

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MOBİL CİHAZLAR ÜZERİNDE ÇALIŞAN
SES KULLANICI ARAYÜZÜ UYGULAMALARININ
KULLANILABİLİRLİĞİNİ ARTTIRAN YÖNTEM BELİRLEME

YAŞAR YAŞA
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

GEBZE
2015

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**MOBİL CİHAZLAR ÜZERİNDE ÇALIŞAN
SES KULLANICI ARAYÜZÜ
UYGULAMALARININ
KULLANILABİLİRLİĞİNİ ARTTIRAN
YÖNTEM BELİRLEME**

YAŞAR YAŞA
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMANI
DOÇ. DR. MEHMET GÖKTÜRK

GEBZE
2015

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**DETERMINING METHODS WHICH
INCREASES THE USABILITY OF VOICE
USER INTERFACES ON MOBILE DEVICES**

YAŞAR YAŞA
**A THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF SCIENCE**
DEPARTMENT OF COMPUTER ENGINEERING

THESIS SUPERVISOR
ASSOC. PROF. MEHMET GÖKTÜRK

GEBZE
2015



GTÜ Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 01/06/2015 tarih ve 2015/33 sayılı kararıyla oluşturulan jüri tarafından 03/06/2015 tarihinde tez savunma sınavı yapılan Yaşar YAŞA'nın tez çalışması Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI) : Doç Dr. Mehmet GÖKTÜRK

ÜYE

: Doç. Dr. Hacı Ali MANTAR

ÜYE

: Yrd. Doç. Dr. Mehmet AKTAŞ

ONAY

Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
...../...../..... tarih ve/..... sayılı kararı.

İMZA/MÜHÜR

ÖZET

Günümüzde mobil cihazlar yaygınlaşmakta ve insan hayatının neredeyse her safhasına girmektedir. Bu kadar yaygın olan cihazları kullanmak için yeni yöntemler de araştırılmaktadır. Son teknolojiler arasında yer alan ses arayüzü ile etkileşim, kullanıcıların yoğun ilgi gösterdiği yöntemler arasındadır.

Mobil cihazların ses komutları ile yönetilebilmesi için, klasik grafiksel kullanıcı arayüzü (GKA) uygulamalarının, ses girdilerini alıp işlem yapabilecek şekilde yeniden geliştirilmesi gerekmektedir. Klasik GKA uygulamalarındaki gibi görsel bir arayüze sahip olmayacak olan bu uygulamaların geliştirilmesi ses kullanıcı arayüzü (SKA) adı verilen farklı bir metodoloji ile geliştirilmelidir.

SKA, kullanıcının sesini girdi olarak alan ve sonucunda da önceden kaydedilmiş sesleri veya sentezlenmiş(robotik) sesleri çıktı olarak verebilen bir kullanıcı arayüzü tipidir. SKA, her birinin kesin bir göreve denk geldiği ses komutlarını içerir. Bu komutlar SKA'nın bir parçası olan ses algılama kısmında gerçekleştirilir. İyi tasarlanmış ve iyi gerçekleşmiş bir SKA, ses algılama işlemindeki hata olasılıklarını azaltır, kullanıcıya hatırlaması ve kullanması kolay uygulama akışı sağlar. Kötü tasarlanmış bir SKA ise, anlaşılamayan, hatırlaması zor, ses algılama motoru tarafından algılanması zor ve bu sebeplerden hata yapmaya müsait komutlar içerir.

SKA'nın ana hedefi, kullanıcıyı sıkmayacak ve basit kullanıma sahip bir uygulama gerçekleştirmektir. Bu hedef doğrultusunda yapılan tez çalışmasında, SKA uygulaması geliştirirken hangi yöntemlerin kullanılabileceği araştırılmış ve kullanılabilirlik açısından değerlendirilerek kullanılabilirliği arttıran daha iyi yöntemler sunulmuştur.

Anahtar Kelimeler: Mobil Cihazlar, Ses Kullanıcı Arayüzü, Android, Sesten – Metin Çevirisi.

SUMMARY

Nowadays, mobile devices are very popular and many people use them. As they are very popular, new usage techniques are searched. One of and most interesting usage technique is voice user interface.

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. For example, sending e-mails while driving or changing the current playing song on mobile device just by voice commands.

VUI is a user interface kind which can receive voice as input and can output pre-recorded or synthesized (robotic) voices. VUI includes voice commands to correspond to a definite task. These commands are carried out at the voice recognition which is a part of VUI. A well-designed and well-implemented VUI, reduces the possibility of error at voice recognition stage, allows the user to remember and easy to use application flow. A poorly designed VUI, includes unclear, difficult to remember and hard to be detected commands by voice recognition engine that causes the error.

The main objective of VUI creating a simple, easy to use application and not boring the user. In this study, the methods to be used in developing the VUI application which increases the usability, researched and evaluated in terms of usability.

Key Words : Mobile Devices, Voice User Interface, Android, Speech – to – Text.

TEŐEKKÜR

Çalıőmalarım boyunca deęerli yardım ve katkılarıyla beni yönlendiren Hocam Doç. Dr. Mehmet Göktürk'e,

Tez çalıőmalarım sırasında benden hiç bir yardımı esirgemeyen, bana daima destek olan çalıőma arkadaşlarıma,

Bugünlere gelmemde büyük emeęi olan, hayatım boyunca maddi ve manevi desteklerini her zaman yanımda hissettięim aileme en içten dileklerle teşekkür ederim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	v
SUMMARY	Hata! Yer işareti tanımlanmamış.
TEŞEKKÜR	vii
İÇİNDEKİLER	viii
SİMGELER ve KISALTMALAR DİZİNİ	xi
ŞEKİLLER DİZİNİ	xii
TABLOLAR DİZİNİ	xiii
1. GİRİŞ	1
2. SES KULLANICI ARAYÜZÜ	3
2.1. Ses Kullanıcı Arayüzü Nedir?	3
2.2. Ses Kullanıcı Arayüzü Üzerine Yapılan Çalışmalar	4
2.2.1. Menü Yapısı Üzerine Yapılan Çalışmalar	4
2.2.2. Dilbilgisi Üzerine Yapılan Çalışma	5
2.2.3. Görev Bazlı Yapılan Çalışma	6
3. KULLANILABİLİRLİK	7
3.1. Kullanılabilirlik Değerlendirme Yöntemleri	7
3.1.1. Sezgisel Değerlendirme	8
3.1.2. Üzerinden Geçme	10
3.1.3. Kullanıcı Testleri	11
3.2. Tez Çalışmasında Kullanılabilirlik Yöntemlerinden Yararlanması	12
4. SES ARAYÜZÜ UYGULAMA TASARIM YÖNTEMLERİ	13
4.1. Ses Aksiyonları İçin Uygun Görevlerin Seçimi	13
4.2. Kullanıcı ve Uygulamanın Çıktılarına Karar Verilmesi	15
4.3. Başarıyı Arttırmak İçin Ses Girdilerinin Sınırlanması	15
4.4. Kullanıcılar İçin Eğitim	16

