-1'de verilmiştir.

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    android:gravity="center">
        <Button
            android:id="@+id/keBTNBaslat"
            android:layout_width="100dp"
            android:layout_height="60dp"
            android:text="Başlat"
            android:gravity="center"/>
        <Button
            android:id="@+id/keBTNBitir"
            android:layout_width="100dp"
            android:layout_height="60dp"
            android:text="Bitir"
            android:gravity="center"/>
        <Button
            android:id="@+id/keBTNTumunuTemizle"
            android:layout_width="100dp"
            android:layout_height="60dp"
            android:text="Temizle"
            android:gravity="center"/>
    </LinearLayout>

```

Şekil 3.9. Konuşma ekranındaki işlev butonlarının ara yüzde tanımlanması.

RecognizerIntent sınıfının birçok özelliği bulunmaktadır. Geliştirilmiş uygulamada yapılmasını istediğimiz işlere göre bu özellikler kullanılmaktadır. Çizelge 3.1.'de ReconizerIntent sınıfının projede kullanılan özellikleri açıklamaları ile birlikte verilmiştir.

Çizelge 3.1. RecognizerIntent sınıfının projede kullanılan özellikleri

| Özellik                                           | Açıklama                                                         |
|---------------------------------------------------|------------------------------------------------------------------|
| ACTION_RECOGNIZE_SPEECH                           | Konuşma penceresini açar.                                        |
| EXTRA_LANGUAGE_MODEL                              | Hangi konuşma modelinin kullanılacağını tanımlar.                |
| LANGUAGE_MODEL_FREE_FORM                          | Serbest konuşmaya dayalı bir modelin kullanılacağını tanımlar.   |
| EXTRA_LANGUAGE                                    | Dil etiketini tanımlar.                                          |
| EXTRA_PROMPT                                      | Kullanıcıya konuşmasını belirten bir diyalog penceresi tanımlar. |
| EXTRA_SPEECH_INPUT_COMPLETE_SILENCE_LENGTH_MILLIS | Sesin işlenmeden önce kabul edilen sessizlik süresi.             |
| EXTRA_SPEECH_INPUT_MINIMUM_LENGTH_MILLIS          | Bir sözcüğün minimum uzunluğunu tanımlar.                        |
| EXTRA_MAX_RESULTS                                 | Opsiyonel maksimum sonuç limiti                                  |

```

private void Basla() {
    Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra
(R.recognizerintent.extra_language_model, RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault());
    intent.putExtra(RecognizerIntent.EXTRA_PROMPT, getString
(R.string.kekonusmakomutu));
    intent.putExtra
(R.recognizerintent.extra_speech_input_complete_silence_length_millis, 5000);
    intent.putExtra(RecognizerIntent.EXTRA_SPEECH_INPUT_MINIMUM_LENGTH_MILLIS, new
Long(20000));
    intent.putExtra
(R.recognizerintent.extra_speech_input_possibly_complete_silence_length_millis,
5000);
    intent.putExtra(RecognizerIntent.EXTRA_MAX_RESULTS, 3);
    try {
        startActivityForResult(intent, KodKanali);
    } catch (ActivityNotFoundException a) {
        Toast.makeText(getApplicationContext(), a.getMessage().toString(),
            Toast.LENGTH_SHORT).show();
    }
}

```

Şekil 3.10. Speech-to-Text ekranının tanımlanması ve ayarlarının yapılması.

Özellikler tanımlandıktan sonra `startActivityForResult` komutuyla belirlenen özelliklerde bir Speect-to-Text konuşma penceresi aktivitesi açılmaktadır. Bu işlemler Basla adındaki fonksiyon içerisinde gerçekleştirilmektedir.

Ses, metne çevrilmek üzere sunucuya ulaştıktan sonra sunucu gerekli çevrimleri yapar ve sonuçları tekrar istemciye ulaştırmak üzere hazırlar. Şekil 3.11.'de gösterildiği şekilde istemci `onActivityResult` metodunu çağırarak bu metinleri sunucudan alır.

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    switch (requestCode){
        case KodKanali:
            if(resultCode==RESULT_OK && null!=data){
                ArrayList<String> sonuc=data.getStringArrayListExtra
                (RecognizerIntent.EXTRA_RESULTS);
                konusma+=sonuc.get(0)+" ";
                konusma = konusma.replace("null","");
                etKonusmalar.setText(konusma);
                KonusmayiVeritabaninaKaydet(konusma);
                Basla();
            }
            break;
    }
}
```

Şekil 3.11. `onActivityResult` metodu ile sunucusundan verilerin alınması.

`onActivityResult` metodunda sunucudan gönderilen istek kodu kontrol edilir. Eğer bu kod ses ile birlikte sunucuya gönderdiğimiz kod ile örtüşüyor ise sonuçlar bir dizi yapısında tutulur. Gerçekleştirilen uygulamada her defasında bu metot çağrılarak metne çevrilmiş konuşmalar veri tabanına gönderilmektedir. Veri tabanına gönderim işlemi Şekil 3.12'de gösterilen fonksiyon ile yapılmaktadır.

```

private void KonusmayiVeritabaninaKaydet(String icerik){
    try {
        if(internet.isConnectingToInternet()){
            Statement query = conn.createStatement();
            boolean sonuc = query.execute("UPDATE tbldersler set dersicerik=' " +
            icerik + "' WHERE id="+gelenveriler.getString("Ders_Id")+" AND hocaId=" +
            gelenveriler.getString("Hoca_Id") + " AND aktifmi=1;");
            if(!sonuc){
                MesajGoster(getApplicationContext(), "Konuşmalarınız veritabanına
                başarıyla kaydedilmiştir.", Gravity.CENTER, 0, 0);
            }
            else{
                MesajGoster(getApplicationContext(), "Konuşmalarınız veritabanına
                kaydedilemedi.", Gravity.CENTER, 0, 0);
            }
        }
        else{
            MesajGoster(getApplicationContext(), getString(R.string.internetyok),
            Gravity.CENTER, 0, 0);
        }
    } catch (SQLException e) {
        MesajGoster(getApplicationContext(), e.getMessage().toString(),
        Gravity.CENTER, 0, 0);
    }
}

```

Şekil 3.12. Konuşmayı veri tabanına kaydetme fonksiyonu.

KonusmayiVeritabaninaKaydet fonksiyonunda parametre olarak onActivityResult metodundan gelen ve konuşma değişkeninde tutulan konuşma metinleri alınır. İnternet bağlantısı durumu kontrol edilir, sonuc değişkenine SQL sorgusu ile konuşma metinleri ilgili tabloya yazdırılır. Sonuç değişkeninin değerine göre kullanıcıya işlemin başarılı olup olmadığı durumu bildirilir.

### 3.1.4. AJAX entegrasyonu

Web sayfaları geleneksel olarak bir dizi HTML sayfasından ve bu sayfaları oluşturan bölümlerden oluşur. Bu sayfaların içeriklerinin bir bölümünün ya da tamamının güncellenebilmesi için bir butona basma suretiyle yeniden yüklenmeleri gerekir. Klasik web sayfalarında sayfanın bir bölümünün güncellenmesi mümkün değildir, her güncelleme işleminde sayfa tümüyle güncellenmektedir. Bunun aksine AJAX (Asynchronous JavaScript Technology and XML) temelli Web 2.0 ile, sayfanın tamamının güncellenmesi yerine sadece istenilen bölümlerini, herhangi bir buton tıklamasına dahi gerek duymadan anlık olarak güncellemek mümkündür. Bu da sayfanın kullandığı bant genişliğini önemli ölçüde azaltmasının yanı sıra yazılımcılara oldukça geniş imkânlar sunmaktadır (Yang ve ark., 2008).

İstemci ve sunucu arasındaki senkron veri alışverişinin getirdiği kısıtlamaları çözmek için geliştiriciler HTTP isteklerinin asenkron yapılabilmesi konusunda zamanla bazı ilerlemeler kaydetmişlerdir. Asenkron veri alışverişi konusunda ilk çözüm Microsoft tarafından IFrame'ler kullanılarak geliştirilmiştir. Internet Explorer üzerinde kullanılan IFrame'ler ile içeriğin asenkron bir şekilde yüklenmesi sağlanmıştır. IFrame'lerden sonra geliştirilen teknoloji XMLHTTP ActiveX Control yapısıdır. Bu teknoloji daha sonra tüm browserler tarafından XMLHttpRequest Object adı altında XMLHttpRequest API'leri ile desteklenmeye başlandı. Bu API'ler HTTP ve HTTPS istekleri ile sunucu-istemci arasında senkron ve asenkron veri alışverişi yapılabilmesini sağlamaktadır. Asenkron veri alışverişi, geliştiricilere sayfalarını çeşitli bölümlere ayırma ve her bölümün kendi içinde bağımsız olarak veri alışverişi yapabilmesini sağlamaktadır. Bu teknoloji sadece talep üzerine veri gönderdiğinden gereksiz yere tüm verinin tek seferde çağrılması ile oluşan fazla hafıza kullanımı, hız yavaşlığı gibi sorunların çözümünün yanı sıra, geliştiricilerin daha interaktif yazılımlar geliştirebilmesini sağlamaktadır.



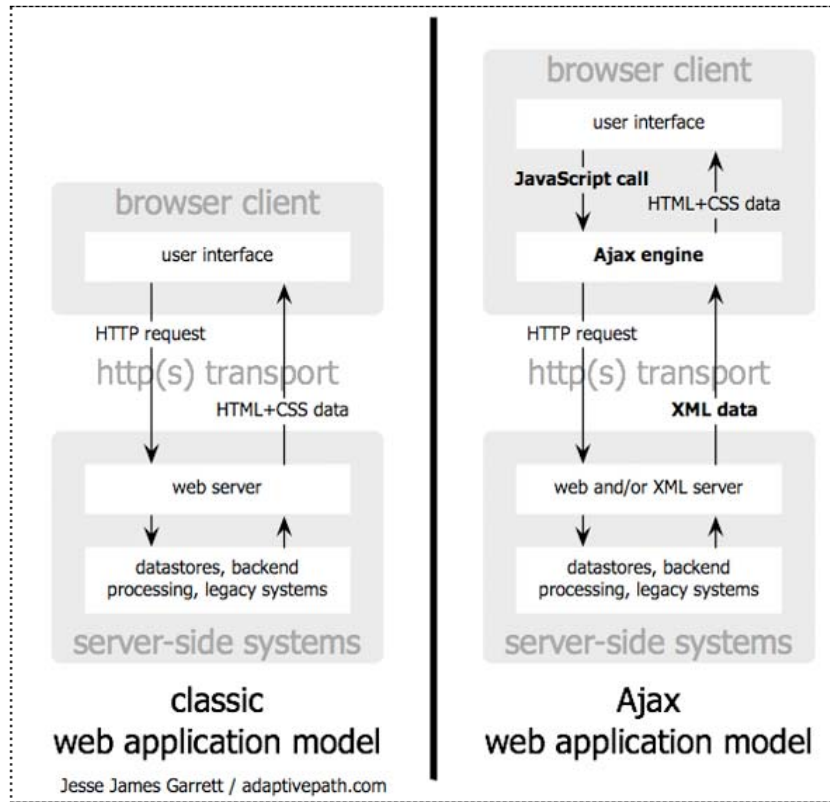
AJAX başlı başına bir teknoloji değildir, daha çok birden fazla teknolojinin kullanıldığı bir yapıdır diyebiliriz. Bu teknolojiler kendi içlerinde gelişimlerine devam ederler ve bir araya geldiklerinde güçlü bir yapı oluştururlar (Garrett, 2005).

### **Ajax Bileşenleri:**

- XHTML ve CSS kullanarak standart veri gösterimi;
- DOM (Document Object Model) kullanılarak dinamik görüntü ve etkileşim;
- XML ve XSLT kullanarak veri değişimi ve manipülasyonu;
- XMLHttpRequest kullanarak asenkron veri alma;
- ve JavaScript kullanarak teknolojileri bir araya getirme.

