

## Güvenlik Görevlileri İçin Bulut Bilişim Destekli, Ses Arayüzü ile Çalışan Asistan Sistemi

Yaşar Yaşa<sup>1</sup>, Mehmet Göktürk<sup>2</sup>

<sup>1</sup> Gebze Yüksek Teknoloji Enstitüsü, Kocaeli

<sup>2</sup> Gebze Yüksek Teknoloji Enstitüsü, Kocaeli

yasar.yasa@hotmail.com, gokturkmehmet@gmail.com

**Özet:** İnsanlar, mobil cihazlarını ses komutları ile yönetebilmeye yoğun bir ilgi göstermişlerdir. Bu ilginin sebeplerinden birisi, küçük klavyede yazı yazmaktan hoşlanmamaları ve diğeri de gündelik yaşam işlerini devam ettiren, elleri ve gözleri cihazlarına bağımlı olmadan telefonlarını kullanmak istemeleridir. Bu çalışmada, güvenlik görevlilerinin ellerinden ve gözlerinden bağımsız olarak kullanabilecekleri T.C. kimlik numarası ve plaka sorgulama sistemi ses ara yüzü geliştirme prensiplerine göre geliştirilmiştir. Bu çalışmanın amacı, bu ve benzer pozisyonlar için sesli etkileşimde arayüz prensiplerini ortaya çıkarmaktır. El ve göz bağımsızlığından yola çıkılarak ses ara yüzü prensipleri ile geliştirilen uygulama devamlı olarak kullanıcılar ile test edildi ve alınan geri bildirimlere göre kullanılabilirlik açısından iyileştirmeler yapıldı ve iyi bir ses arayüzü uygulaması ortaya konulmaya çalışıldı.

**Anahtar Sözcükler:** Ses kullanıcı ara yüzü, ses tanıma, metinden ses çevrimi, android

**Abstract:** People have shown an intense interest in mobile devices to be managed through voice commands. One of the reasons is that, they don't like using small keyboards for typing and other reason is that, they want to continue to their daily life while they are using their mobile devices without being hands and eyes dependent. In this study, a system which security guards can use it to query identity number or plate license without being hands and eyes dependent, was developed according to voice user interface design principles. This system was always tested with the users and developed its features according to feedbacks from users. So that an optimal voice user interface solution was tried to be found out.

**Keywords:** Voice user interface, speech recognition, text-to-speech, android

### 1. Giriş

İnsanlar, mobil cihazlarını ses komutları ile yönetebilmeye yoğun bir ilgi göstermişlerdir. Bu ilginin sebeplerinden birisi, küçük klavyede yazı yazmaktan hoşlanmamaları ve diğeri de gündelik yaşam işlerini devam ettiren, elleri ve gözleri telefona bağımlı olmadan telefonlarını kullanmak istemeleridir. Örnek olarak, bir kişinin arabasını sürerken telefonunu ses komutları ile yönetip yeni mesaj gönderebilmesini düşünebiliriz.

Ses algılamanın yukarıda bahsedilen özellikleri sayesinde, güvenlik görevlileri için bu özellikleri kullanarak bir çalışma yapıldı. Kimlik veya plaka sorgulama sırasında kullandıkları cihazlar onların el ve göz bağımsızlığını ihlal etmektedir ve bu da onların güvenliklerini tehlikeye atmaktadır[11]. Yapılan çalışma, güvenlik görevlileri için kimlik ve plaka sorgulamanın ses komutları ile mobil cihazları üzerinden yapılmasını sağlamaktadır. Bu çalışma sayesinde güvenlik görevlileri, elleri ve gözleri bağımlı olmayacak bir şekilde sorgulama işlemlerini ses yardımı ile diğer işlerini yaparken de gerçekleştirebilirler. Çalışma geliştirilen uygulama, Ses Kullanıcı Ara yüzü(SKA) prensipleri temel alınarak geliştirildi.

SKA, kullanıcının sesini girdi olarak alan ve sonucunda da önceden kaydedilmiş sesleri veya sentezlenmiş(robotik) sesleri çıktı olarak veren bir

kullanıcı arayüzü tipidir. Ses ara yüzü üzerine yapılmış birçok çalışma bulunmaktadır[8, 9, 10, 13]. Fakat bu çalışmaların hepsi, bir telefon numarası ile bağlantılı işlem yapılabilen ve el-göz bağımsızlığını tamamen kaldırmayan çalışmalardır.