4.5. Başarının Teyit Edilmesi ve Hatalardan Kurtulmak İçin Kullanıcıya Yardım Edilmesi	16
4.6. Ses Algılamının Yanlışıklıkla Aktif Hale Getirilmesi	18
4.7. Kullanıcıya Net Konuşmanın Öğretilmesi	19
5. PROTOTİP SKA UYGULAMASI	20
5.1. Uygulamanın Amacı	20
5.2. Uygulama Mimarisi	20
5.2.1. Android İşletim Sistemi ve Ses Arayüzleri	21
5.2.1.1. Ses Algılayıcı (Speech Recognizer)	21
5.2.1.2. Metinden Ses (Text-to-Speech)	22
5.2.2. PHP ve JSON	23
6. HİPOTEZ	25
6.1. Dilbilgisi (Grammar) Yapısı	25
6.2. Komutların Cümle İçinde Aranması	26
6.3. Kelimenin Kökünün Bulunması	27
7. YÖNTEM	28
7.1. Görevlerin Seçilmesi	28
7.2. Görev Yöntemlerinin Seçilmesi	28
7.2.1. Menü Uygulaması	28
7.2.2. Sadece Komut Kelimesinin Beklenmesi	29
7.2.3. Komut Kelimesinin Köküne İnilmeden Beklenmesi	30
7.3. Karşılaştırılacak Görev Yöntemlerinin Seçilmesi	30
7.3.1. Dilbilgisi Yapısı Uygulaması	31
7.3.2. Kelime Algılama Algoritması	31
7.3.3. Kelimenin Kökünün İncelenmesi	32
8. SKA'LARDA KULLANILABİLİRLİK DEĞERLENDİRMESİ	33
8.1. Sezgisel Değerlendirme	33
8.2. Kullanıcı Deneyleri	33
8.2.1. Sesli Düşünme	34
8.2.2. Görev Gerçekleme	34

9. DENEY DEĞERLENDİRMESİ	36
9.1. Örneklem	36
9.2. Deney Ortamı	36
9.3. Deney Aşamaları	37
9.4. Ölçüm	37
10. SONUÇLAR	39
10.1. Dilbilgisi ve Menü Yapısı	39
10.2. Kelime Algılama Yöntemi	41
10.3. Kelimenin Köküne Göre İşlem Yapılması	42
10.4. Uygulama Aktifleştiriciler	43
11. KAYNAKLAR	45
12. ÖZGEÇMİŞ	47

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler ve

Kısaltmalar Açıklamalar

GKA : Grafiksel Kullanıcı Arayüzü

MS : Metinden Ses

SKA : Ses Kullanıcı Arayüzü

ŞEKİLLER DİZİNİ

<u>Sekil No:</u>	<u>Sayfa</u>
5.1: Uygulama Mimarisi.	21
5.2: Ses Tanımlayıcı.	22
5.3: Metinden Ses Çıktısı.	23
6.1: Dilbilgisi Yapısı.	25
6.2: Kelime Algılama Algoritması.	27
7.1: Menü Uygulama Mimarisi.	29
10.1: Görevlerin tamamlanma zamanları.	40
10.2: Görevlerde hataya düşme sayısı.	41
10.3: Kelime algılama yöntemi.	42
10.4: Kelimenin köküne göre işlem.	43

TABLÖLAR DİZİNİ

<u>Tablo No:</u>	<u>Sayfa</u>
5.1: Kimlik Tablosu.	23
5.2: Plaka Tablosu.	24
9.1: Donanım Bilgisi.	37

1. GİRİŞ

İnsanlar birçok sebepten mobil cihazlarını ses komutları ile yönetebilmeye yoğun ilgi göstermişlerdir. Bu ilginin sebebi olarak, küçük klavyede yazı yazmaktan hoşlanmamaları, mobil cihazları ile sesli etkileşimin klavye etkileşiminden daha kolay olması gösterilebilir. Bunların yanında bir diğer sebep de gündelik yaşam işlerini devam ettirirken, elleri ve gözleri cihazlarına bağımlı olmadan telefonlarını kullanmak istemeleridir. Kişisel kullanımın yanında SKA uygulamaları firmalar tarafından da tercih edilmektedir. Firmaların tercih sebepleri arasında gider azaltma(çalışan sayısı azaltıldığı için), daha etkili müşteri ilişkileri gibi konular sayılabilir [Cohen, 2004].

Geçmişte insanların ses komutları ile mobil cihazlarını yönetebilmeleri mümkün değildi. Çünkü ses komutları ile işlem yapabilmek için yüksek donanım ihtiyacı vardı. Son zamanlardaki mobil cihazların donanım özelliklerinin artması ve sesten-metin çevirisi çalışmalarının başarısının artması, mobil cihazların ses komutları ile yönetilebilmesi açısından önemli adımlar olmuşlardır.

Mobil cihazların ses komutları ile kontrol edilebildiği uygulamalar popüler olmaya başlayınca, akla ilk gelen sorulardan biri de ses kullanıcı arayüzü (SKA) ile nasıl kullanıcı dostu uygulama geliştirilebileceği sorusudur.

Bu çalışmada, mobil cihazlar üzerinde kullanıcı dostu bir SKA uygulamasının nasıl geliştirilebileceği üzerine araştırma yapılmıştır. Çalışma öncelikle SKA alanında yapılan çalışmaların incelenmesi ile başlamış ve kullanılan yöntemler belirlenmiştir. Bir sonraki aşamada mobil cihazlar üzerinde SKA uygulaması geliştirmek için uygun olan araçlar araştırmalar sonucunda belirlenmiştir. Son olarak, güvenlik görevlilerinin T.C. kimlik numarası veya plaka sorgulaması yapabilecekleri bir prototip SKA uygulaması geliştirilmiştir. Denekler yardımı ile SKA uygulaması üzerinde SKA uygulama geliştirme yöntemleri test edilerek mobil cihazlar üzerinde çalışacak bir SKA uygulaması geliştirilmesi için en iyi geliştirme yöntemleri belirlenmiştir.