Klasik web uygulamalarında, HTTP istekleri direk olarak kullanıcıdan sunucuya iletilir ve sunucu bu istekleri yorumladıktan sonra kullanıcıya HTML yapısı ile tekrar geri dönüş sağlamaktadır. Ajax modelinde ise kullanıcı ile sunucu arasında bir Ajax motoru bulunur. Browserda oturum açıldığında ilk olarak JavaScript ile oluşturulan bu motor yüklenir ve sunucu ile kullanıcı arasında asenkron olarak gerekli işlemleri yapmak üzere hazır hale gelir (Yang ve ark., 2008). Klasik yöntemde kullanıcı sunucuya bir istek gönderdiğinde, sunucu isteği yanıtlaya kadar beklemelidir. Sunucu yanıt vermeden kullanıcının bir istek daha gönderme şansı yoktur. Ajax modelinde ise sayfa yüklendiğinde hazır olan Ajax motoru kullanıcı ile etkileşime anlık olarak hazırdır, kullanıcı birden fazla istek gönderdiğinde bu istekler Ajax motoru tarafından karşılanır ve gerekli işlemler sunucuya asenkron bir şekilde gönderilip alınır. Sayfaya fazladan bir Ajax motoru eklemek işi daha karmaşık hale getirebileceği düşünülebilir ancak, gerçek bunun tam tersidir (Garrett, 2005). Klasik web uygulamaları ile Ajax modelinin karşılaştırılması Şekil 3.13’de verilmiştir.





Şekil 3.13. Klasik web uygulamaları ile Ajax modelinin karşılaştırılması (Garrett,2005).

Klasik modeldeki “istek-bekle-cevap” şeklindeki yapıda her kullanıcı aktivitesi için sayfanın yeniden yüklenmesi gerekir. AJAX yapısında ise birden çok istek arka planda asenkron bir şekilde kabul edilerek ön işlemlere tabi tutulur ve gerektiğinde sayfanın belirli bir bölümünü güncellemek için kullanılır. Bu yaklaşım kullanıcılara daha etkileşimli ve hızlı web deneyimleri sunar. AJAX kavramı ortaya çıkmadan önce bu yapı Google Suggest, Google Map ya da Gmail, Flickr, Digg gibi uygulamalarda kendini güçlü bir şekilde hissettirmeye başlamıştı (Mesbah ve van Deursen, 2008).

Tez kapsamında geliştirilen mobil uygulama ile konuşmalar metne çevrilmekte ve anlık olarak uzak bir sunucuya aktarılmaktadır. Uzak sunucudaki bu metinler aynı anda bir bilgisayara veya mobil uygulamaya iletilmektedir. İşte bu noktada AJAX devreye girmekte ve kullanıcı sayfa yenilenmesine gerek kalmadan anlık olarak konuşmaları takip edebilmektedir.

Uygulamada AJAX timer yapısı kullanılarak sonuca ulaşılmıştır. Canlı ders takibi yapılacak sayfada <script> tagları arasında Şekil 3.14.'deki aşağıdaki kodlar yazılmıştır.

```
<script type="text/javascript">
$(document).ready(function() {
$('#fontsize').timer({
    delay: 1000,
    repeat: 99999,
    autostart: false,
    callback: function( index ) {
        var kod = '<?php echo $kod; ?>';
        $.post("ajax_islemler.php",{islem:"icerik_getir",derskod:kod},
        function(data){$("#fontsize").html(data);
        });
    }
});
});
});
</script>
```

Şekil 3.14. Konuşmayı anlık olarak çeken AJAX kodu.

Sayfa yüklendiğinde bir fonksiyon çağrılmıştır. Id'si fontsize olan div içerisinde timer fonksiyonu çağrılmış ve her bir saniyede veri tabanı kontrolü sağlanmıştır. Ders takibi yapmak isteyenlerin ilgili dersin canlı takip sayfasına ulaşabilmesini sağlayan ders kodu Şekil 3.15'deki ajax\_islemler.php sayfasına gönderilmiş ve bu koda göre sayfadan gelecek verilerin ekrana yazılması sağlanmıştır.

```
<?php
require_once("admin/system/ayar.php");
require_once("admin/system/system.php");
if(isset($_POST["derskod"]) and !empty($_POST["derskod"]) and
$_POST["islem"]=="icerik_getir"){
    icerikgetir($_POST["derskod"]);
};
?>
```

Şekil 3.15. ajax\_islemler.php sayfası.

ajax\_islemler.php sayfasında ayar.php ve system.php dosyaları çağrılmıştır. ayar.php’de veri tabanına bağlantı kodları ve metinlerde Türkçe karakter sıkıntısı yaşanmaması için gerekli kodlar yazılmıştır. sytem.php’de ise function.php’ye bağlantı sağlanmış ve Şekil 3.16.’deki icerikgetir fonksiyonu çağrılmıştır.

```
function icerikgetir($kod){
    $sorgu = mysql_query("
    SELECT dersicerik FROM tbldersler WHERE derskodu='$kod'");
    if($sorgu and mysql_num_rows($sorgu)>0){
        while($oku = mysql_fetch_array($sorgu)){
            echo $oku[0];
        }
    }
}
```

Şekil 3.16. İçerik getir fonksiyonu.

İçerik getir fonksiyonu SQL sorgusu ile derse özel oluşturulmuş koda göre dersler tablosundan konuşma içeriklerini getirmektedir. Yukarıda anlatılan tüm işlemlerin sonucunda istemci web sayfasından ve mobil uygulama üzerinden konuşulanlar anlık olarak takip edilebilmektedir.

### 3.2. SDYT (Sesli Dersi Yazılı Takip) Uygulaması

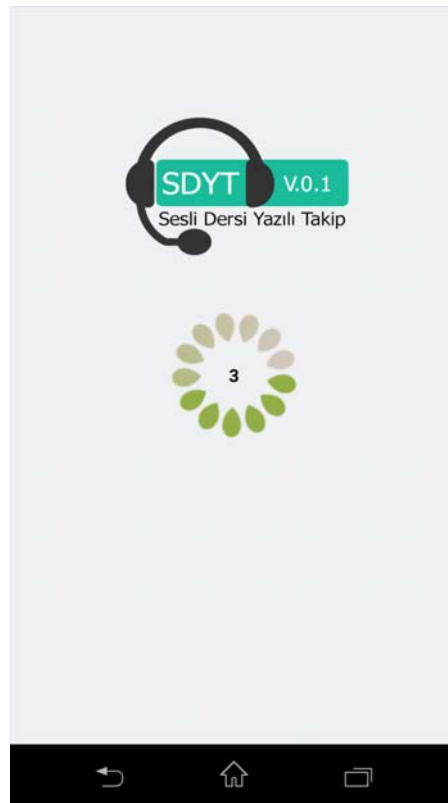
Gerçekleştirilen bu tez çalışması ile temel hedef olan; işitme engellilerin herhangi bir dersi görme duyularını kullanarak takip etmeleri sağlanmıştır. Bu kapsamda ders anlatıcının konuşmaları Android tabanlı bir yazılım ile senkronize bir şekilde yazıya dönüştürülerek, mobil telefonlar üzerinden veya bir projektör ile ekrana yansıtılıp öğrencilerin dersi yazılı olarak takip etmeleri sağlanmıştır. Ayrıca; Tez kapsamında Ders Yönetim Sistemi oluşturulmuş ve bu sistem üzerinden anlatılan derslerin içeriğine ulaşım sağlanabildiği gibi ders ekleme, düzenlenme, silme vb. işlemlerinde gerçekleştirilebilmesi sağlanmıştır. Geliştirilmiş yazılımda birbirine bütünleştirilmiş teknolojilerle sonuca ulaşılmıştır.

Bu tezde gerçekleştirilen proje için hem mobil hem de web tabanlı iki ayrı uygulama geliştirilmiştir. Her iki uygulamada ortak bir veri tabanına bağlanmıştır.

Uygulama temel olarak üç kısımdan oluşmaktadır. Bunlar; Ders Yönetim Bölümü, Ders Anlatma Bölümü ve Ders Takip Bölümleridir. Bu bölümler hakkında detaylı açıklamalar ve ekran görüntüleri ilerleyen kısımlarda ayrıntılı olarak verilmiştir.

### 3.2.1. Android uygulama

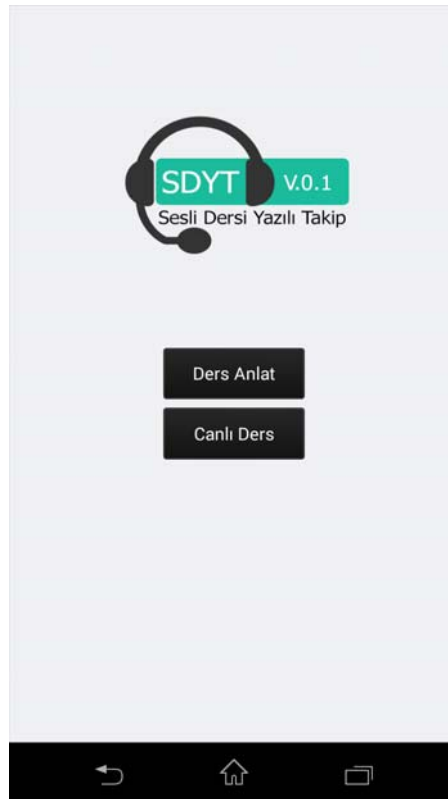
Bu bölümde Android uygulamanın ekran ara yüzlerine ve uygulamanın kullanımı hakkında bilgilere yer verilmiştir. SDYT (Sesli Dersi Yazılı Takip) adı verilen uygulamanın logosu Şekil 3.17’te gösterildiği gibidir. Bu ekran aynı zamanda uygulamanın açılış ekranıdır.



Şekil 3.17 Açılış ekranı.

### 3.2.1.1. Giriş ekranı

Uygulama ara yüzü tasarlanırken kullanıcıyı menü karmaşasından uzak tutmak için kullanıcı dostu bir tasarım gerçekleştirilmeye çalışılmıştır. Bu kapsamda uygulama görselleri ve buton yerleşimleri minimal düzeyde tutulmuştur. Mobil uygulamada iki farklı kullanıcı türü bulunmaktadır. Bunlar dersi anlatan ve dersi dinleyen kullanıcılardır. Kullanıcı öncelikle giriş türünü Şekil 3.18’de gösterilen butonlara tıklayarak seçmektedir. “Dersi Anlat” butonuna tıklandığında Şekil 3.19’da gösterildiği gibi bir konuşmacı giriş ekranı açılmaktadır. Sistemde daha önceden kayıtlı olan kullanıcı, kullanıcı adı ve parolasını girerek sisteme erişmektedir. “Canlı Ders” butonuna tıklandığında ise Şekil 3.20’de gösterilen, dersi takip etmek için gerekli ve her ders için benzersiz olan kodun istendiği ve doğrulandığı canlı takibe giriş ekrana ulaşılmaktadır. Uygulamanın sol alt köşesinde ise bir ana menü yer almaktadır.

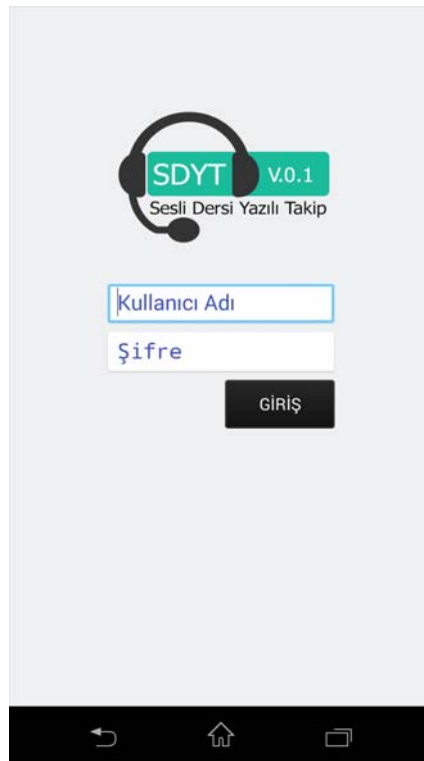


Şekil 3.18. Giriş ekranı.