Ses ara yüzü için gerekli olan ses algılama ve ses çıktısı üretme işlemleri için geliştirilmiş birçok kütüphane bulunmaktadır. Dünyada çok popüler olan ve açık kaynak olan Android işletim sistemi, kendi içinde tümleşik olarak ses tanıma özelliğini bulundurmaktadır[1]. Yapılan çalışma Android işletim sistemli mobil cihazlar için geliştirildiğinden, Android ses tanıma ara yüzleri kullanılmıştır.

Google tarafından sunulan ses tanıma ara yüzleri bulut bilişim desteklidir. Bulut bilişim, kullanılan alan, yazılım, hizmet gibi bilgilerin uzaktan da kullanılabilmesidir. Yapılan çalışmada Google'ın sunduğu ses tanıma hizmetini kullanıldı.

Geliştiriciler için uygulama geliştirme ara yüzleri (API) sunan Android işletim sistemi, ses girdilerinin alınıp, ses çıktıları üretilebilmesini sağlamaktadır. Android ses tanıma sisteminin çalışma mantığı kısaca şu şekildedir.

1. Kullanıcıdan alınan ses girdisi, sentezlenerek metne çevrilmek için Google Sunucu'larına gönderilir. Sunucuların nerede olduğu veya nasıl haberleşmek gerektiği kısımları tamamen

kütüphane tarafından çözümlenir.

2. Sunucularda sentezlenen ses için, olabilecek en yakın metin çıktılarını tekrar uygulamaya gönderilir. Kullanıcının hangi metni istediğini bulmak veya seçmek geliştiriciye kalmıştır.

Android işletim sisteminde, metin-ses çeviri işlemlerinin çevrimdışı yapılabilmesi de mümkündür. Çevrimdışı özelliği için fazladan dil paketine ihtiyaç vardır. Google, birçok dil için böyle bir destek vermesine rağmen Türkçe için maalesef çevrimdışı dil desteği yoktur. Türkçe metin-ses sentezlemesi çalışmaları yapılmış[5] ve farklı şirketlerin bu alanda ürünleri de bulunmaktadır[6,7]. Uygulamada denenilen ve en iyi sonuç alınan SVOX<sup>26</sup>, Türkçe dil desteği olan en iyi çalışmalardandır. Gerçeklenen uygulama hem SVOX hem de Android API ile çalışabilmektedir.

Makalenin sonraki kısımlarında çalışma için gerçekleştirilen adımlar anlatılmaktadır. 2. bölümde SKA uygulaması geliştirirken dikkat edilmesi gereken konulardan bahsedilecektir. Bu konular ışığında bu çalışma için gerçekleştirilen uygulamanın mimari yapısı anlatılacak ve geliştirme esnasında uygulanan özel yöntemlerden 3. bölümde bahsedilecektir. Son olarak yapılan çalışma için, sonuç bölümü yer alacaktır.

## 2. SKA Uygulaması Geliştirmek İçin Genel Hususlar

Ses arayüzü uygulaması gerçekleştirilmeden önce düşünülmesi gereken ilk konu, gerçekleştirilmek istenen görevin ses arayüzüne uygun olup olmadığıdır[2]. Bunun sebebi ise, ses girdi olarak alınırken oluşabilecek olumsuz durumlardır. Bu durumları aşağıdaki gibi sıralayabiliriz[1,2].

1. Ses tanımanın aktif hale gelmesi, kullanıcıdan ses girdisinin alınması ve bu girdinin metine çevrilmesi için yanıtın beklenmesi.
2. Ses tanımanın başarısız olduğu durumlar azımsanmayacak kadar çok olabilir ve kullanıcıdan tekrar aynı girdi istemek zorunda kalınabilir.
3. Ses geçicidir ve kolay unutulur. Dolayısıyla, kullanıcı söylediği komutları veya uygulamadan alınan yanıtları kolaylıkla unutabilir.

Yukarıda SKA ile uygulama geliştirilmesinin dezavantajlarından kısaca bahsedilmiştir.

Bu dezavantajları sebebi ile her görevin SKA ile geliştirilmesi uygun olmayabilir. SKA ile geliştirilmesi uygun olan görevleri şu şekilde sıralayabiliriz.