Temel kullanılabilirlik prensiplerinin SKA'lara uygulanması, kullanıcıların bilgisayarla etkileşimini daha etkili, verimli ve tatmin edici hale getirecektir. Bu amaç doğrultusunda, çalışma kapsamında kullanıcıların ihtiyaçları incelenmiştir. Literatür taraması işlemiyle çeşitli kaynaklardan SKA'ların kullanılabilirliğine etki

eden faktörler toplanmıştır. Ayrıca literatürde kullanılabilirlik değerlendirilmelerinde kullanılmış olan yöntemler SKA'lara uyarlanarak bir değerlendirme yöntemi oluşturulmuştur. Hem toplanan faktörlerin bir alt kümesini geçirmek, hem de ortaya konan SKA kullanılabilirliğini arttıran yöntemlerin değerlendirmesi amacıyla, her bir görev için deneyler gerçekleştirilmiştir.

Kullanıcılar ile gerçekleştirilen deneyler sonucunda, SKA'ların kullanılabilirliğini arttıran yöntemler tespit edilmiştir. Temel olarak menü ve dilbilgisi yapılarının incelendiği deneyler sonucundan, kullanıcıların dilbilgisi yöntemini daha kullanılabilir buldukları tespit edilmiştir. Bunun yanında "Kelime Algılama Algoritması" ve kelimenin köküne göre işlem yapılması, dilbilgisi ve menü yönteminin kullanılabilirliğini arttıran ek yöntemler olarak saptanmıştır.

Tez çalışması toplam on bölümden oluşmaktadır.

Giriş bölümü olan birinci bölümün ardından ikinci bölümde, SKA'nın ne olduğu konusunda bilgi verilmiş ve devamında SKA alanında yapılan ve çalışmamızda da yararlandığımız SKA çalışmalarından bazıları anlatılmıştır.

Çalışmanın amacı kullanılabilirlik açısından en iyi SKA geliştirme yöntemlerini bulmak olduğundan, üçüncü bölümde kullanılabilirlik ve kullanılabilirlik değerlendirme yöntemleri anlatılmıştır. Bu bölümde anlatılan yöntemler, geliştirilen prototip uygulamanın kullanılabilirliğinin değerlendirilmesinde kullanılmıştır.

Dördüncü bölümde yapılan literatür taraması sonucu ortaya çıkarılan SKA geliştirme yöntemleri anlatılmıştır. Çalışma kapsamında gerçekleştirilen prototip SKA uygulaması da ilk etapta bu bölümde anlatılan yöntemler takip edilerek geliştirilmiştir.

Beşinci bölümde çalışma kapsamında Android işletim sistemi üzerinde gerçekleştirilen prototip SKA uygulamasının mimarisi anlatılmıştır.

Altıncı bölüm içerisinde, çalışma kapsamında SKA uygulamalarının kullanılabilirliğini olumlu etkilediği belirtilen hipotez yöntemleri anlatılmıştır.

Tezin yedi, sekiz ve dokuzuncu bölümlerinde, kullanıcılar ile hipotez yöntemleri üzerine yapılan deneylerin gerçekleştirilme yöntemi anlatılmış ve son olarak onuncu bölümde deney sonuçları ve deney sonuçlarının değerlendirilmesi anlatılmıştır.

2. SES KULLANICI ARAYÜZÜ

Bu bölümde ses kullanıcı arayüzü (SKA) konusunda bilgi verilecektir. Öncelikle SKA'nın tanımı yapılacak ve bunun devamında SKA konusunda yapılmış olan çalışmalardan bazıları anlatılacaktır.

2.1. Ses Kullanıcı Arayüzü Nedir?

Ses kullanıcı arayüzü, insanın bilgisayarla etkileşimini ses arayüzü ile mümkün hale getirerek, bir işlemin veya servisin, ses komutları aracılığı ile aktif edilebilmesini sağlamaktadır [Milette, 2012].

Kısa bir süre önceye kadar, bilgisayar ile konuşabilmek sadece bilim kurgu olarak nitelendirilmekteydi. Fakat teknolojideki son zamanlarda olan gelişmeler, bilgisayarlarla konuşabilmenin artık sadece bilim kurgu olmadığını göstermiştir. Bu gelişmeler sayesinde insanlar, ses arayüzü sayesinde eller ve gözler bağımsız olarak işlemlerini yapabilmenin avantajından yararlanabilmektedirler [Thompson et al., 1998].

SKA'nın çok ilgi gören avantajları olmasına rağmen, dezavantajları da yok değildir. İnsanların söylediklerinin bilgisayar tarafından algılanmamasına karşı oldukça az sabırları vardır [Milette, 2012]. Bu sebepten, SKA uygulamaları hatalar için lükse sahip değildirler. SKA uygulamalarının kullanıcılarından kabul görebilmeleri için girdilere hızlı ve güvenli bir şekilde cevap vermeleri gerekmektedir. Bunlara paralel olarak, iyi bir SKA uygulaması tasarlamak, bilgisayar bilimlerinde, dilbiliminde ve insan psikolojisi alanında bilgi sahibi olunmasını gerektirmektedir.

Hedef kullanıcının karakteristik özellikleri çok önemlidir. Örnek olarak, genel kullanıcılar için geliştirilmiş olan bir SKA uygulaması, özellikle ilk kullananlar için normalden fazla sayıda yardım veya rehberlik içermelidir. Buna karşın belirli bir hedef kitle için geliştirilmiş bir SKA uygulamasının ilk amacı kullanılabilirliğin iyi olmasıdır ve buna bağlı olarak yardım ve rehberlik daha az önemsiz kalabilmektedir. Kısaca, SKA uygulamaları genelleştirilebilecek olan uygulama tipleri değildirler. Her iş modeli ve hatta her iş modelinin farklı her görevi için, farklı tasarımlar yapılması gerekmektedir [Cohen, 2004].