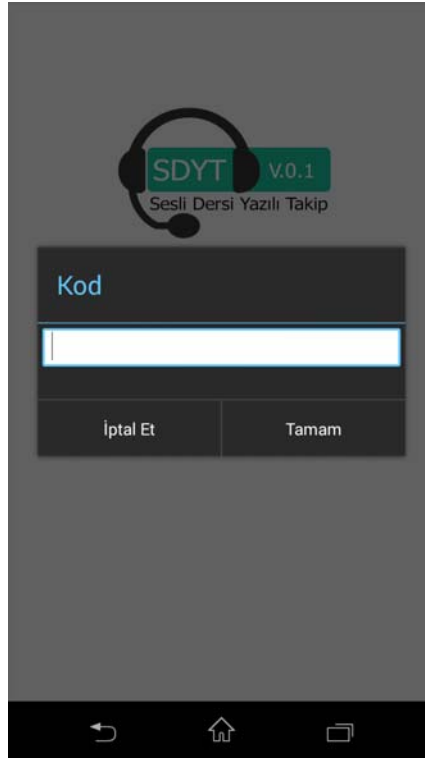
### 3.2.1.2. Giriş paneli

“Ders Anlat” butonuna tıklayan kullanıcı Şekil 3.19’da ki ekrana yönlendirilir. Bu ekranda, kullanıcı sistemde var olan kullanıcı adı ve şifresini girerek sisteme ulaşır. Bu ekran konuşmacının sisteme eriştiği ekrandır. Doğru bir şekilde giriş yapan kullanıcı Şekil 3.21’de gösterilen ekrana ulaşır. Kullanıcı bu ekrandan sistemde var olan bir dersi seçip anlatabileceği gibi yeni bir ders girerek de anlatabilir.

“Canlı Ders” butonuna tıklayan kullanıcı Şekil 3.20’deki ekrana yönlendirilir. Bu ekranda kullanıcı, konuşmacının verdiği iki haneli kodu girerek dersi takip edebileceği ekrana ulaşır. Uygulama aynı anda birden fazla konuşmacı-takipçi oturumunu desteklediği için girilen bu kod istenilen derse ulaşmak için gereklidir ve benzersiz bir koddur.



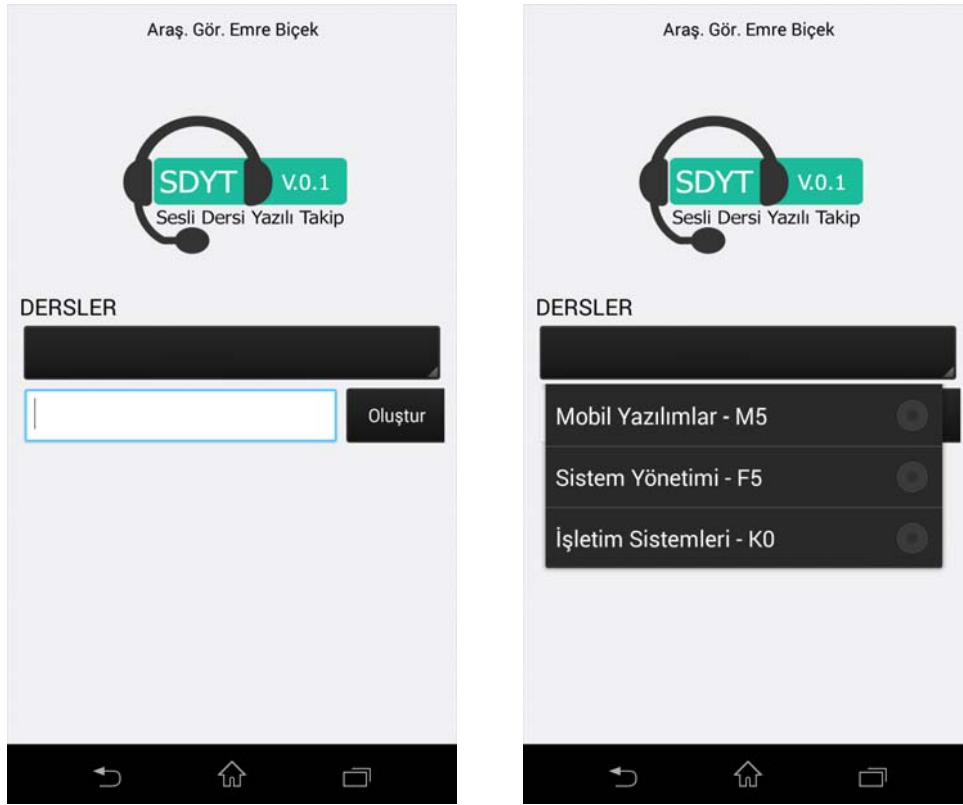
Şekil 3.19. Konuşmacı giriş ekranı.



Şekil 3.20. Canlı takibe giriş ekranı.

### 3.2.1.3. Ders seçim ekranı

Sisteme konuşmacı olarak giriş yapan kullanıcı Şekil 3.21'deki ekrana ulaşır. Bu ekranda kullanıcının sisteme daha önceden girmiş olduğu dersler listelenmektedir. Kullanıcı bu listeden anlatmak istediği dersi seçebilmektedir. Farklı bir ders anlatmak isteyen kullanıcı yeni ders adı alanına anlatılacak olan yeni dersin adını yazmalıdır. Her ders için benzersiz bir ders kodu vardır. Ders kodu dersi takip etmek isteyen kullanıcılardan sisteme ilk girişlerinde istenen iki haneli koddur. Bu kodlar anlık olarak konuşmacı tarafından üretilebildiği gibi daha önceden sisteme girilmek sureti ile de oluşturulabilmektedir. Bu ekranda yer alan ders kodu üret butonuna tıklanıp benzersiz bir kod üretilebilir veya konuşmacı tarafından sisteme daha önceden girilmiş bir ders kodu var ise ders kodum var butonuna tıklanarak daha önceden girilmiş kodun aktif olması sağlanır.



Şekil 3.21. Ders seçim ekranı.

#### 3.2.1.4. Sesi yazıya dönüştürme ekranı

Kullanıcı Ders Seçim Ekranındaki gerekli işlemleri yaptıktan sonra Şekil 3.22’de gösterilen Sesi yazıya dönüştürme ekranına ulaşır. Ekranın üst kısmında konuşmacının adı, soyadı ve anlattığı dersin adı yer alır. Orta kısımda yazı akış alanı bulunur. Alt kısımda ise Başlat ve Bitir butonları yer alır. Bu ekranda kullanılan Speech-to-Text algoritması konuşmacının sesini anlık olarak yazıya dönüştürmektedir. Yazıya dönüştürülen ses sistemin veri tabanına anlık olarak kaydedilmektedir ve ayrıca konuşmacının konuşmasını metin olarak takip edebilmesi de bu ekrandan sağlanmaktadır. Konuşmacı anlatımı bitirdiğinde bitir butonuna basarak dersi bitirir.

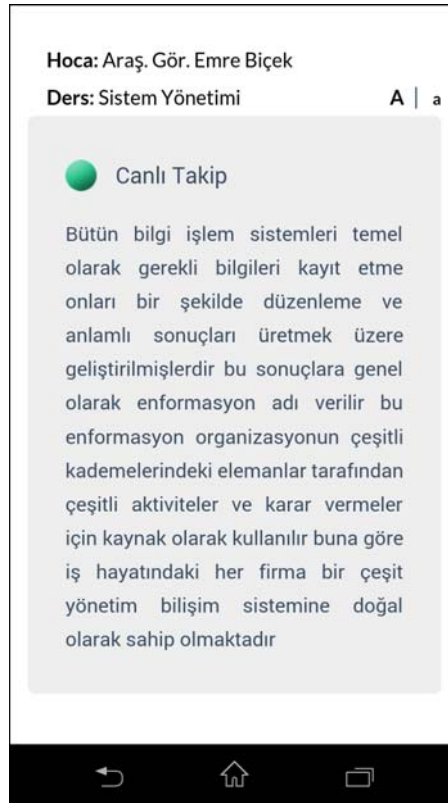




Şekil 3.22. Sesi yazıya dönüştürme ekranı.

### 3.2.1.5. Canlı ders takip ekranı

Dersi takip etmek isteyen kullanıcı canlı takibe giriş ekranından (Bkz. Şekil 3.20) istenilen dersin takip kodunu girerek Şekil 3.23’da gösterilen Canlı Ders Takip Ekranına ulaşır. Bu ekranın sol üst köşesinde dersi anlatan kullanıcının adı, soyadı ve anlatmış olduğu dersin adı yer almaktadır. Orta kısımda kullanıcının dersi takip edeceği yazı akış alanı yer almaktadır. Yazı akış alanındaki metinler konuşmayla eş zamanlı olarak ekrana sıralı bir şekilde yazılır. Bitir butonuna basıldığında ise oturum sonlanır ve dersten çıkış yapılmış olur.



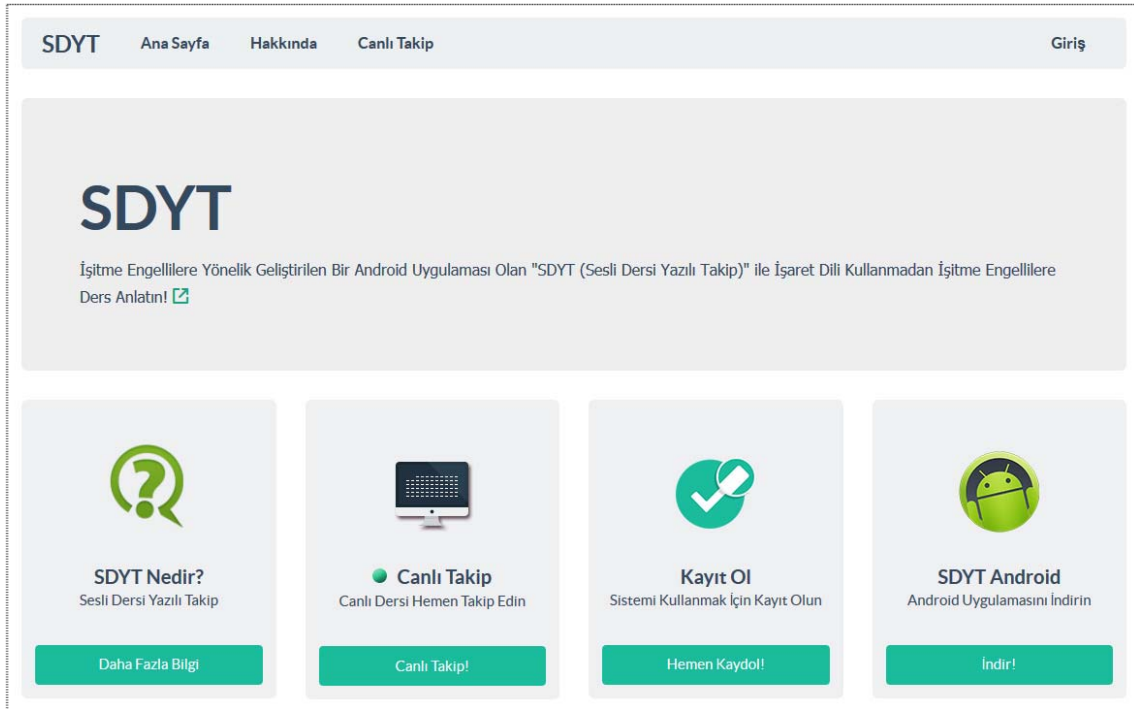
Şekil 3.23. Canlı ders takip ekranı.

### 3.2.2. Web uygulama

Web bölümü uygulama kullanıcıları için geliştirilmiş yönetim sayfalarını, dersi internetten takip etmek isteyenler için canlı ders takip ekranını, mobil uygulama indirme linkini ve uygulama hakkında bilgilerin verildiği sayfaları içerir. Ders Yönetim Bölümünde iki farklı giriş tipi bulunmaktadır. İlki anlatıcılar diğeri ise takipçiler için geliştirilmiştir. Her iki giriş tipinin yetkileri farklı olacak şekilde ayarlanmıştır. İlerleyen bölümlerde bu yetkilerin neler olduğu ve sistemin nasıl kullanıldığı ara yüzler üzerinden anlatılmıştır.

### 3.2.2.1. Ana sayfa

Ana sayfa Şekil 3.24'te görüldüğü gibi üç bölümden oluşmaktadır. Üst bölümde ana sayfa, hakkında, canlı takip ve giriş linklerinin bulunduğu ana menü bulunmaktadır.



Şekil 3.24 Ana sayfa.