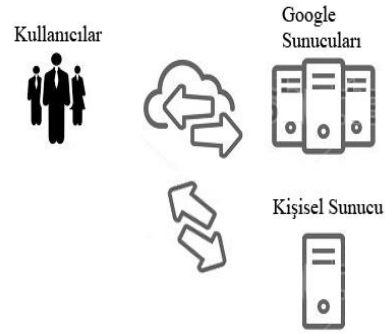
1. Eğer kullanıcının elleri ve gözleri bağımlı/meşgul durumda ise ses arayüzü uygulaması burada yapılacak işlemler için uygun olabilir. Örnek

olarak araba sürerken maillerin okunması diyebiliriz.

2. Grafıksel Kullanıcı Ara yüzlerinin(GKA) kullanımının daha genel tercih edilmesine rağmen[12], SKA'nın da GKA girdilerine göre daha fazla avantaj sağladığı durumlar vardır. Buna örnek olarak da, yaklaşık 100 kadar elemanı bulunan listeden bir tanesinin seçilmesi klasik GKA'da zaman alan işlemdir. SKA uygulamasında ise bu durum söz konusu değildir.

## 3. Uygulama Mimarisi ve Geliştirilmesi

Geliştirilen uygulama açık kaynak olan Android işletim sistemli cihazlar üzerinde çalışmakta ve Android ses algılama ara yüzlerini kullanmaktadır. Önceki konularda da bahsedildiği gibi Android Api ses sentezleri için Google sunucularını kullanmaktadır. Google sunucularından alınan sonuçlarla işlem yapan uygulama son olarak kimlik ve plaka sorgulamalarını yapmak için kişisel sunucuyu kullanmakta ve sorgulama sonuçlarını buradan almaktadır. Şekil-1 uygulamanın genel mimarisini göstermektedir.



Şekil-1: Uygulama mimarisi

### 3.1 Ses Komutları İçin Uygun Görevlerin Belirlenmesi

Her görev ses ara yüzü ile gerçekleştirilmek için uygun değildir[3]. Bu sebepten çalışmada öncelikle, uygulamanın ne tip ses aksiyonları içereceği belirlendi[2]. Tek yönlü komutlar olabileceği gibi(sadece tek bir defa ses girdisi içeren komutlar), ileri ve geri gibi kullanıcıdan tekrar girdi istenebilecek, çift yönlü komut dizisi de olabilir. Yapılan çalışmada, menü yapısı olması gerektiği için çift yönlü bir yapı kullanılmıştır. Örnek olarak kullanıcıdan ilk aşamada ne tip sorgulama yapmak istediğini ses girdisi olarak alınır, sonraki adımda o adıma özel tekrar ses girdileri alınır ve yanıtlar sağlanır. Android arayüzlerinin sağladığı yapı tek yönlü yapı temellidir (bir ses girdisi alınır ve sonlanır). Çift yönlü bir uygulama tasarlandığı için, buna uygun bir mimari geliştirmek durumunda kalınmıştır.

Bir sonraki adım ise konuşma diyaloglarının belirlenmesidir. Bu adımda kullanıcının ne söyleyeceği(ses komutları ile alınacak girdi), ve

<sup>26</sup><http://svoxmobilevoices.wordpress.com/>

buna karşılık olarak uygulamanın nasıl, ne zaman cevap vereceği kararlaştırılmıştır. Bu işlemi akış mimarisi olarak adlandırabiliriz. Kullanıcıyı sıkımsamak ve uygulamanın kullanılabilir olması için, her adımda kullanıcıya ayrıntılı çıktı verilmelidir. Örnek olarak, ses algılama yapılırken eğer ses algılanamadıysa, sebep olarak neden algılanmadığı kullanıcıya belirtilmiştir. Bu kısım Android arayüzlerinin sağladığı sesten metine çevirme işlemi sırasında gelen sonuçların güvenilirlik değerlerine göre çeşitli ses çıktıları ile sağlanmıştır. Tablo-1 üretilen ses çıktıların özetlemektedir.