2.2. Ses Kullanıcı Arayüzü Üzerine Yapılan Çalışmalar

Çalışma kapsamında ilk olarak, ses kullanıcı arayüzü üzerine yapılan çalışmalar incelenmiştir. Bu bölümde, yapılan tez çalışmasında içeriğinden en çok yararlanılan çalışmalar anlatılacaktır.

İncelenen çalışmalar, hem SKA uygulamalarının gerçekleştirme yöntemleri bakımından hem de yapılabilecek çalışmalar bakımından yol göstermişlerdir.

2.2.1. Menü Yapısı Üzerine Yapılan Çalışmalar

SKA uygulamalarında kullanıcıdan ses girdilerini almanın farklı yöntemleri bulunmaktadır. Bu yöntemlerden biri olan menü yöntemi, kullanıcıdan girdilerin hiyerarşik bir yapıda sırayla alınması şeklinde tanımlanabilir. Kullanıcı, uygulamanın bir görevi tamamlayana kadar ihtiyaç duyacağı girdileri uygulamaya genelden-özele doğru olacak şekilde vermek durumundadır. Uygulamaların kullanıcıdan gerekli girdileri alma yöntemleri değişkenlik gösterebilir.

Bu alanda, Nicholas Ikhu Omoregbe tarafından yapılmış çalışma, sağlık sektöründeki bir problemi çözmek için ses kullanıcı arayüzlü ve menü yapısında geliştirilmiş bir çözüm sunmuştur [Omoregbe, 2010]. Çalışma, hastanelerde doktorların yazdıkları reçetelerin daha okunaklı olması için ses arayüzlü bir sistem sunmaktadır. Doktorların kötü el yazıları ile yazdıkları reçetelere göre verilen ilaçlarda veya ilaçların dozlarında yanlışlıklar olmaktadır. Çalışma, ses kullanıcı arayüzü kullanılarak, doktorların el ile yazdıkları reçeteleri ses girdileri ile okunaklı yazmalarına olanak sağlamaktadır.

Bu çalışma kapsamında geliştirilen uygulama telefon hattı üzerine kurulmuştur. Doktorlar reçete yazmak için yerel olarak işlevsellik sağlayan bir telefon numarasını çevirip işlem yapabilmektedirler. Sistem menü yapısı ile geliştirilmiştir ve uygulamada işlem yapabilmek için hiyerarşik menü yapısı ile kullanıcı yönlendirilmektedir. Örnek olarak, kullanıcı ilaç yazmak amacı ile sisteme başvurduğu zaman aşağıda belirtilen diyalog ortaya çıkmaktadır.

- Sistem : *Yazmak istediğiniz ilacın adını söyleyin.*
- Kullanıcı : *Örnek ilaç.*
- Sistem : *İlacın dozajını söyleyin.*

- Kullanıcı : *Günde 2 defa tok.*

Bir grup Google çalışanı tarafından yapılan diğer bir menü yapısına dayanan çalışma, Google arama motorunu ses arayüzü ile kullanılabilir hale getirmiştir [Neustein, 2010]. Kullanıcılar aramak istediklerini ses ile girdi olarak vererek arama sonuçlarını elde edebilmektedirler.

Yapılan çalışmada farklı bir menü yapısı kullanılmıştır. Uygulama, kullanıcıya sorular sorarak birden fazla seçenek olabilecek seviyede, ihtiyaç duyduğu girdileri olarak işlem yapmaktadır. Bu şekilde kullanıcının uygulamayı kullanmak için aklında tutması gereken bir akış veya bir komut dizisi gerekli olmamaktadır. Örnek bir sorgu işlemi aşağıdaki gibi olmaktadır.

- Uygulama : *Hangi il ve ilçe?*
- Kullanıcı : *İstanbul, Kadıköy*
- Uygulama : *Ne aramak istiyorsunuz?*
- Kullanıcı : *Pizzacı....*

2.2.2. Dilbilgisi Üzerine Yapılan Çalışma

SKA uygulamalarında kullanıcıdan ses girdilerini almanın diğer bir yöntemi de, belirlenen bir dilbilgisi yapısı üzerinden gerekli olan girdilerin alınmasıdır. Bu yöntemde kullanıcı uygulamanın ihtiyacı olan girdileri tek bir cümle ile ve belirlenmiş kalıpta vererek uygulamanın görev için gerekli olan girdileri almasını sağlar. Bu alanda, Microsoft araştırma laboratuvarlarında, Li DENG ve Dong YU tarafından gerçekleştirilen çalışmada ses merkezli ve dilbilgisi yapısı üzerine dayalı bir harita sorgulama uygulaması geliştirilmiştir [DENG ve YU, 2005].

Çalışma kapsamında kullanıcıların harita üzerinde sorgulamalar yapabilecekleri bir uygulama gerçekleştirmişlerdir. Bu uygulama ile kullanıcılar harita üzerinden yol tarifleri alabilmekte veya bilmedikleri bir yeri harita üzerinde sorgulayıp bulabilmektedirler. Farklı sorgulama yönteminden bahsedilen çalışmada kullanıcıların yapabilecekleri sorgulama yöntemleri dilbilgisi yapısına dayanmaktadır.

Geliştirilen sorgulama yöntemlerine yön tarif sorgulaması örnek olarak gösterilebilir. Bu yöntemle kullanıcılar bir yerden başka yere yol tarifi alabilmek için

uygulamaya belirlenmiş formatta bir girdi vermelidirler. Bu girdiye örnek olarak, “<başlangıç yeri’nden><bitiş yeri’ne> nasıl gidilir” sorusu gösterilebilir. Bu şekilde verilecek bir girdi sonrası uygulama “başlangıç ve bitiş” yerlerini girdiden çekerek, kullanıcıya harita üzerinde bir yol tarifi çizmektedir.

2.2.3. Görev Bazlı Yapılan Çalışma

Görev tabanlı çalışma, belirlenmiş olan bir görevin farklı yöntemlerle yerine getirilmesi ve bu şekilde yöntemlerin karşılaştırılması olarak tanımlanabilir. Bu alanda Edim Azom tarafından yapılan çalışmanın amacı, gelişmemiş toplumlardaki kişilerin tercihlerini öğrenmektir [Azom, 2010]. Bunun için belirli görevler seçilmiş ve bu görevler mobil uygulama olarak gerçekleştirilmiştir. Gerçeklenen iki tür uygulama vardır, bunlardan birincisi klasik grafik kullanıcı arayüzü ile geliştirilmiş uygulama ikincisi ise ses kullanıcı arayüzü ile geliştirilmiş bir uygulamadır. Çalışma, geliştirilen uygulamalar üzerinden belirlenen görevleri denekler üzerinde test edilmesi şeklinde devam etmektedir. Kullanıcılardan iki uygulama üzerinden de aynı görevleri yapmaları isteniyor ve sonuç olarak kullanıcıların hangi uygulama üzerinde daha başarılı oldukları gözlemleniyor.