Giriş linki yönetim paneline ulaşmak için kullanıcı doğrulama sayfasına yönlendirilmektedir. Orta bölümde uygulama hakkında kısa bir bilgi verilmiş ve proje hakkında detaylı bilgi için yönlendirme linki verilmiştir. Alt bölümde ise SDYT Nedir?, Canlı Takip, Kayıt Ol, SDYT Android başlıkları altında dört adet kutucuk yer almaktadır. Bunlar; sırasıyla uygulama hakkında detaylı açıklamaların yer aldığı ekrana, canlı ders takip ekranına, sisteme kayıt olma ekranına ve Android uygulamayı indirme ekranına yönlendirmeleri içeren kısımlardır.

### 3.2.2.2. Hakkında

Bu bölümde Sesli Dersi Yazılı Takip uygulamasının amacı, kapsamı ve nasıl işlediği hakkındaki bilgiler Şekil 3.25’de verilmiştir.



Şekil 3.25. Hakkında ekranı.

### 3.2.2.3. Kayıt paneli

Ders anlatmak isteyen kullanıcıların uygulamayı kullanabilmeleri için öncelikle sisteme üye olmaları gerekmektedir. Dersi canlı olarak takip edecek kullanıcıların üye olma zorunluluğu yoktur; ancak takipçiler için hazırlanmış ders yönetim panelini kullanabilmek ve daha önceden yapılmış olan derslere ulaşmak için üye olmaları gerekmektedir.

Ana sayfada alt bölümde bulunan Kayıt Ol linkine (Bkz. Şekil 3.24) tıklandığında kayıt paneli açılır. Kayıt olmak için; ad, soyad, kullanıcı adı, şifre, cep telefonu ve e-mail adresinin girilmesi yeterlidir. Kayıt Paneli Şekil 3.26’da verilmiştir.

The image shows a web application interface with a registration panel (Kayıt Paneli) overlaid on the main content. The main content area has a header with 'SDYT', 'Ana Sayfa', and 'Hakkımızda' links. Below the header, there is a large 'SDYT' logo and a section titled 'İşitme Engellilere Yönelik Geliştirilmiş Sesli Ders Anlatımı' with a green checkmark icon. To the left, there is a section titled 'SDYT Nedir?' with a green question mark icon and a 'Daha Fazla Bilgi' button. To the right, there is a section titled 'SDYT Android' with a green Android robot icon and an 'İndir!' button. The registration panel (Kayıt Paneli) is a white box with a close button (X) in the top right corner. It contains the following fields and buttons:

- Ad: Input field
- Soyad: Input field
- Kullanıcı Adı: Input field
- Şifre: Input field
- Şifre Tekrar: Input field
- Cep Telefonu: Input field
- E-Mail: Input field
- Kaydet: Green button
- Kapat: Grey button

Şekil 3.26. Kayıt paneli.

### 3.2.2.4. Canlı takip

Anlatılan dersi canlı olarak izlemek isteyen kullanıcı ana sayfada bulunan Canlı Takip butonuna tıklar. Karşısına Şekil 3.27’de gösterilen ekran gelir. Bu ekranda kullanıcıdan ders takip kodu istenir. İlgili kodu giren kullanıcı, sistem tarafından kod doğrulaması yapıldıktan sonra Şekil 3.28’teki dersin canlı takip sayfasına yönlendirilir.

Şekil 3.27 Canlı ders giriş ekranı.

Şekil 3.28’de gösterilen canlı ders takip ekranında sol üst köşede dersi anlatan hocanın adı, soyadı ve ders adı bilgileri bulunur. Sağ üst köşede ise karakterleri görme güçlüğü çeken takipçiler için metin puntosunu büyütüp küçültmeye yarayan fonksiyon tuşları yerleştirilmiştir. Eğer ders aktifse yani dersi anlatan kullanıcı sisteme bağlı ve aktif olarak konuşuyor ise metin bloğunun üst kısmındaki yeşil buton yanıp söner; kullanıcı aktif değilse daha önceden anlatılmış bir ders ise buton kırmızı olarak yanıp söner. Kullanıcılar kodunu bildikleri derse, dersin aktif veya pasif olması durumuna bakılmaksızın ulaşabilirler.

SDYT
Ana Sayfa
Hakkında
Canlı Takip
Giriş

Hoca: Yrd. Doç. Dr. M. Nuri Almalı  
Ders: Devre Analizi
A | a

## Canlı Takip

Merhaba arkadaşlar bu dersimizde sizlere devre analizinden bahsedeceğim devre analizi bir elektrik devresinde bulunan bütün düğüm gerilim voltajlarını ve kollarındaki elektrik akımı akımları bulmak için gerçekleştirilen bir yöntemdir lineer DC devreleri bağımsız voltaj ve akım kaynakları bağımlı akım ve voltaj kaynakları ve lineer dirençler içerir Lineer AC devreleri de minimum bir lineer diferansiyel eleman ek olarak minimum bir AC kaynak içerir eğer bir devrede kondansatör ve bobin yoksa DC dönem analiz yöntemleri uygulanabilir eğer devrede bir veya daha çok lineer diferansiyel eleman ve bir AC kaynak var ise AC dönem analiz yöntemleri uygulanmalıdır.

Şekil 3.28 Canlı ders takip ekranı.

### 3.2.2.5. Anlatıcı yönetim paneli

Yönetim panelinde iki ayrı yetki türü vardır. Bunlar; anlatıcı yetkisi ve takipçi yetkisidir. Yetkiler her iki kullanıcı için de farklı olarak kurgulanmıştır. Anlatıcı yönetim paneli ana sayfası Şekil 3.29’da gösterildiği gibidir.

SDYT
Ana Sayfa
Üye İşlemleri
Dersler
Çıkış

Hoşgeldiniz, Yrd. Doç. Dr. M. Nuri Almalı



### Dersler

Ders Ekleyebilir, Ders Silabilir, Ders İçeriklerini Güncelleyebilirsiniz.

[+ Yeni Ders Ekle](#)

Ara:

| Sıra | Ders Adı          | Hoca                         | Tarih               | İşlemler                                                          |
|------|-------------------|------------------------------|---------------------|-------------------------------------------------------------------|
| 1    | Kimya             | Yrd. Doç. Dr. M. Nuri Almalı | 2015-10-18 14:53:18 | <a href="#">İndir</a> <a href="#">Düzenle</a> <a href="#">Sil</a> |
| 2    | Devre Analizi     | Yrd. Doç. Dr. M. Nuri Almalı | 2015-10-18 01:15:55 | <a href="#">İndir</a> <a href="#">Düzenle</a> <a href="#">Sil</a> |
| 3    | Mikroişlemciler 2 | Yrd. Doç. Dr. M. Nuri Almalı | 2015-10-18 01:16:49 | <a href="#">İndir</a> <a href="#">Düzenle</a> <a href="#">Sil</a> |

Şekil 3.29 Anlatıcı yönetim paneli ana sayfa.

Anlatıcının yetkileri şu şekildedir;

- i. Ders indirme
- ii. Ders düzenleme
- iii. Ders silme
- iv. Yeni ders ekleme
- v. Kişisel bilgi güncelleme

Yönetim paneli ana sayfasında bulunan menü her iki yetki türü için aynıdır. Menüde ana sayfa, üye işlemleri, dersler ve çıkış linkleri bulunmaktadır. Menü'nün sağ alt tarafında giriş yapan kullanıcının adı ve soyadı bilgileri yer almaktadır. Orta bölümde “Yeni Ders Ekle” butonu yer almaktadır. Bu butona tıklandığında Şekil 3.30’de gösterilen ders ekleme ekranına ulaşılır.

Alt bölümde giriş yapan kullanıcının sisteminde var olan dersler listelenmektedir. Tabloda sırasıyla sıra no, ders adı, hoca adı, ders tarihi ve işlemler sütunları yer almaktadır. İşlemler sütununda sırasıyla indir, düzenle ve sil butonları vardır. İndir butonuna tıklandığında ilgili dersin PDF formatında indirilmesi sağlanmaktadır.

Düzenle butonuna tıklandığında Şekil 3.31’de gösterilen ekran açılır ve bu ekranda ders içeriği güncellenebilir, yazım hataları giderilebilir. Dersin takipçiler tarafından takip edilebilmesi için gerekli kod atanabilir ve değiştirilebilir. Ayrıca ders adı ve ders tarihi bilgileri de güncellenebilir. Ders kodu boş bırakıldığı takdirde mobil uygulama üzerinden sisteme giriş yapan kullanıcı ders seçim ekranından (Bkz. Şekil 3.21) kodu olmayan bir ders seçtiğinde sistem otomatik olarak benzersiz bir kod üretir ve üretilen kod veri tabanına kaydedilir. Sil butonuna tıklandığında ders ile ilgili tüm veriler veri tabanından silinmiş olur.



SDYT
Ana Sayfa
Üye İşlemleri
Dersler
Çıkış

Hoşgeldiniz, Yrd. Doç. Dr. M. Nuri Almalı



### Dersler

Ders Ekleyebilir ve Ders İçeriklerini Güncelleyebilirsiniz.

Tüm Dersler

Ders Adı

Ders Kodu

Kaydet

Proje Hakkında

Şekil 3.30 Anlatıcı yönetim paneli ders ekleme ekranı.

SDYT
Ana Sayfa
Üye İşlemleri
Dersler
Çıkış

Hoşgeldiniz, Yrd. Doç. Dr. M. Nuri Almalı



### Ders Güncelleme

Tüm Bilgileri Eksiksiz Doldurunuz.

Tüm Dersler

Ders Adı

Ders İçeriği

Merhaba arkadaşlar bu dersimizde sizlere devre analizinden bahsedeceğim devre analizi bir elektrik devresinde bulunan bütün düğüm gerilim voltajlarını ve kollarındaki elektrik akımı akımları bulmak için gerçekleştirilen bir yöntemdir lineer DC devreleri bağımsız voltaj ve akım kaynakları bağımlı akım ve voltaj kaynakları ve lineer dirençler içerir Lineer AC devreleri de minimum bir lineer diferansiyel eleman ek olarak minimum bir AC kaynak içerir eğer bir devrede kondansatör ve bobin yoksa DC dönem analiz yöntemleri uygulanabilir eğer devrede bir veya daha çok lineer diferansiyel eleman ve bir AC kaynak var ise AC dönem analiz yöntemleri uygulanmalıdır.

Ders Kodu

Ders Tarihi

Güncelle

Şekil 3.31 Anlatıcı yönetim paneli ders düzenleme ekranı.

Ana menüde bulunan üye işlemleri linkine tıklandığında Şekil 3.32’de gösterilen Anlatıcı yönetim paneli üye bilgileri ekranı açılır. Bu ekranda ad, soyad, kullanıcı adı, şifre, cep telefonu, e-mail bilgileri bulunmaktadır. Kullanıcı üyelik bilgilerini bu ekrandan güncelleyebilmektedir. Takipçi yönetim panelinde de üyelik bilgileri ekranı aynı şekildedir ve her iki kullanıcı türü içinde girilmesi gereken bilgiler aynıdır.

SDYT Ana Sayfa Üye İşlemleri Dersler Çıkış

Hosgeldiniz, Yrd. Doç. Dr. M. Nuri Almalı

**Üye Bilgileri**  
Tüm Bilgileri Eksiksiz Doldurunuz.

Ad  
Yrd. Doç. Dr. M. Nuri

Soyad  
Almalı

Kullanıcı Adı  
mehmet

Şifre  
\*\*\*\*\*

Şifre Tekrar

Cep Telefonu  
05354178512

E-Mail  
mna1@yyu.edu.tr

Güncelle

Şekil 3.32 Anlatıcı yönetim paneli üye bilgileri ekranı.

### 3.2.2.6. Takipçi yönetim paneli

Uygulama üzerinden daha önceden yapılmış dersleri takip etmek ve indirmek isteyen kullanıcıların kullanması gereken panel takipçi yönetim panelidir. Bu paneldeki ana menü anlatıcı yönetim panelindeki ile aynıdır. Şekil 3.33’te gösterilen takipçi yönetim paneli ana sayfasındaki listede ders adı, hoca adı, ders tarihi ve işlemler sütunları yer almaktadır. İşlemler sütununda anlatıcı yönetim panelindekinden farklı olarak sadece indirme butonu yer almaktadır. Takipçi yetkisine sahip kullanıcılar sadece dersleri indirebilmektedirler. Dersler ile ilgili herhangi bir düzenleme işlemi yapamamaktadırlar. Sistemdeki tüm dersler kullanıcı tarafından görülmektedir.