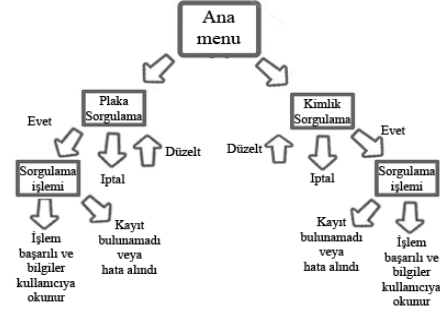
Sentez sonucu	Güvenilirlik aralığı	Kullanıcı bildirimi
Başarısız ses algılama	0.0 ile 0.3	Kusura bakmayın anlayamadım çünkü iyi duymadım.
Metin çevirisi yapılamaması	0.3 ile 0.9	Kusura bakmayın anlayamadım.
Komutlarda bulunamaması	0.9 ile 1.0	Böyle bir komut bulunmuyor

**Tablo-1:** Güvenilirlik tablosu

### 3.2 Kullanıcıdan Alınacak Girdilerin Sınırlanması

Doğal dilin bir sınırı yoktur. Dolayısıyla çok farklı ve sınırsız varyasyonlu girdiler alınabilir. Fakat bu durum uygulama için algılama işlemini çok zor hale getirmektedir. Uygulama geliştirilirken bu durum göz önünde bulundurulmuştur, girdinin karmaşıklığı belirlenmiş ve kullanıcının kullanabileceği komutlar sınırlandırılmıştır. Ses girdilerinin sınırlandırılması, sesin algılanma işleminin doğruluk yüzdesini dolayısıyla hem arttırmakta hem de işlemin daha çabuk olmasını sağlamaktadır. Örnek olarak, kullanıcıdan onay için beklenen bir girdinin : “tamam”, “evet”, “yapılsın”, “kesinlikle yap” gibi farklı seçenekler yerine tek bir “evet” komutu kabul edilmektedir (edilmiştir).

Çalışmada kullanılan ses komutları ve akış şeması Şekil-2’de basitçe belirtilmiştir.



**Şekil-2:** Uygulamanın akış şeması

### 3.3 Kullanıcının Eğitilmesi

SKA doğası gereği GKA’dan farklıdır. GKA’da kullanıcının alabileceği aksiyonlar görsel olarak ve kalıcı halde bulunmaktadır, fakat SKA’da kullanıcı ne söyleyeceğini ve uygulamanın ne anladığını bilmez yada hepsini hatırlayamaz. Özellikle kullanıcıdan alınan ses girdilerini sınırlandırıldığı için de, kullanıcının komutlar konusunda eğitilmesi gerekmektedir. Aksi halde, ne söyleyeceğini, nasıl cevap vereceğini bilemeyebilir veya o an için hatırlayamayabilir. Eğitim için farklı stratejiler güdülebilir. Bunlara örnek vermek gerekirse;

- Komutların görsel olarak listelenmesi. Kullanıcı kullanacağı komutları görsel ve kalıcı olarak görebilir. Bu yöntem Google Play Store<sup>27</sup>’de da bulabileceğiniz çoğu uygulamanın kullandığı bir yöntemdir. Eğer komut sayınız çok fazla ise bu yöntem şu ana kadar bulunabilmiş ve kullanılan en iyi yöntemdir. Yapılan çalışmada, yardım menüsünden kullanıcı bütün komutları görebilir ve ayrıntılı bilgi alabilir.
- Kullanıcının uygulama tarafından yönlendirilmesi. Bu yöntemde uygulama kullanıcıyı yönlendirmek zorundadır, kullanıcı ne yapacağını bilmediği durumlarda sesli olarak ona söyleyebileceği komutlar tekrarlanabilir. Bu şekilde kullanıcı hiçbir GUI ile karşılaşmaz ve uygulama tamamen ses tabanlı olmuş olur. Örnek olarak : Uygulamanın, “Kimlik sorgulama için KİMLİK demeniz yeterli” şeklinde kullanıcıya bilgilendirici bir söylem çıkartması, kullanıcının hangi komutları kullanabileceğini gösteriyor.

Yukarıda bahsedilen yöntemlerin ikisi de yapılan çalışmada kullanılmıştır. İki yöntemi birbirinden farklı yöntemler olarak kullanmak yerine, tamamlayıcı olarak kullanıp, uygulamanın kullanılabilirliğini arttırma amaçlandı.