Çalışma sonucunda, kullanıcıların grafik kullanıcı arayüzü uygulamasında görevleri daha başarılı bir şekilde yerine getirdikleri gözlemlenmiştir. Sonuç olarak da, ses kullanıcı arayüzü uygulamalarının, grafik arayüzü uygulamalarının bir sonraki basamağı olarak nitelendirilmiştir.

3. KULLANILABİLİRLİK

Kullanılabilirlik, kullanıcıların bir sistemle etkileşimlerinin kolaylığını ifade eder. ISO 9241’de kullanılabilirlik, “Hedef kullanıcıların gerekli görevleri etkili, verimli ve tatmin edici bir şekilde yerine getirebilmeleri” olarak tanımlanmaktadır [ISO 9241, 1998].

Kullanılabilirlik tanımında da belirtilen insan-bilgisayar etkileşimi hedefleri şunlardır:

- Etkili Olma (“Effectiveness”)

Kullanıcının yerine getirmesi gereken görevleri eksiksiz ve doğru bir şekilde tamamlayabilmesi için gerekli işlevselliğin sağlanmasıdır.

- Verimlilik (‘Efficiency’)

Kullanıcının eksiksiz ve doğru bir şekilde yerine getirebildiği görevlerin, harcadığı fiziksel ve zihinsel kaynaklara (zaman veya fiziksel uğraş gibi) oranıdır.

- Kullanıcı Tatmini (‘User Satisfaction’)

Kullanımın rahat ve kabul edilebilir oluşudur. Kullanıcının bilgisayarla etkileşimi süresince yaşadığı sinir bozukluğu ile ters orantılıdır.

3.1. Kullanılabilirlik Değerlendirme Yöntemleri

Uzmanlar ve kullanıcılar tarafından yapılan kullanılabilirlik değerlendirmeleri, kullanıcı odaklı arayüz tasarımının önemli bir bileşenidir. Sistemlerin kullanılabilirlik gereksinimlerini karşılayıp karşılamadıklarını ortaya çıkarmak için değerlendirmeler yapılmalıdır. Farklı avantajlara ve maliyetlere sahip başlıca yöntemler aşağıda anlatılmıştır. Bir sistemi değerlendirmek için birden fazla yöntem bir arada kullanılabilir. Değerlendirmeler sonunda ortaya çıkan problemler giderilerek daha kullanılabilir bir sistem elde edilebilir.

3.1.1. Sezgisel Değerlendirme

Sezgisel değerlendirme (“heuristic evaluation”), kullanıcı arayüzünün uzman bir değerlendirici tarafından, belirli bir kriter kümesine göre değerlendirilerek o kriterlere uyum ölçüldüğü bir yöntemdir. Değerlendirme uzman tarafından yapıldığı için “Uzman Değerlendirmesi” olarak da adlandırılır. Sezgisel değerlendirme, ciddi kullanılabilirlik hatalarının tasarım aşamasında yakalanarak düzeltilmesine olanak tanır. Kullanıcı davranışlarının yorumlanmasını gerektirmediği ve belirli bir kriter kümesine dayandığı için kolay ve hızlıdır. Tasarım aşamasında kullanılabilirlik hatalarını teşhis edebildiği için maliyet açısından avantajlıdır. Diğer taraftan değerlendiricilerin tarafı veya yetersiz olmaları durumunda hata oluşabileceğinden kullanıcı deneyleriyle birlikte kullanılmalıdır. Bu yöntem, kullanıcı grafik arayüzleri için 10 kriter ile birlikte, kullanılabilirlik uzmanı Jacob Nielsen tarafından ortaya konmuştur [Nielsen ve Molich, 1990]. Grafik arayüzleri için konan bu kriterler, SKA uygulamaları için de geçerlidir ve tezin devamında bu kriterler açıklanmaktadır:

- Sistem Durumu ile İlgili Geri Besleme (“Feedback”)

Sistem kullanıcıları mevcut durum ile ilgili sürekli bilgilendirmelidir. Kullanıcı, verdiği girdinin nasıl yorumlandığı ve o anda ne yapıldığı hakkında daima bilgi sahibi olmalıdır. En yaygın kullanılan örnek, sistem kullanıcıyı bekleteceği zaman ses çıktısı olarak kullanıcıya işlemin devam ettiğine dair çıktı vermesidir. Eğer işlem birden fazla adımdan oluşuyorsa, her adım için kullanıcıya ses çıktısı verilip işlemin ilerlediği kullanıcıya bildirilir.

- Kullanıcının Dilini Konuşma (“Speak the Users’ Language”)

SKA uygulamaları insan-bilgisayar etkileşiminin ses yolu ile oluşturulduğu bir yöntemdir. Bu nedenle, kullanılan terimler, kelimeler, yada kavramlar, kullanıcıya tanıdık olanlardan seçilmeli, sisteme yönelik tanımlamalardan kaçınılmalıdır. Kullanılan kelimeler ya da komutlar, gerçek hayat ile analogi kurabilmelidir.

- Kullanıcı Kontrolü (“Clearly Marked Exits”)

Kullanıcılar bir sistemin bazı kısımlarına yanlışlıkla girerler, ya da bazı değişiklikler yanlışlıkla yapılabilir. Bu yanlışlıklar kullanıcıların kendilerini tuzağa

düşmüş hissetmesine neden olur. Sistemler kullanıcıya, mümkün olan durumlar için, kolaylıkla bulabileceği çıkışlar sunmalıdır. Örneğin kullanıcıdan karar beklenen durumlarda “Onay” komutu sağlamak, uzun süren işlemlerin kesilebilmesine imkân tanımak, istenen her hangi bir anda “İptal” ya da yanlış yapılan işlemlerin üstesinden gelmek için “Düzel” opsiyonu sunmak tavsiye edilen stratejiler arasında yer alır.