Uygulama bir açık ders platformu olarak düşünüldüğünden herhangi bir kısıtlama yapmadan tüm dersler kullanıcılar ile paylaşılmaktadır.

SDYT Ana Sayfa Üye İşlemleri Dersler Çıkış

Hosgeldiniz, Kerim Mercan

**Dersler**  
Dersleri .pdf Formatında İndirebilirsiniz.

Ara:

| Sıra | Ders Adı          | Hoca                         | Tarih               | İşlemler              |
|------|-------------------|------------------------------|---------------------|-----------------------|
| 1    | Kimya             | Yrd. Doç. Dr. M. Nuri Almalı | 2015-10-18 14:53:18 | <a href="#">İndir</a> |
| 2    | Devre Analizi     | Yrd. Doç. Dr. M. Nuri Almalı | 2015-10-18 01:15:55 | <a href="#">İndir</a> |
| 3    | Mikroişlemciler 2 | Yrd. Doç. Dr. M. Nuri Almalı | 2015-10-18 01:16:49 | <a href="#">İndir</a> |
| 4    | Akım Teorisi      | Hasan Sayar                  | 2015-12-10 01:15:41 | <a href="#">İndir</a> |
| 5    | Elektrik          | Emre Biçek                   | 2015-10-18 01:16:52 | <a href="#">İndir</a> |
| 6    | Sistem Yönetimi   | Emre Biçek                   | 2015-10-18 01:16:35 | <a href="#">İndir</a> |
| 7    | Mikroişlemciler   | Emre Biçek                   | 2015-10-18 01:15:41 | <a href="#">İndir</a> |

Proje Hakkında

Şekil 3.33 Takipçi yönetim paneli ana sayfa.

#### 4. TARTIřMA VE SONUÇ

İletiřim hayatın vazgeçilmez bir unsurudur. İletiřim bireyler arasında başlayıp aileleri, toplumları ve ülkeleri birbirine bağlayan en güçlü bağıdır. Tarih boyunca insanların her zaman iletiřim ihtiyacı içerisinde olduđu gözlemlenmiřtir. Bireyler sosyal yaşamlarında sağlıklı iliřkiler kurmak için birbirleriyle iyi iletiřim içinde olmalıdırlar. İletiřim sadece sözcüklerle yapılan bir eylem değıldir. İletiřimin, söz, iřaret ve yazı gibi birçok yolu vardır. Her toplum kendine özgü sözcüklerle iletiřim kursa da aslında iletilmek istenen duygu her dilde aynıdır. Dil insanlar arasındaki en önemli iletiřim aracıdır.

Dünya sağılık örgütü verilerine göre dünya üzerinde 360 milyon insan iřitme problemi yaşamaktadır. İřitme güçlüğü çeken bireyler iřitme cihazları vb. teknolojilerin yardımıyla çevreleri ile iletiřime geçebilirken. Hiç duymayan insanlar iletiřim kurabilmek için iřaret dilini kullanırlar. Toplumumuzda herhangi bir iřitme problemi olmayan insanların büyük çoğunluğu iřaret dilinin ne olduđu hakkında bilgi sahibidir ancak bu dili öğrenen ve iřitme engelliler ile iletiřime geçmede kullanan kiři sayısı çok azdır. Bu durumun bir sonucu olarak da toplum ile iletiřim kurmada güçlük çeken iřitme engelliler, kendi aralarında çok daha güçlü bir iletiřime sahiptirler. Dolayısıyla dıř dünya ile iletiřimleri zayıf olduğından sosyal ve kültürel alanlarda zorluklar yaşarlar.

İřitme engellilerin eğitimi birçok bakımdan zorluklar içeren bir konudur. İnsanlar doğumdan başlamak üzere, yaşamları sona erinceye kadar sürekli olarak bir şeyler öğrenirler. İřitme engelliler henüz bebeklik çağlarında ebeveynleri ile yaşadıkları iletiřim problemlerinden dolayı öğrenmeleri gereken temel bilgilerden dahi yoksun kalabilmektedirler. Çocuđu iřitme engeli olan ebeveynler bu durumu bir an önce kabullenip, çocuklarının eğitim alanlarında onlara yardımcı olacak uzmanlarla iletiřim kurmalıdırlar. Onların daha sağlıklı bir eğitim hayatına sahip olmalarını sağlamalıdırlar. İleri seviyede iřitme kaybı olan çocuklar tüm eğitim hayatlarında özel eğitime ihtiyaç duyarlar. Onlar için özel olarak hazırlanmış eğitim programları uygulayan eğitim kurumlarında öğrenim görürler. Bu eğitim kurumları onların özel gereksinimlerini karşılayabilecek gerekli teknolojik altyapıya sahip olmalıdır.

Hızla gelişen teknolojiler her alanda insanların yaşamlarını kolaylaştırıcı çözümler sunmaktadır. Günümüzde çok popüler hale gelen mobil uygulamalar, işitme engeline sahip insanların bundan birkaç yıl öncesine kadar yapamadıkları şeyleri yapabilmelerini sağlamaktadırlar. SMS, E-Posta ve anlık mesajlaşma uygulamaları uzun yıllardır kullanılmakta ve işitme engellilerin, işitme engeli olmayan bireyler ile iletişimlerini oldukça kolaylaştırmaktadır. Günümüz teknolojileri ile ses ve görüntü işleme üzerine atılan adımlar, işitme engellilere yönelik geliştirilen uygulamalarda kullanılarak, onların hayatlarını değiştirecek çözümlerin ortaya çıkmasını sağlamaktadır. Mobil telefon kullanan birey sayısı göz önüne alındığında bu teknolojilerin ne denli ulaşılabilir olduğu ortaya çıkmaktadır.

Bu tez çalışmasının ilk bölümlerinde geçmişten günümüze işitme engellilere yönelik geliştirilmiş cihazlar ve yazılımlardan bahsedilmiştir. Bu teknolojilerin bazıları görüntünün işlenmesi üzerine kurulu iken bazıları da sesin işlenmesi üzerine kuruludur. Bu tez kapsamında gerçekleştirilen uygulamada sesin işlenmesi ile sonuca ulaşılması hedeflenmiştir. Tez kapsamında işitme engellilerin eğitim hayatına yönelik bir çalışma yapılmıştır. Bilindiği üzere işitme engelliler kendilerine özel eğitim verilen eğitim kurumlarında öğrenim görmektedirler. Bu eğitim kurumlarında alanında uzman, işaret diline hâkim öğretmenler görev almaktadır ancak işaret dili bilmeyen bir öğretmenin bu okullardaki öğrencilerle bilgilerini paylaşabilmesi çok zordur. Buradan hareketle işaret dili bilmeyen öğretmenlerin bu okullarda ya da işitme engellilerin bulunduğu bir toplulukta bilgilerini paylaşabilmeleri amaçlanmıştır.

Tez kapsamında geliştirilen uygulama ile işaret dilini bilmeyen öğretmenlerin işitme engellilere mobil bir uygulama üzerinden ders anlatabilmeleri sağlanmıştır. Anlatılan bu derslerin hem mobil bir uygulama üzerinden hem de web tabanlı bir sistem üzerinden anlık olarak takip edilebilmesi sağlanmıştır. Uygulama aynı zamanda işitme engellilere yönelik hazırlanmış bir açık ders platformu niteliğindedir. Sisteme takipçi olarak giriş yapan kullanıcılar daha önce anlatılmış tüm derslere ulaşabilmekte ve ders içeriklerini indirebilmektedirler. Sisteme anlatıcı olarak giriş yapan kullanıcılar ise anlatmış oldukları tüm dersleri görebilmekte ve anlattıkları derslerin metinleri üzerinde değişiklik yapabildikleri gibi yeni ders ekleme, silme gibi diğer birçok özelliğe de erişebilmektedirler.

Uygulama ses işleme üzerine yapılmış ve yüksek maliyetlerle sahip farklı uygulamalara alternatif bir ücretsiz uygulama niteliğindedir. Geçmişte yapılmış ses işlemeye dayalı uygulamalar daha çok karşılıklı diyalog üzerine kurulmuştur. Bu tez kapsamında geliştirilmiş uygulamada ise bütünleşik bir sistem geliştirilerek her alandan eğitmenin işitme engellilere ders anlatabilmesi sağlanmış ve bu derslerin internet üzerinden paylaşılmasıyla büyük bir işitme engelli kitleye ulaşılmıştır. Proje sadece tamamı işitme engellilerden oluşan ortamlarda değil aynı zamanda işitme engeli olmayan bireylerinde bulunduğu konferans, toplantı gibi organizasyonlarda da kullanılabilir durumdadır.

İnsanın tüm yönleriyle gelişiminin en önemli faktörü olan eğitim hakkı kutsaldır. Toplumumuzun bu bilinçle hareket ederek çeşitli nedenlerden dolayı eğitim almakta güçlük çeken bireylere yönelik adımlar atması gerekmektedir. Bu tezde gerçekleştirilen projeye de bu bilinçle başlanılmış ve sonuca ulaşılmıştır. Bu sistemin Türkiye'deki tüm işitme engelliler okullarında kullanılabileceği düşünülmektedir.

## 5. KAYNAKLAR

- Akçamete G., 2003. *İşitme Engellilerin Eğitiminde Öğretmen El Kitabı*. MEB, Özel Eğitim Rehberlik ve Danışma Hizmetleri Genel Müdürlüğü, Ankara.
- Anonim, 2014. Nuance Communications Inc. <http://www.nuance.com/>. Erişim tarihi:10.10.2014.
- Anonim, 2015a. Cut through the clutter for better sound clarity with wireless hearing aid technology. <http://www.carsonhearing.com/wireless-technology/>. Erişim tarihi: 10.05.2015.
- Anonim, 2015b. MySQL. <https://tr.wikipedia.org/wiki/MySQL> Erişim Tarihi: 02.06.2015
- Anonim, 2015c. Liberated Learning. <http://www.transcribeyourclass.ca>. Erişim tarihi: 05.07.2015.
- Anonim, 2015d. Ava: Group conversations made accessible. <https://www.indiegogo.com/project/ava-group-conversations-made-accessible> Erişim tarihi:16.10.2015
- Aran, O., Akarun, L., 2007. Sign language processing and interactive tools for sign language education. *Signal Processing and Communications Applications*. 11-13 Ocak 2007, Eskişehir. 1-4.
- Ataş, A., Genç, A., Belgin, E., 2003. Odyoloji’de kullanılan temel kavramlar, *Pediyatrik Kulak Burun Boğaz Hastalıkları*. (Akyol U). Güneş Kitabevi, Ankara. 35-50.
- Downey, G. J., 2008. *Closed Captioning: Subtitling, Stenography and the Digital Convergence of Text with Television*. JHU Press, Baltimore. 391.
- Edwards, B., 2007. The future of hearing aid technology. *Trends in amplification*, 11(1): 31-45.
- Garrett, J., 2005. Ajax: a new approach to web applications. <http://adaptivepath.org/ajax-new-approach-web-application>. Adaptive Path. Erişim Tarihi:14.03.2015
- Gürboğa, Ç., Kargın, T., 2003. İşitme engelli yetişkinlerin farklı ortamlarda kullandıkları iletişim yöntemlerinin/becerilerinin incelenmesi. *Ankara Üniversitesi Eğitim Bilimleri Fakültesi Dergisi*, 36: 52-64.
- Hashimi, S. Y., Komatini, S., MacLean, D. 2010. *Pro Android 2*. Apress, New York, 718.