### 3.4 Komutların Kullanıcıya Geri Bildirimi

Sesi girdi olarak kullanmak riskli bir işlemdir, çünkü algılama sırasında çok fazla hata çıkabilir. GKA’da

<sup>27</sup> <https://play.google.com/>

“İptal” butonu yerine “Tamam” butonuna basma veya ikisine birden basamama ihtimali çok düşüktür. Fakat bu oran SKA uygulamalarında oldukça yüksektir. Özellikle ses olarak birbirlerine yakın komutlar kullanılmışsa, komutların birbirine karışma durumu oldukça fazla olmaktadır. Karışma veya yanlış algılama durumlarının kullanıcıya geri bildirimler yaparak en aza indirilmesi sağlanmıştır. Örnek olarak : Uygulamanın T.C. kimlik no ile sorgulama yapılması istensin. Kullanıcıdan 11 haneli kimlik numarası alındıktan sonra, sesin yanlış algılanma veya kullanıcının kimlik numarasını yanlış söylemesi ihtimaline karşı kullanıcıya kimlik numarası tekrar okunur ve eğer kimlik numarası kullanıcı tarafından doğrulanır ise bu işlem sonunda devam edilmesi istenir. Bu şekilde yanlış sorgulama yapılmasından kurtulmuş olundu.

### 3.5 Kullanılacak Komutların Tipinin Belirlenmesi

SKA uygulamaları için farklı şekillerde komut algılama işlemi yapılabilir. Bunlardan bazıları aşağıdaki gibidir :

1. Sabit komutlar : Bu komutlar sabit ve değişmezdirler. Uygulamada tutulma yerleri ise tasarıma göre değişmektedir. Eğer işlenecek fazla sayıda komut varsa veritabanında saklanıp oradan sorgulama yapılabilir. Eğer az sayıda komut varsa hafızada tutup sorgulama işlemi oradan hafızadan yapılabilir. Örnek olarak : “Kimlik sorgula” cümlesindeki “Kimlik” bizim için sabit bir komuttur.
2. Serbest komutlar : Bu tip komutlar sabit komutlara göre daha karmaşıktır. Uygulamada serbest komutlar kullanılacaksa bir gramer yapısı olmalıdır. Örnek olarak : “12345678921 kimlik numarasını sorgula” örnek bir serbest komut zinciri olabilir. Burada “kimlik” yine komut olarak algılanabilir fakat, girdi olarak alınan cümle içinde kimlik numarasının bulunması ve doğru algılanabilmesi zor ve karmaşık bir işlemdir.

Gerçeklenen uygulamada komut sayısının az olması ve uygulamanın basit tutulması amacı ile sabit komutlar kullanılmıştır.

### 3.6 Ses Girdilerinden Komutların Bulunması

Kullanıcıdan alınan ses girdilerinden komutların başarılı bir şekilde çıkartılabilmesi uygulamanın başarısı için çok önemlidir. Komutların başarılı bir şekilde algılanamaması, yanlış işlem yapılmasına veya kullanıcıdan tekrar girdi istenmesine sebep olmaktadır. Bu durumda kullanıcı uygulamayı kullanmak istemeyebilir. Yapılan çalışmada ses girdilerinden kelimeler üretildikten sonra alınan girdiler içerisinde

komutların çekilebilmesi için özel yöntemler kullanılmıştır. Bunlar :

1. Kelime Tanıma Algoritması : Bu algoritma ile cümle içinde geçen komutun bulunması sağlanır. Örnek olarak : Kullanıcının “KİMLİK” komutu yerine “KİMLİK SORGULAMA YAPALIM” cümlesini kullandığını düşünelim. Aslında komut “KİMLİK” olmasına rağmen, kelime tanıma algoritması ile “SORGULAMA” ve “YAPALIM” kelimeleri yok sayılır ve “KİMLİK” komutu söylenmiş gibi işlem yapılır.

Kelime tanıma algoritması kullanımı sonrası yapılan testlerde, kullanılabilirliğin arttığı gözlemlenmiş ve kullanıcıların %100’ünün yeni yöntemi daha kullanılabilir bulmuştur.

2. Kelimenin Kökünün Alınması : Bu aşamada kullanıcının komut kelimesini tam olarak değil de ekleri ile birlikte söylemesi durumu da göz önünde bulundurulmuş ve bunun sonucunda kullanılabilirlik artırılmıştır. Alınan girdi kelimelerinin kökünün bulunması ve kelime kökleri ile işlem yapılması hem uygulamanın başarısını arttırmıştır hem de kullanıcının daha özgür olmasını sağlamıştır. Örnek olarak : “KİŞİ” kelimesi uygulamada kimlik sorgulama işlemi için anahtar kelimedir. Fakat kullanıcı bu aşamada “KİŞİYİ SORGULAYALIM” gibi bir girdi verdiği zaman dahi, uygulamanın kelime kökleri ile işlem yapmasından dolayı kimlik sorgulama komutunu algılayacak ve ona göre işlem yapacaktır. Yapılan çalışmada Apache projelerinden Snowball[15] ile Türkçe kök bulma[14] işlemleri için kullanılmıştır.