- Tutarlılık (“Consistency”)

Bir üründe, gerek ürünün muadilleri ile, gerekse ürünün kendi içinde tutarlılık sağlanmalıdır. Farklı isimler ya da durumlar daima farklı anlamlara gelmelidir, benzer şeyler farklı biçimlerde tarif edilmemelidir. Kullanıcılar farklı isimlerin veya durumların aynı şey anlamına gelip gelmediğini düşünmek zorunda bırakılmamalıdır.

- Hata Önleme (“Error Prevention”)

Hata mesajlarının anlaşılır ve yönlendirici olması gerekliliği ile birlikte, hatalar henüz oluşmadan engellenmesi daha faydalıdır. SKA uygulamalarında bu durum için kullanıcının ses girdilerinin sınırlamayı örnek olarak verebiliriz. Girdilerin sınırlandırılması, sesten-metin çevirisinin başarısını arttırdığı için, hataya düşme olasılığı arttırılmış ve daha hataya düşmeden önlenmiş olunur.

- Hatırlama Yerine Tanıma (“Recognition rather than Recall”)

Kullanıcıların kısa süreli hafızaları sınırlı olduğu için, kısa süreli hafıza üzerindeki yükü azaltmak gerekmektedir. Bu amaçla, kullanıcıların girdileri ya da komutları hatırlamaları yerine, sınırlı bir kümeden seçmeleri ve komutların günlük hayattan seçilmesi kullanılabilirliği arttırır.

- Hızlandırıcıların Kullanımı (“Accelerators”)

Bir uygulama bir süre kullanıldıktan sonra, kullanıcılar tecrübe kazanırlar. Tecrübeli kullanıcıların sık yaptıkları işleri kolaylaştıracak kısa yol tuşları, komut tamamlayıcılar, en son kullanılan dosyaların listesi gibi hızlandırıcılar, ürünün uzun vadeli kullanımı için vazgeçilmezdir. Bu kriterin SKA uygulamalarındaki karşılığı, en son yapılan işlemin tekrar başlatılmadan aynı işlemleri yapmadan işlenebilmesi

söylenbilir. Örnek olarak, kullanıcı kimlik sorgulaması işlemini bitirdikten sonra, tekrar aynı işlemi başa dönmeden yapabilmelidir.

- Estetik ve Minimalist Tasarım (“Minimalist Design”)

Kullanıcıya sunulan ses arayüzleri, ilgisiz ya da nadiren kullanılacak bilgi içermemelidir. Çünkü sunulan ilgisiz bilgiler, arayüzdeki gerekli ve çok kullanılacak bilgileri karmaşıklığa sokmakta ve fark edilmelerini güçleştirmektedir. Bu nedenle, estetiğin yanı sıra minimalist tasarıma da dikkat edilmelidir. Ve ayrıca ses arayüzleri geçici hafızaya hitap ettiği için, arayüzlerin basit tutulması kullanıcının hangi komutları kullanabileceğini hatırlayabilmesi açısından da önemlidir.

- Açık Hata Mesajları (“Good Error Messages”)

Uygulamalar kullanıcıları mümkün olduğunca hatadan uzak tutmalıdır, fakat yine de hataların oluşması kaçınılmazdır. Kullanıcı hatalı girdiler sağladığında, kimi sistemler hatalı olan isteği yerine getirmez, fakat kullanıcıya da herhangi bir hata mesajı sunmaz. Yanlış kayıtların oluşması engellenmiş olsa da, bu defa kullanıcı ne olup bittiğini anlamayacağı için kontrolü yitirdiğini hissedecektir. Kimi sistemler ise, kullanıcıyı uyardıktan hatalı girdiyi düzeltirler. Fakat bu durum da yanlış düzeltmeler yapılabileceğinden risklidir. Doğru olan, kullanıcının bir hata oluştuğuna dair açıkça ve kullanıcının dilini kullanan bir mesaj verilmesi ve varsa olası düzeltmelerin kullanıcıya birer seçenek olarak sunulmasıdır. Hata mesajları açıklayıcı olmalı ve gerçeklemeye dair detay içermemelidir. Yapıcı öneriler sunmalı ve kullanıcıyı suçlayıcı ifadelerden kaçınılmalıdır.

- Yardım ve Dokümantasyon (“Help and Documentation”)

Sistemin dokümantasyonsuz kullanılabilir şekilde tasarlanması gerekliliği ile birlikte, yardım ve destekleyici dokümantasyon yine de gerekli olabilir. Sunulan yardım ve dokümantasyon kaynağı kolayca erişilebilir olmalıdır. Üzerinde uğraşılacak görevle ilgili arama yapılabilmesi, sade ve basit olması kullanımı kolaylaştıracaktır.

3.1.2. Üzerinden Geçme

Üzerinden geçme, bir arayüzün gerçek kullanıcılar olmaksızın değerlendirilmesidir. Bu yöntemde geliştiriciler, analistler ve İBE uzmanları,

kendilerini tecrübesiz kullanıcının yerine koyarak, belirlenmiş görevlerin üzerinden geçerler. Kullanıcıların çoğu yeni bir sistemi öğrenmek için eğitim almak veya kullanım kılavuzları okumak yerine, deneme yanılma yöntemini tercih etmektedir [Carroll, 1987]. Üzerinden geçme ile değerlendirilmek olan bir sistemde, bir görevi yerine getirmek için geçilen aşamaların hepsi için başarılı ve başarısız senaryolar için, kullanıcının başarılı senaryoları seçip seçmeyeceği ve nedenleri raporlanır. Bu yöntem ilk olarak Lewis tarafından önerilmiştir [Lewis, 1990].

Üzerinden geçme yöntemi, görevlere özel kullanılabilirlik problemlerini bulmada çok etkilidir. Kullanıcıların hedeflerini ve varsayımları anlamayı kolaylaştırır. Ayrıca geliştirimin erken evrelerinde de kullanılabilir. Diğer taraftan, zaman alıcı olması ve sadece öğrenilebilirlik problemlerini ortaya çıkardığı için kullanım alanı dardır.

3.1.3. Kullanıcı Testleri

Sezgisel değerlendirme geliştirimin erken aşamalarında kullanılabilir, fakat kullanıcı testleriyle desteklenmelidir, çünkü gerçek kullanıcıların ne beklediklerini tahmin etmek her zaman mümkün olmamaktadır. Mevcut kullanılabilirlik seviyesine karar verebilecek olanlar son kullanıcılar veya kullanıcı adaylarıdır. Ürün son kullanıcıların hizmetine girmeden önce, kritik görevler kümesi bir grup aday kullanıcı tarafından laboratuarda ya da kullanıcının çalışma ortamında test edilir [Preece, 1994].