- Hasselbring, T. S., Glaser, C. H. W. 2000. Use of computer technology to help students with special needs, *The Future of Children*, **10**. Princeton University, 102-122.
- John P.S., 2015. Android architecture. <http://www.eazytutz.com/android/android-architecture>. Erişim tarihi: 10.06.2015.
- Kovalik, G. 1992. "Silent" Films Revisited: Captioned Films for the Deaf." *Library Trends*. **41**(1): 100-117
- Lang, H. G., 2000. *A Phone of Our Own: The Deaf Insurrection Against Ma Bell*. Gallaudet University Press. Washington, D.C. 245.
- Meier, R., 2012. *Professional Android 4 Application Development*. John Wiley & Sons, Indianapolis. 817.
- Mesbah, A., van Deursen, A., 2008. A component- and push-based architectural style for ajax applications. *The Journal of Systems & Software*. **81**(12): 2194-2209.
- Moller, A. R., 2006. Hearing Impairment, Chap. 9. *Hearing: Anatomy, Physiology and Disorders of the Auditory System*. Academic Press, Burlington, MA, USA. 206.
- Pražák, A., Loose, Z., Trmal, J., Psutka, J. V., Psutka, J., 2012. Captioning of Live TV Programs Through Speech Recognition and Re-Speaking, Part III. *Text, Speech and Dialogue* (Editors: Sojka, P., Horák, A., Kopeček, I., Pala, K.). Springer Berlin Heidelberg, Czech Republic. 513-519.
- Reddy, B. R., Mahender, E., 2008. Speech to text conversion using android platform. *International Journal of Engineering Research and Applications*, **3**(1): 253-258.
- Tebelskis, J., 1995. *Speech Recognition Using Neural Networks* (Doctoral dissertation). Carnegie Mellon University, School of Computer Science, Pittsburgh, Pennsylvania.
- Tüfekçioğlu, U., 1998. İşitme Engelliler, Ünite 8. *Özel Eğitim* (Editör: Eripek S.). T.C. Anadolu Üniversitesi Yayınları, 1018, Eskişehir. 113-114.
- WHO, 2013. World Health Organization: Deafness and Hearing Loss. <http://www.who.int/mediacentre/factsheets/fs300/en/> ErişimTarihi: 20.03.2015
- Wilson, B. S., Dorman, M. F., 2009. The Design of Cochlear Implants, Chap. 7. *Cochlear Implants: Principles & Practices* (Editor: J. K. Niparko). Lippincott Williams and Wilkins, Philadelphia. 357.



- Yang, J., Liao, Z., Liu, F., 2008. The impact of ajax on network performance. *The Journal of China Universities of Posts and Telecommunications*, 14(1): 32-34.
- Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, Xunying., Moore, G. Odel, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P. 2006. *The HTK book version 3.4 Manual*. Cambridge University. 375.
- Zárate, S., 2010. Subtitling for deaf children, *Perspectives on Audiovisual Translation*. Peter Lang, Frankfurt. 107.

## EKLER

### Ek-1: Mobil uygulama kodları

#### CanliDers.java

```
package tez.emrebicek.sesliderstakip;

import android.app.Activity;
import android.app.ProgressDialog;
import android.graphics.Bitmap;
import android.os.Bundle;
import android.view.Window;
import android.view.WindowManager;
import android.webkit.WebView;
import android.webkit.WebViewClient;

public class CanliDers extends Activity {
    private WebView webview;
    private ProgressDialog progress;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);

        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.canlidars);

        init();
    }

    private void init() {
        final Bundle b = getIntent().getExtras();
        webview = (WebView) findViewById(R.id.cdWebView);
        webview.getSettings().setJavaScriptEnabled(true);
        progress = ProgressDialog.show(CanliDers.this, "SDYT Canlı
Ders", "Yükleniyor...", true);
        webview.setWebViewClient(new WebViewClient() {
            @Override
            public void onPageStarted(WebView view, String url, Bitmap
favicon) {
                super.onPageStarted(view, url, favicon);
                progress.show();
            }
            public void onPageFinished(WebView view, String url) {
                super.onPageFinished(view, url);
                progress.dismiss();
            }
        }

        public void onReceivedError(WebView view, int errorCode,
String description, String failingUrl) {
```

```

        progress.dismiss();
    }

    @Override
    public boolean shouldOverrideUrlLoading(WebView view,
String url) {

        webview.loadUrl(getString(R.string.sewebsayfa_url) +
"?Kod=" + b.getString("CanliDersKodu"));
        progress.dismiss();
        return true;
    }
});
webview.loadUrl(getString(R.string.sewebsayfa_url) + "?Kod=" +
b.getString("CanliDersKodu"));
}
}

```

### GirisEkrani.java

```

package tez.emrebicek.sesliderstakip;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.view.Gravity;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import tez.emrebicek.sesliderstakip.helper.BaglantiKontrolu;
import tez.emrebicek.sesliderstakip.helper.Veritabani;

public class GirisEkrani extends Activity {
    private EditText etKadi,etSifre;
    private Button btnGiris;
    Connection conn=null;
    BaglantiKontrolu internet;
    Veritabani vt;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}

```

```

getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
    WindowManager.LayoutParams.FLAG_FULLSCREEN);
requestWindowFeature(Window.FEATURE_NO_TITLE);
setContentView(R.layout.girisekrani);

VeritabaniBaglanti();

init();
}

private void init() {
    etKadi = (EditText) findViewById(R.id.geETKadi);
    etKadi.setFocusable(true);
    etKadi.setFocusableInTouchMode(true);
    etKadi.performClick();

    etSifre = (EditText) findViewById(R.id.geETSifre);

    btnGiris = (Button) findViewById(R.id.geBTNGiris);
    btnGiris.setOnClickListener(new GirisKontrol());

}

private class GirisKontrol implements View.OnClickListener {
    @Override
    public void onClick(View v) {
        KullaniciKontrol();
    }
}

private void VeritabaniBaglanti(){
    internet = new BaglantiKontrolu(getApplicationContext());
    if(internet.isConnectingToInternet()){
        vt =new Veritabani(getApplicationContext());
    }
}

private void KullaniciKontrol(){
    try {
        if(internet.isConnectingToInternet()){
            conn=vt.veritabaniAc();
            Statement query = conn.createStatement();
            ResultSet kayitlar = query.executeQuery("SELECT
id,ad,soyad FROM tblusers WHERE kadi='" + etKadi.getText().toString()
+ "' AND sifre='" + etSifre.getText().toString() + "' AND aktifmi=1
LIMIT 0,1;");
            if(kayitlar.next()){
                Intent i = new
Intent(GirisEkрани.this,HocaEkрани.class);
                Bundle b = new Bundle();
                b.putString("Hoca_Id", kayitlar.getString("id"));
                b.putString("Hoca_Adi", kayitlar.getString("ad"));
                b.putString("Hoca_Soyadi",
kayitlar.getString("soyad"));
                i.putExtras(b);
                startActivity(i);
            }
        }
    }
}

```

```

        finish();
    }
    else{
        MesajGoster(getApplicationContext(),"Sistemde
böyle bir kullanıcı bulunamadı",Gravity.CENTER,0,0);
    }
    else{
        MesajGoster(getApplicationContext(),getString(R.string.internetyok),
Gravity.CENTER,0,0);
    }
    }catch (SQLException e){
        MesajGoster(getApplicationContext(),e.getMessage().toString(),
Gravity.CENTER,0,0);
    }
}

private void MesajGoster(Context c,String s, int g, int x, int y){
    Toast t = Toast.makeText(c, s, Toast.LENGTH_LONG);
    t.setGravity(g,x,y);
    t.show();
}
}

```

## HocaEkrani.java

```

package tez.emrebicek.sesliderstakip;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.text.InputType;
import android.util.Log;
import android.view.Gravity;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.AdapterView;
import android.widget.AdapterViewAdapter;
import android.widget.Button;
import android.widget.EditText;

```

```

import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Random;

import tez.emrebicek.sesliderstakip.helper.BaglantiKontrolu;
import tez.emrebicek.sesliderstakip.helper.SpinnerBosSecim;
import tez.emrebicek.sesliderstakip.helper.Veritabani;

public class HocaEkrani extends Activity {
    Connection conn=null;
    BaglantiKontrolu internet;
    ArrayList<String> dersler,dersler_id;
    Bundle gelenveriler;
    TextView tvHocaAdiSoyadi;
    Button btnDersOlustur;
    EditText etYeniDersAdi;
    Spinner spDersler;
    Veritabani vt;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.hocaekrani);

        VeritabaniBaglanti();

        init();
    }

    private void VeritabaniBaglanti(){
        internet = new BaglantiKontrolu(getApplicationContext());
        if(internet.isConnectingToInternet()){
            vt =new Veritabani(getApplicationContext());
            conn=vt.veritabaniAc();
        }
    }

    private void init() {
        gelenveriler = getIntent().getExtras();
        dersler = new ArrayList<String>();
        dersler_id = new ArrayList<String>();
        Log.d("gelenveri",gelenveriler.getString("Hoca_Adi"));
        etYeniDersAdi = (EditText) findViewById(R.id.heTYeniDersAdi);
        spDersler = (Spinner) findViewById(R.id.heSPDersler);
    }
}

```

```

DersleriCek();

    tvHocaAdiSoyadi = (TextView)
findViewById(R.id.heTVHocaAdiSoyadi);
    tvHocaAdiSoyadi.setText(gelenveriler.getString("Hoca_Adi") + "
" + gelenveriler.getString("Hoca_Soyadi"));

    btnDersOlustur = (Button) findViewById(R.id.heBTNDersOlustur);
    btnDersOlustur.setOnClickListener(new DersOlustur());
}

private void DersleriCek() {
    try {
        if(internet.isConnectingToInternet()){
            Statement query = conn.createStatement();
            ResultSet kayitlar = query.executeQuery("SELECT
id,dersadi,derskodu FROM tbldersler WHERE
hocaId="+gelenveriler.getString("Hoca_Id")+" AND aktifmi=1 ORDER BY id
DESC;");

            dersler.clear();
            while(kayitlar.next()){
                dersler.add(kayitlar.getString("dersadi")+" -
"+kayitlar.getString("derskodu"));
                dersler_id.add(kayitlar.getString("id"));
            }
            if(dersler.size()>0){
                ArrayAdapter<String> adapterbolum = new
ArrayAdapter<String>(getApplicationContext(),android.R.layout.simple_l
ist_item_single_choice,dersler);
                spDersler.setAdapter(new
SpinnerBosSecim(adapterbolum,R.layout.spinnerbossecim,HocaEkrani.this
);
                spDersler.setOnItemSelectedListener(new
DerslerTiklama());
            }
        }
        else{
            MesajGoster(getApplicationContext(),getString(R.string.internetyok),
Gravity.CENTER,0,0);
        }
    }catch (SQLException e){
        MesajGoster(getApplicationContext(),e.getMessage().toString(),
Gravity.CENTER,0,0);
    }
}

private class DersOlustur implements View.OnClickListener {
    @Override
    public void onClick(View v) {
        try {
            Statement query = conn.createStatement();
            boolean sonuc = query.execute("INSERT INTO
tbldersler(hocaId,dersadi,derskodu,derstarihi,aktifmi) VALUES("
+gelenveriler.getString("Hoca_Id")+",'"+etYeniDersAdi.getText().toStri

```

```

ng()+"', '"+DersKoduUret(2)+"', '"+SuankiZamaniUret()+"',1)");
    if(!sonuc) {
        MesajGoster(getApplicationContext(), "Ders Ekleme
işlemi başarılı", Gravity.CENTER,0, 0);
        etYeniDersAdi.setText("");
        //Mevcut activity yenile
        Intent intent = getIntent();
        finish();
        startActivity(intent);
    }
    else{
        MesajGoster(getApplicationContext(),"Ders Ekleme
işlemi başarısız", Gravity.CENTER,0,0);
    }
} catch (Exception e){

MesajGoster(getApplicationContext(),e.getMessage().toString(),
Gravity.CENTER,0,0);
    }
}

private String SuankiZamaniUret() {
    DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss");
    Calendar cal = Calendar.getInstance();
    return dateFormat.format(cal.getTime());
}

private String DersKoduUret(int len) {
    String AB = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    Random rnd = new Random();
    StringBuilder sb = new StringBuilder( len );
    for( int i = 0; i < len; i++ )
        sb.append( AB.charAt( rnd.nextInt(AB.length()) ) );

    if(KodKontrolu(sb.toString())){
        DersKoduUret(2);
    }

    return sb.toString();
}

private boolean KodKontrolu(String str){
    try {
        if(internet.isConnectingToInternet()){
            Statement query =
conn.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
            ResultSet kayitlar = query.executeQuery("SELECT
COUNT(*) FROM tbldersler WHERE derskodu='"+str+"'");
            if(kayitlar.next()){
                if(kayitlar.getInt(1)>0){
                    return true;
                }
            }
            else{
                return false;
            }
        }
    }
}