Komutların algılanması sırasında kelimenin köküne göre yapılan algılama işlemi ile söylenen kelimelerden komutların algılanması %65 civarında artmıştır. Bu sonuçlar da hem kelime tanıma algoritması hem de kelime köküne göre komut algılama işleminin başarısını ortaya koymaktadır.

### 3.7 Çok Turlu Ses Girdileri Alma Ve Ses Çıktısı Üretme

Android arayüzleri, sestene metine çevirme ve metinden ses çevirme işlemlerini sadece tek seferlik bir işlem olarak yapılmasına izin vermektedir. Yapılan çalışmada, işlemlerin yapılması için kullanıcı ile uygulama arasında diyalog tarzı bir etkileşim oluşturulmuştur. Aşağıda kimlik sorgulama diyalogunun bir kısmını bulabilirsiniz.

- Uygulama : Lütfen 11 haneli T.C. kimlik numarasını söyleyin.
- Kullanıcı : 12345678932.
- Uygulama : Sorgulamak istediğiniz kimlik

numarası, 12345678932. Sorgulamaya devam etmek istiyor musunuz?

- Kullanıcı : Evet (Hayır, Düzelt)
- Uygulama : Sorgulama yapılıyor.....

Yukarıdaki örnekte ilk konuşmada uygulama ses çıktısı üretmek için metinden sese çevirme özelliğini kullanır. Bu özellik ile girdi olarak verilen metinler ses çıktısına çevrilir ve uygulama bu özelliği kapatır. Sonraki adımlarda ses çıktısı üretmek için bu özelliğin tekrar açılması gerekecektir. Hemen ardından uygulama ses girdisi almak için ses tanıma özelliğini çalıştırır, kullanıcıyı dinler ve sonrasında kullanıcıya tekrar bir ses çıktısı verir. Bahsedilen sırada girdi alıp çıktı üretilebilmesi için kullanıcının uygulamayı tekrar aktif etmesi gerekmekte idi. Fakat kullanıcılardan alınan geri bildirimden tek turlu girdilerin kullanılabilirliği önemli ölçüde düşürdüğü gözlemlenmiştir. Kullanılabilirliğin artırılması için çok turlu girdi mekanizmasının kurulmasına karar verildi. Çok turlu girdi mekanizması ile yapılan araştırma sonucu kullanıcıların tamamının çok turlu yapıyı daha kullanılabilir bulduğu gözlemlenmiştir.

### 3.8 Ses Algılama Özelliğinin Aktif Hale Getirilmesi

Çalışmada son olarak, uygulamanın aktif hale gelmesi konusunda iyileştirme yapılmıştır. Bu aşamadan önce kullanıcı, uygulamayı çalıştırıp sorgulama yapabilmek için, her seferinde ses algılamayı “Başlat” düğmesi ile aktif hale getirmek zorunda kalmaktaydı. Fakat kullanıcıların tamamen GKA ile etkileşimde bulunmaktan kaçındıkları gözlemlendiği için bu probleme de servis aktifleştiriciler ile çözüm geliştirildi. Uygulamada kullanılan iki çeşit aktifleştirici özellik aşağıda listelenmiştir :

1. Ses algılama ile aktifleştirme : Bu özellik için Android işletim sisteminin servis yapısı kullanılmıştır ve uygulama arka planda devamlı olarak kullanıcı dinler hale getirilmiştir. Kullanıcının anahtar kelimeyi söylemesini bekleyen aktifleştirici, kullanıcıdan “ASİSTAN” olarak belirlenen anahtar komut kelimesini algıladığı zaman, sorgulama için geliştirilen uygulamayı aktif hale getirir ve kendisinin ses algılama özelliğini durdurur. Bu işlemin durması sorgulama işleminin sonlanmasına kadar devam eder. Sorgulama işlemi bittikten sonra, tekrar anahtar kelimeyi dinlemeye başlar ve duyana kadar devam eder.