Bu yöntemde ilk yapılması gereken test planının hazırlanmasıdır [Shneiderman, 1997]. Kullanılabilirlik mühendisleri, tasarımcılarla bir araya gelerek testlerin başlamasından 2-6 hafta önce test planını hazırlamaya başlarlar. Planda, testlerde gerçekleştirilecek görevler, başarı kriterleri ve toplanacak metrikler seçilir. Her deney için verilecek süre ve toplam katılımcı sayısı da planda belirtilir. Ayrıca katılımcıların hangi kaynaklardan bulunacağı da planlanır. Planın hazırlanmasının ardından, deneye en geç bir hafta kala, 1-3 pilot deney gerçekleştirilir, böylece deneyde var olan eksiklikler gerçek deneyler başlamadan düzeltilir.

Deneyler tasarlandıktan sonra, gerçek kullanıcıları temsil eden kullanıcılar seçilmelidir. Deneyin konusuna göre değişmekle birlikte, kullanıcının ilgili alandaki tecrübesi, eğitimi, dil becerisi seçimi etkileyen unsurlar arasındadır. Seçilen kullanıcılar, kendilerinin becerilerinin değil programın test edildiği konusunda ikna

edilmelidir. Katılım tamamen gönüllü olmalıdır ve kullanıcı istediği zaman deneyi bırakabilmelidir. Deney başlamadan önce, katılımın karşılığında simgesel de olsa bir hediye verilmelidir.

Deneyler sırasında hata oranı, gerçekleştirme süresi, yardım dokümanlarına başvurma sayısı gibi metrikler planda belirtildiği yöntemlerle toplanır. Deneyler sırasında, bir gözetmen kullanıcıyla ilgilenir ve bu sırada gözlemsel bazı verileri toplar. Ayrıca deneyler video kamera ile ya da masaüstü kaydedici programlar ile kaydedilir. Ayrıca, kullanıcıdan deney boyunca aklından geçenleri sesli olarak ifade etmesi istenir.

Deneyler tamamlandıktan sonra ortaya çıkan kullanılabilirlik problemleri, kullanılabilirlik mühendisleri tarafından önceliklendirilerek raporlanır. Hatalar giderildikten sonra, evrimsel geliştirme metodolojisine göre gerek görülürse deneyler tekrarlanır ve ürün olgunlaşmaya kadar bu süreç devam eder. Fakat kullanılabilirlik problemlerinin bir kısmı fark edilmeden kalabilir. Bu durumda, gerçek kullanım sırasında elde edilen geri beslemeler değerlendirilerek gerekli iyileştirmeler yapılır.

3.2. Tez Çalışmasında Kullanılabilirlik Yöntemlerinden Yararlanılması

Bölüm 3'de kullanılabilirlik değerlendirme yöntemleri anlatılmıştır. Tez çalışmasında kullanıcı dostu bir SKA uygulaması geliştirmek için bu bölümde anlatılan yöntemler takip edilmiştir. Uygulama geliştirilirken ilk aşamada sezgisel değerlendirme yöntemi dikkate alınarak, bu bölümde belirtilen kıstaslara göre uygulamanın tasarımı yapılmıştır. Sonraki adımda üzerinden geçme yöntemi ile tez çalışmasını yapanlar geliştirilen prototip SKA uygulamasını, kendilerini tecrübesiz kullanıcılar yerine koyarak kullanmışlardır. En son aşamada da kullanıcılar ile deneyler yapılarak çalışmada belirtilen yöntemlerin kullanılabilirliği incelenmiştir.

4. SES ARAYÜZÜ UYGULAMA TASARIM YÖNTEMLERİ

Doğal ve iyi bir ses arayüzü tasarlamak, iyi planlamayı ve sürekli test etmeyi gerektirir [Cohen, 2004]. Bu bölümde, bir ses arayüzü tasarımı yapılırken ihtiyaç duyulabilecek olan tasarım tekniklerine yer verilmiştir. Çalışma için geliştirilen ses arayüzü uygulamasının tasarımı yapılırken kullanılabilirlik değerlendirme yöntemlerinden sezgisel değerlendirmenin ilkeleri dikkate alınarak tasarım yapılmıştır.

Klasik GKA uygulamaları, kullanıcıların çeşitli görevlerini, grafiksel arayüzlerin yardımcılığı ile dokunarak ve göstererek gerçekleştirmelerini sağlarlar. SKA'da ise benzer işlemler ses aracılığı ile yapılır. Bir SKA'da kullanıcı konuşur, dinler ve genel olarak ekrana bakmasına veya ekranda bir şeye dokunmasına gerek kalmadan işlemini bitirir. Bu bölümde, bir SKA geliştirirken uygulanabilecek yöntemlerden bahsedilmiştir.

4.1. Ses Aksiyonları İçin Uygun Görevlerin Seçimi

SKA tasarımı yapılırken öncelikle bir görevin ses arayüzü için uygun olup olmadığına incelenmesi gerekmektedir [Milette, 2012]. Çünkü her görev ses arayüzü ile gerçeklenmek için uygun değildir [Welie, 2001]. Temelde görevin ses arayüzü için uygun olup olmadığına karar vermek ses girdi ve çıktıların özelliklerine bağlıdır. Sesi girdi olarak kullanmak, kullanıcı için sıkıcı olabilecek fazladan işlemler gerektirmektedir. Bu işlemler, ses algılamanın aktif hale getirilmesi, konuşma ve cevabı bekleme olarak sıralanabilirler. Hatta bazı durumlarda ses algılama işlemi başarısız olabilir ve kullanıcının bu işlemi tekrarlaması, yani tekrar konuşması gerekebilir. Bahsedilen dezavantajların hepsine rağmen, ses arayüzünün o görev için kullanım avantajları, bütün dezavantajların göz ardı edilmesine sebep olabilir ise, bu durum o görevin ses arayüzü için uygun olduğunu göstermektedir.

Ses çıktısı kalıcı değildir ve kolayca unutulur. Eğer görsel bir öğe mevcut değilse kullanıcı, uygulama tarafından üretilen bütün ses çıktılarını hatırlamak zorunda kalacaktır. Bu durumda, kullanıcı uygulamanın bir dakika önce ürettiği ses çıktılarını unutabilir.