```



```

        }
    }
}
} catch (SQLException e){
}
return false;
}

private class DerslerTiklama implements
android.widget.AdapterView.OnItemClickListener {
    private boolean ilkgiris = false;
    @Override
    public void onItemClick(AdapterView<?> adapter, View view,
int position, long id) {
        if (ilkgiris == true)
        {
            position--;
            String dersid = dersler_id.get(position);
            String dersadi = adapter.getSelectedItem().toString();
            Intent i = new
Intent(HocaEkrani.this, KonusmaEkrani.class);
            Bundle b = new Bundle();
            b.putString("Hoca_Id",
gelenveriler.getString("Hoca_Id"));
            b.putString("Hoca_Adi",
gelenveriler.getString("Hoca_Adi"));
            b.putString("Hoca_Soyadi",
gelenveriler.getString("Hoca_Soyadi"));
            b.putString("Ders_Id", dersid);
            b.putString("Ders_Adi", dersadi);
            i.putExtras(b);
            startActivity(i);
            finish();
        }
        else{
            ilkgiris = true;
        }
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {

    }

}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // TODO Auto-generated method stub
    super.onCreateOptionsMenu(menu);
    MenuInflater m=getMenuInflater();
    m.inflate(R.menu.menu, menu);
    return super.onCreateOptionsMenu(menu);
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // TODO Auto-generated method stub

```

```

switch (item.getItemId()) {
    case R.id.dersyonetimsistemi:
        DersYonetimSistemi();
        return true;
    case R.id.dersanlat:
        DersAnlat();
        return true;
    case R.id.canlidervers:
        CanliDersEkrani();
        return true;
    case R.id.hakkinda:
        HakkindaEkrani();
        return true;
    case R.id.cikis:
        CikisIslemi();
        return true;
}
return false;
}

private void HakkindaEkrani() {
    // TODO Auto-generated method stub
    final AlertDialog.Builder alert_dlg = new
AlertDialog.Builder(HocaEkrani.this);
    alert_dlg.setTitle("Hakkında");
    alert_dlg.setMessage(getString(R.string.hakkinda));

    alert_dlg.setNegativeButton(getString(R.string.tamam),
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int
id) {
                dialog.cancel();
            }
        });

    alert_dlg.setCancelable(true);
    AlertDialog alertDialog = alert_dlg.create();
    alert_dlg.show();
}

private void CikisIslemi() {
    AlertDialog.Builder builder = new
AlertDialog.Builder(HocaEkrani.this);
    builder.setTitle("Çıkış");
    builder.setMessage(getString(R.string.uygulamapapatmaistegi));
    builder.setCancelable(false);
    builder.setPositiveButton(getString(R.string.evet),
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int
id) {
                Intent homeIntent = new
Intent(Intent.ACTION_MAIN);
                homeIntent.addCategory( Intent.CATEGORY_HOME
);
                homeIntent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                startActivity(homeIntent);
            }
        });
}

```

```

        finish();
        System.exit(1);
    }
});
builder.setNegativeButton(getString(R.string.hayir),
    new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int
id) {
            Toast.makeText(getApplicationContext(),
getString(R.string.iptal), Toast.LENGTH_SHORT).show();
            dialog.cancel();
        }
    });
AlertDialog alertDialog = builder.create();
alertDialog.show();
}

private void CanliDersEkrani() {
    AlertDialog.Builder builder = new
AlertDialog.Builder(HocaEkrani.this);
    builder.setTitle(getString(R.string.sekod));

    // Set up the input
    final EditText input = new EditText(HocaEkrani.this);
    input.setInputType(InputType.TYPE_TEXT_FLAG_CAP_CHARACTERS);
    builder.setView(input);

    // Set up the buttons
    builder.setPositiveButton(getString(R.string.tamam), new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            String Kod = input.getText().toString();

            Intent i = new Intent(HocaEkrani.this,
tez.emrebicek.sesliderstakip.CanliDers.class);
            Bundle b = new Bundle();
            b.putString("CanliDersKodu", Kod);
            i.putExtras(b);
            startActivity(i);
        }
    });
    builder.setNegativeButton(getString(R.string.iptalet), new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.cancel();
        }
    });

    builder.show();
}

private void DersAnlat() {
}

```

```

        private void DersYonetimSistemi() {
            startActivity(new
Intent(HocaEkrani.this,DersYonetimSistemi.class));
        }

        private void MesajGoster(Context c,String s, int g, int x, int y){
            Toast t = Toast.makeText(c, s, Toast.LENGTH_LONG);
            t.setGravity(g,x,y);
            t.show();
        }
    }
}

```

### KonusmaEkrani.java

```

package tez.emrebicek.sesliderstakip;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.ActivityNotFoundException;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.speech.RecognizerIntent;
import android.text.InputType;
import android.util.Log;
import android.view.Gravity;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.Locale;

import tez.emrebicek.sesliderstakip.helper.BaglantiKontrolu;
import tez.emrebicek.sesliderstakip.helper.Veritabani;

public class KonusmaEkrani extends Activity {
    Bundle gelenveriler = null;
    Connection conn=null;
    BaglantiKontrolu internet;
    Veritabani vt;
}

```

```

TextView tvAdiSoyadi, tvDersAdi;
Button btnBaslat, btnBitir, btnTemizle, btnHocaEkranı;
EditText etKonusmalar;
String konusma = "";
final int KodKanali=100;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
        WindowManager.LayoutParams.FLAG_FULLSCREEN);

    requestWindowFeature(Window.FEATURE_NO_TITLE);
    setContentView(R.layout.konusmaekrani);

    VeritabaniBaglanti();

    init();
}

private void VeritabaniBaglanti(){
    internet = new BaglantiKontrolu(getApplicationContext());
    if(internet.isConnectingToInternet()){
        vt =new Veritabani(getApplicationContext());
        conn=vt.veritabaniAc();
    }
}

private void init() {
    gelenveriler = getIntent().getExtras();
    tvAdiSoyadi = (TextView) findViewById(R.id.keTVHocaAdiSoyadi);
    tvAdiSoyadi.setText(gelenveriler.getString("Hoca_Adi")+"
"+gelenveriler.getString("Hoca_Soyadi"));
    tvDersAdi = (TextView) findViewById(R.id.keTVDersAdi);
    tvDersAdi.setText(gelenveriler.getString("Ders_Adi"));
    etKonusmalar = (EditText) findViewById(R.id.keETDersIcerik);
    konusma = "";
    etKonusmalar.setText("");

    btnBaslat = (Button) findViewById(R.id.keBTNBaslat);
    btnBaslat.setOnClickListener(new KonusmayaBasla());
    btnBitir = (Button) findViewById(R.id.keBTNBitir);
    btnBitir.setOnClickListener(new KonusmayiBitir());
    btnTemizle = (Button) findViewById(R.id.keBTNTumunuTemizle);
    btnTemizle.setOnClickListener(new KonusmayiTemizle());
    btnHocaEkranı = (Button)
findViewById(R.id.keBTNHocaEkranıGeriDon);
    btnHocaEkranı.setOnClickListener(new HocaEkranınaGeriDon());

    KonusmayiCek();
}

private void KonusmayiCek() {
    try {
        if(internet.isConnectingToInternet()){
            Statement query = conn.createStatement();

```

```

        Log.d("dersid", gelenveriler.getString("Ders_Id"));
        ResultSet kayitlar = query.executeQuery("SELECT
dersicerik FROM tbldersler WHERE
id="+gelenveriler.getString("Ders_Id")+" AND
hocaId="+gelenveriler.getString("Hoca_Id")+" AND aktifmi=1 ORDER BY id
DESC LIMIT 0,1;");
        if(kayitlar.next()){
            konusma = kayitlar.getString("dersicerik");
            etKonusmalar.setText(konusma);
        }
        else{

MesajGoster(getApplicationContext(),getString(R.string.internetyok),
Gravity.CENTER,0,0);
        }
    }catch (SQLException e){

MesajGoster(getApplicationContext(),e.getMessage().toString(),
Gravity.CENTER,0,0);
    }
}

private class KonusmayaBasla implements View.OnClickListener {
    @Override
    public void onClick(View v) {
        Basla();
    }
}

private void Basla() {

    Intent intent = new
Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);

intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,RecognizerIntent
.LANGUAGE_MODEL_FREE_FORM);

    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE,
Locale.getDefault());

intent.putExtra(RecognizerIntent.EXTRA_PROMPT,getString(R.string.kekon
usmakomutu));

intent.putExtra(RecognizerIntent.EXTRA_SPEECH_INPUT_COMPLETE_SILENCE_L
ENGTH_MILLIS, 5000);

intent.putExtra(RecognizerIntent.EXTRA_SPEECH_INPUT_MINIMUM_LENGTH_MIL
LIS, new Long(10000));
intent.putExtra(RecognizerIntent.EXTRA_SPEECH_INPUT_POSSIBLY_COMPLETE_
SILENCE_LENGTH_MILLIS, 5000);

    intent.putExtra(RecognizerIntent.EXTRA_MAX_RESULTS, 3);
    try {
        startActivityForResult(intent, KodKanali);
    } catch (ActivityNotFoundException a) {
        Toast.makeText(getApplicationContext(),

```

```

a.getMessage().toString(),
    Toast.LENGTH_SHORT).show();
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    switch (requestCode){
        case KodKanali:
            if(resultCode==RESULT_OK && null!=data){
                ArrayList<String>
sonuc=data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
                konusma+=sonuc.get(0)+" ";
                konusma = konusma.replace("null", "");
                etKonusmalar.setText(konusma);
                KonusmayiVeritabaninaKaydet(konusma);
                Basla();
            }
            break;
    }
}

private class KonusmayiBitir implements View.OnClickListener {
    @Override
    public void onClick(View v) {
        if(konusma.length()>0){
            String tumkonusmalar =
etKonusmalar.getText().toString();
            KonusmayiVeritabaninaKaydet(tumkonusmalar);
        }
        else{
            MesajGoster(getApplicationContext(),"Boş içerik
kaydedilemez.Lütfen biraz konuşun.", Gravity.CENTER,0,0);
        }
    }
}

private void KonusmayiVeritabaninaKaydet(String icerik){
    try {
        if(internet.isConnectingToInternet()){
            Statement query = conn.createStatement();
            boolean sonuc = query.execute("UPDATE tbldersler set
dersicerik='"+ icerik + "' WHERE
id="+gelenveriler.getString("Ders_Id")+" AND hocaId=" +
gelenveriler.getString("Hoca_Id") + " AND aktifmi=1;");
            if(!sonuc){
                MesajGoster(getApplicationContext(),
"Konuşmalarınız veritabanına başarıyla kaydedilmiştir.",
Gravity.CENTER, 0, 0);
            }
            else{
                MesajGoster(getApplicationContext(),"Konuşmalarınız veritabanına
kaydedilemedi.", Gravity.CENTER,0,0);
            }
        }
    }
}

```

```

    }
    else{

MesajGoster(getApplicationContext(),getString(R.string.internetyok),
Gravity.CENTER,0,0);
    }
    }catch (SQLException e){

MesajGoster(getApplicationContext(),e.getMessage().toString(),
Gravity.CENTER,0,0);
    }
}

private class KonusmayiTemizle implements View.OnClickListener {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        AlertDialog.Builder onay= new
AlertDialog.Builder(KonusmaEkrani.this);
        onay.setMessage("Konuşmalarınız hem veritabanından hem de
canlı ekran akışından silinecek.Onaylıyor musunuz?");
        onay.setTitle("Uyarı");
        onay.setNegativeButton("Hayır", null);
        onay.setPositiveButton("Evet", null);
        onay.setCancelable(true);

        onay.setPositiveButton("Evet", new
DialogInterface.OnClickListener() {

            @Override
            public void onClick(DialogInterface dialog, int which)
{
                // TODO Auto-generated method stub
                KonusmayiSil();
            }
        });

        onay.setNegativeButton("Hayır", new
DialogInterface.OnClickListener() {

            @Override
            public void onClick(DialogInterface dialog, int which)
{
                // TODO Auto-generated method stub
                MesajGoster(getApplicationContext(),"Silme işlemi
kullanıcı tarafından iptal edildi.",Gravity.CENTER,0,0);
            }
        });

        onay.create().show();
    }
}

private void KonusmayiSil() {
    try {
        if(internet.isConnectingToInternet()){
            Statement query = conn.createStatement();