Ses algılama ile aktifleştirme işlemi çok gürültülü olan ortamlarda araya başka seslerin karışması durumunda istenilen başarıya ulaşamamıştır. Özellikle güvenlik görevlilerinin yoğun trafik veya karmaşa ortamında bu özelliği çok verimli kullanamadıkları gözlemlenmiştir ve farklı bir aktifleştirici daha sisteme eklenmiştir.

2. Hareket algılama ile aktifleştirme : Ses algılama ile aktifleştirme servisi, gürültünün az olduğu ortamlarda oldukça başarılı olmuştur. Cihazın hareket sensörlerini kullanarak geliştirilen bu yöntemde, cihaz ani hareket ettirildiği zaman uygulama kullanıcıdan ses girdisi beklemeye başlar. Sorgulama işleminin bu şekilde aktif hale getirildiği bu yöntem, gürültünün fazla olduğu ortamlarda tercih sebebi olabilmektedir.

### 4. Sonuçlar

Bu çalışmada, güvenlik görevlileri için ses ara yüzü ile çalışan bir asistan sistemi geliştirilmiştir. Geliştirilen uygulama üzerinde yapılan araştırmalarla, çalışılan şartlar altında en iyi ses ara yüzü tasarımı bulunmaya çalışıldı. Komut algılama için kullanılan kelime tanıma algoritması ile kelimenin kökünün alınma işlemleri, menü hiyerarşi yapısı, kullanıcıya yapılan sesli geri bildirimler kullanılabilirlik açısından çalışmanın başarısını arttıran etkenler olmuştur.

Çalışma sonucunda kullanıcılardan alınan geri bildirimler, ses ara yüzü ile gerçekleştirilen prototip asistan uygulamasının %80 oranda var olan sisteme göre daha çok tercih edildiğini ortaya koymuştur.

### 5. Kaynaklar

- [1]Greg Milette, Adam Stroud, “Professional Android Sensor Programming” (2012), John Wiley & Sons
- [2]Michael H. Cohen, James P. Giangola, Jennifer Balogh, “Voice User Interface Design” (2004), Addison-Wesley Professional
- [3]Martijn van Welie, Doktora Tezi, “Task based user interface design”, (2001)
- [4]Ikhu-Omoregbe N. A., Azeta A. A. “A Voice-based Mobile Prescription Application for Healthcare Services (VBMOPA)”, International Journal of Electrical & Computer Sciences IJECS-IJENS Vol:10 No:02,(2010)
- [5]Sel, İ., Hanbay, D., Karabatak, M., “Beyin Bilgisayar Arayüzleri için Türkçe Metinden Konuşma Sentezleme Sistemi”, Fırat Üniversitesi Elektrik-Elektronik Bilgisayar Sempozyumu, Bildiri Kitabı II:273-276 (2011).
- [6]<http://www.acapela-group.com/turkish-50-text-to-voice.html>
- [7]<http://www.sestek.com.tr/tts-demo>
- [8]Dr. Mustafa Murat İnceoğlu, “Bir eğitim kurumu için etkileşimli sesli yanıt sistemi”, The Turkish Online Journal of Educational Technology – TOJET April

2004 ISSN: 1303-6521 volume 3 Issue 2 Article 17,  
(2004)

[9]James Raymond Davis, “A voice interface to a Direction giving program”,Media Laboratory, Massachusetts Institute of Technology, (1987)

[10]Rifat Edizkan, Burak Tiryaki, Tunç Büyükcan, İsmail Uzun, Ses Komut Tanıma ile Gezgin Araç Kontrolü, Akademik Bilişim’07 - IX. Akademik Bilişim Konferansı 31 Ocak - 2 Şubat 2007 Dumlupınar Üniversitesi, Kütahya,(2007)

[11]Krisite Krstoski,Doktora Tezi, Handheld Project54speech user interface,University of New Hampshire (Department of Electrical and Computer Engineering), (2005)

[12]Edim Azom Emmanuel, Doktora Tezi, Mobile Phones Interaction Techniques for Second Economy People, (2010)

[13]Dr. Dirk Schnelle-Walk,Voice User Interface Design, Technical University of Darmstadt, Ders Notları

[14]Gülşen Eryiğit , Eşref Adalı, “An Affix Stripping Morphological Analyzer For Turkish”, 2004, s. 299-304, Innsbruck, 03.09.2008 - 03.09.2008

[15]<http://snowball.tartarus.org/algorithms/turkish/stemmer.html>