```



```

        boolean sonuc = query.execute("UPDATE tbldersler set
dersicerik='' WHERE id=" + gelenveriler.getString("Ders_Id") + " AND
hocaaid = "+gelenveriler.getString("Hoca_Id")+" AND aktifmi=1;");
        if(!sonuc){

MesajGoster(getApplicationContext(),"Konuşmalarınız hem veritabanından
hem de canlı ekran akışından silinmiştir.", Gravity.CENTER,0,0);
            konusma = "";
            etKonusmalar.setText("");
        }
        else{

MesajGoster(getApplicationContext(),"Konuşmalarınız silinemedi.Lütfen
tekrar deneeyiniz.", Gravity.CENTER,0,0);
        }
        }
        else{

MesajGoster(getApplicationContext(),getString(R.string.internetyok),
Gravity.CENTER,0,0);
        }
    }catch (SQLException e){

MesajGoster(getApplicationContext(),e.getMessage().toString(),
Gravity.CENTER,0,0);
    }
}

private class HocaEkraninaGeriDon implements View.OnClickListener
{
    @Override
    public void onClick(View v) {
        Intent ii = new
Intent(KonusmaEkrani.this,HocaEkrani.class);
        Bundle bb = new Bundle();
        bb.putString("Hoca_Id",
gelenveriler.getString("Hoca_Id"));
        bb.putString("Hoca_Adi",
gelenveriler.getString("Hoca_Adi"));
        bb.putString("Hoca_Soyadi",
gelenveriler.getString("Hoca_Soyadi"));
        ii.putExtras(bb);
        startActivity(ii);
        finish();
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // TODO Auto-generated method stub
    super.onCreateOptionsMenu(menu);
    MenuInflater m=getMenuInflater();
    m.inflate(R.menu.menu, menu);
    return super.onCreateOptionsMenu(menu);
}

@Override

```

```

public boolean onOptionsItemSelected(MenuItem item) {
    // TODO Auto-generated method stub
    switch (item.getItemId()) {
        case R.id.dersyonetimsistemi:
            DersYonetimSistemi();
            return true;
        case R.id.dersanlat:
            DersAnlat();
            return true;
        case R.id.canlid ders:
            CanliDersEkrani();
            return true;
        case R.id.hakkında:
            HakkındaEkrani();
            return true;
        case R.id.cikis:
            CikisIslemi();
            return true;
    }
    return false;
}

private void HakkındaEkrani() {
    // TODO Auto-generated method stub
    final AlertDialog.Builder alert_dlg = new
AlertDialog.Builder(KonusmaEkrani.this);
    alert_dlg.setTitle("Hakkında");
    alert_dlg.setMessage(getString(R.string.hakkında));

    alert_dlg.setNegativeButton(getString(R.string.tamam),
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int
id) {
                dialog.cancel();
            }
        });

    alert_dlg.setCancelable(true);
    AlertDialog alertDialog = alert_dlg.create();
    alertDialog.show();
}

private void CikisIslemi() {
    AlertDialog.Builder builder = new
AlertDialog.Builder(KonusmaEkrani.this);
    builder.setTitle("Çıkış");
    builder.setMessage(getString(R.string.uygulamaparmastegi));
    builder.setCancelable(false);
    builder.setPositiveButton(getString(R.string.yevet),
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int
id) {
                Intent homeIntent = new
Intent(Intent.ACTION_MAIN);
                homeIntent.addCategory(Intent.CATEGORY_HOME);

                homeIntent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            }
        });
}

```

```

        startActivity(homeIntent);
        finish();
        System.exit(1);
    }
});
builder.setNegativeButton(getString(R.string.hayir),
    new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int
id) {
            Toast.makeText(getApplicationContext(),
getString(R.string.iptal), Toast.LENGTH_SHORT).show();
            dialog.cancel();
        }
    });
AlertDialog alertDialog = builder.create();
alertDialog.show();
}

private void CanliDersEkrani() {
    AlertDialog.Builder builder = new
AlertDialog.Builder(KonusmaEkrani.this);
    builder.setTitle(getString(R.string.sekod));

    final EditText input = new EditText(KonusmaEkrani.this);
    String dKodu=gelenveriler.getString("Ders_Adi");
    String[] kodlar = dKodu.trim().split("-");
    input.setText(kodlar[1]);
    input.setInputType(InputType.TYPE_TEXT_FLAG_CAP_CHARACTERS);
    builder.setView(input);

    builder.setPositiveButton(getString(R.string.tamam), new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            String Kod = input.getText().toString();

            Intent i = new Intent(KonusmaEkrani.this,
tez.emrebicek.sesliderstakip.CanliDers.class);
            Bundle b = new Bundle();
            b.putString("CanliDersKodu", Kod);
            i.putExtras(b);
            startActivity(i);
        }
    });
    builder.setNegativeButton(getString(R.string.iptalet), new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.cancel();
        }
    });
    builder.show();
}

private void DersAnlat() {

```

```

        Intent iii = new Intent(KonusmaEkrani.this, HocaEkrani.class);
        Bundle bbb = new Bundle();
        bbb.putString("Hoca_Id", gelenveriler.getString("Hoca_Id"));
        bbb.putString("Hoca_Adi", gelenveriler.getString("Hoca_Adi"));
        bbb.putString("Hoca_Soyadi",
gelenveriler.getString("Hoca_Soyadi"));
        iii.putExtras(bbb);
        startActivity(iii);
        finish();
    }

    private void DersYonetimSistemi() {
        startActivity(new
Intent(KonusmaEkrani.this, DersYonetimSistemi.class));
    }

    private void MesajGoster(Context c, String s, int g, int x, int y){
        Toast t = Toast.makeText(c, s, Toast.LENGTH_LONG);
        t.setGravity(g, x, y);
        t.show();
    }
}

```

### SecimEkrani.java

```

package tez.emrebicek.sesliderstakip;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.text.InputType;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.EditText;

public class SecimEkrani extends Activity {
    private Button btnDersAnlat, btnCanliDers;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);

        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.secimekrani);

        init();
    }
}

```

```

    }

    private void init() {
        btnDersAnlat = (Button) findViewById(R.id.seBTNDersAnlat);
        btnDersAnlat.setOnClickListener(new DersAnlat());

        btnCanliDers = (Button) findViewById(R.id.seBTNCanliDers);
        btnCanliDers.setOnClickListener(new CanliDers());
    }

    private class DersAnlat implements View.OnClickListener {
        @Override
        public void onClick(View v) {
            startActivity(new
Intent(SecimEkrani.this, GirisEkrani.class));
            finish();
        }
    }

    private class CanliDers implements View.OnClickListener {
        String Kod="";
        @Override
        public void onClick(View v) {
            AlertDialog.Builder builder = new
AlertDialog.Builder(SecimEkrani.this);
            builder.setTitle(getString(R.string.sekod));

            final EditText input = new EditText(SecimEkrani.this);

            input.setInputType(InputType.TYPE_TEXT_FLAG_CAP_CHARACTERS);
            builder.setView(input);

            builder.setPositiveButton(getString(R.string.tamam), new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which)
            {
                Kod = input.getText().toString();

                Intent i = new Intent(SecimEkrani.this,
tez.emrebicek.sesliderstakip.CanliDers.class);
                Bundle b = new Bundle();
                b.putString("CanliDersKodu", Kod);
                i.putExtras(b);
                startActivity(i);
            }
        });
        builder.setNegativeButton(getString(R.string.ipalet), new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which)
            {
                dialog.cancel();
            }
        });

        builder.show();
    }

```

```

    }
}

```

## Splash.java

```

package tez.emrebicek.sesliderstakip;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.Window;
import android.view.WindowManager;
import android.widget.TextView;

import java.util.Timer;
import java.util.TimerTask;

public class Splash extends Activity {
    private int sayac = 5;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);

        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.splash);

        Yonlendirme();
    }
    private void Yonlendirme() {
        final TextView tvSure = (TextView) findViewById(R.id.tvSure);
        final Handler timer = new Handler();
        final TimerTask mTimerTask;
        Timer t = new Timer();
        mTimerTask = new TimerTask() {
            public void run() {
                timer.post(new Runnable() {
                    public void run() {
                        if(sayac < 0){
                            cancel();
                            startActivity(new
Intent(Splash.this, SecimEkrani.class));
                            finish();
                            sayac = 5;
                        }
                        tvSure.setText(String.valueOf(sayac));
                        sayac--;
                    }
                });
            }
        };
    }
}

```

```

        }};

        t.schedule(mTimerTask, 100, 1000); //
    }
}

```

### BaglantiKontrolu.java

```

package tez.emrebicek.sesliderstakip.helper;

import android.content.Context;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;

public class BaglantiKontrolu {

    private Context _context;

    public BaglantiKontrolu(Context context){
        this._context = context;
    }

    public boolean isConnectingToInternet(){
        ConnectivityManager connectivity = (ConnectivityManager)
        _context.getSystemService(Context.CONNECTIVITY_SERVICE);
        if (connectivity != null)
        {
            NetworkInfo[] info = connectivity.getAllNetworkInfo();
            if (info != null)
                for (int i = 0; i < info.length; i++)
                    if (info[i].getState() ==
NetworkInfo.State.CONNECTED)
                        {
                            return true;
                        }
        }
        return false;
    }
}

```

### Veritabani.java

```

package tez.emrebicek.sesliderstakip.helper;

import android.content.Context;
import android.os.StrictMode;
import android.view.Gravity;
import android.widget.Toast;

```

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class Veritabani {
    Connection conn = null;
    private static final String DB_DRIVER = "com.mysql.jdbc.Driver";
    private static final String DB_CONNECTION =
"jdbc:mysql://sdyt.yyu.edu.tr:3306/admin***?useUnicode=yes&characterEn
coding=UTF-8";
    Context a;
    public Veritabani(Context c) {
        this.a= c;
    }
    private void veritabaniBaglanti() {
        conn = null;
        StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder()
            .permitAll().build();
        StrictMode.setThreadPolicy(policy);
        try {
            Class.forName(DB_DRIVER);
            try {
                conn =
DriverManager.getConnection(DB_CONNECTION,"admin***","sdyt***");
            } catch (SQLException e){
                MesajGoster(a,"Veritabani Bağlantı Hatası",
Gravity.CENTER, 0, 0);
            }
        } catch (Exception e) {
            MesajGoster(a,"Veritabani Bağlantı Hatası",
Gravity.CENTER, 0, 0);
        }
    }

    public Connection veritabaniAc(){
        veritabaniBaglanti();
        return conn;
    }

    public void veritabaniKapat(){
        try {
            conn.close();
        } catch (SQLException e) {
            MesajGoster(a,"Veritabani bağlantı hatası",
Gravity.CENTER, 0, 0);
        }
    }

    private void MesajGoster(Context c,String s, int g, int x, int y){
        Toast t = Toast.makeText(c, s, Toast.LENGTH_LONG);
        t.setGravity(g,x,y);
        t.show();
    }
}

```



## **ÖZGEÇMİŞ**

1985 yılında Van’da doğdu. İlk, orta ve lise öğrenimini Van’da tamamladı. 2005 yılında öğrenime başladığı Bahçeşehir Üniversitesi Bilgisayar Mühendisliği Bölümü’nden 2010 yılında mezun oldu. Mezuniyetinin ardından askerlik görevini tamamladı. 2011 yılında Ankara’daki bir yazılım şirketinde bir yıl süre ile çalıştı. 2012 yılında halen devam ettiği Yüzüncü Yıl Üniversitesi Enformatik Bölümü’nde Araştırma Görevlisi olarak çalışmaya başladı. 2013 yılında Yüzüncü Yıl Üniversitesi Elektrik-Elektronik Mühendisliği Anabilim Dalı’nda Yüksek Lisans eğitimine başladı. Evli ve iyi derecede İngilizce bilmektedir.