Girdi almanın başka yolu yoksa veya üretilen çıktının gösterilemediği bir durum varsa, kullanıcının eller ve gözleri meşgul iken, uygulama ile kullanıcının en güvenli etkileşim yolu ses girdi ve çıktıları olacaktır [Milette, 2012]. Bu şartlar, kullanıcı bir işle meşgul iken başka bir işi de yapmak istediği zaman oluşabilir. Örnek olarak, araba sürerken veya yemek yaparken mail göndermek istenmesi gösterilebilir.

Klasik GKA uygulamalarına göre kullanılabilirlik olarak kullanıcıya daha çok verimlilik sağlıyor ise o görev SKA için uygundur. Ses aksiyonları çok güçlüdür ve tek bir ses girdisi ile birçok işlem yapılabilmesine olanak sağlayabilmektedir [Cohen, 2004]. Eğer ses aksiyonu böyle bir yarar sağlamıyorsa, kullanıcı daha güvenilir ve daha hızlı olan klasik GKA'yı tercih edecektir. Örnek olarak, iyi bir ses aksiyonu, tek bir ses girdisi ile 100 tane içeriği olan çoktan seçmeli arayüzden seçim yapılmasına olanak sağlar veya tek bir ses aksiyonu ile 100 tane içeriğin listelenmesini sağlayabilir. Böyle bir aksiyon klasik GKA ile yapılmak istendiğinde en az iki aksiyon alınması gerekecektir. Bu durum ses aksiyonunun bu tarz işlemler için daha avantajlı olduğunu göstermektedir.

Eğer ses arayüzü ile tasarlanması düşünülen aksiyon yukarıda belirtilen durumlardan birine uyuyorsa, ses arayüzü bu görev için uygun olabilir.

Ses, sade ve sadece kullanıcı hataları tolere edebiliyorsa girdi olarak kullanılmalıdır [Olive, 1999]. Herhangi bir ses girdisi hatalara sebep olabilir. Bazı görevler için, örnek olarak e-posta okunması, hatalar kabul edilebilirdir. Eğer e-posta okuma komutu hataya sebep olduysa, kullanıcı kolaylıkla ve sıkıntı etmeden tekrar e-posta okuma komutu gönderebilir. Fakat acil amaçlı olan görevler için bu tip hatalar kabul edilebilir değildir. Acil amaçlı geliştirilen bir uygulamanın zaman kısıtı vardır ve aksiyonun bir saniye bile gecikmesine sebep olacak hatalar hayati tehlikeye sebep olabilir.

Ses çıktısı az miktarda bilgi verilme amaçlı kullanılmalıdır. Ses çıktıları kısa tutulmalıdır ve uzun çıktılar vermek için ses kullanılmamalıdır [Cohen, 2004]. Ses çıktıları çok miktarda bilgi verilmesi için uygun değildir çünkü ses geçicidir ve kullanıcı unutulabilir. Uzun ses çıktılarının verilmesi bu olasılığı daha da artırır. Örnek olarak, bir tablo hakkında analiz yapması gereken bir kullanıcı, tablonun tamamını incelemeli ve ona göre analizini sonuçlandırmalıdır. Bu işlemin ses arayüzü ile yapıldığı düşünülürse, tablonun her elemanının ses çıktısı olarak kullanıcıya iletilmesi gerekecektir. Bu durumda uzun bir tablo düşünülürse, son tablo elemanına

gelindiği zaman kullanıcı ilk elemanın değerini unutacaktır ve işlemini tamamlayamayacaktır. Bu sebeple, bu tarz görevlerin SKA ile değil GKA ile geliştirilmesi daha uygun olacaktır.

Görevin SKA'ya uygunluğunun ölçülmesi için dikkat edilmesi gereken son kriter ise, çevre etmenlerdir. Sesin girdi olarak alınması veya ses çıktısı üretilmesi çevre etmenler için uygun olmayabilir. Örnek olarak, gürültülü bir ortam ses işlemleri için problem oluşturacaktır. Ses girdisi alınırken, kullanıcının komutlarına başka sesler karışabilir veya kullanıcı üretilen ses çıktısını duyamayabilir. Bu duruma zıt olarak, çok sessiz ortamlar da kullanıcının ses girdisi vermesi veya uygulamanın ses çıktısı üretmesi için uygun olmayabilir. Örnek olarak kütüphane ortamında çalışması için geliştirilen bir SKA uygulaması düşünülebilir.

4.2. Kullanıcı ve Uygulamanın Çıktılarına Karar Verilmesi

Hangi görevlerin ses aksiyonlarına ihtiyacının olduğu belirlendikten sonra, kullanıcının neler söyleyeceğine ve buna karşılık uygulamanın nasıl cevap vereceğine karar verilmelidir. Bu durum görevden göreve değişiklikler göstereceğinden, genel bir çıktı tarzından bahsedilemez.

4.3. Başarıyı Arttırmak İçin Ses Girdilerinin Sınırlanması

SKA uygulamaları için sınırlandırılmayan ses girdileri büyük problemdir. Konuşma dili çok geniştir ve çok fazla sözcük içermektedir. Bu durum, günlük hayat için avantajlı gözükse de, SKA uygulamaları için dezavantajlı bir durumdur. Çünkü çok fazla olan günlük hayat sözcüklerinin uygulama tarafından algılanabilmesi zordur [Yaşa, 2012].

SKA geliştirilirken, ses girdilerinin karmaşıklığı düşünülmelidir. Ses girdilerinin sınırlandırılması, ses algılamının başarısını arttıracaktır [Omogbe, 2010]. Örnek olarak, kullanıcıdan doğrulama amaçlı girdi almanın birçok yolu vardır. Bunlar “Evet”, ”Tamam”, ”Olur”, “Anlaşıldı” vb. gibi girdilerdir. Doğrulama amaçlı olarak kullanıcı sadece “Evet” ile sınırlandırılırsa, uygulamanın bunu algılayabilmesi daha olanaklı olacaktır. Sonuç olarak kullanıcıdan alınacak girdilerin sınırlandırılması ile uygulamanın başarısı artırılabilir, kullanıcı da uygulama için sınırlandırılan girdileri vermeyi zaman geçtikçe daha kolay öğrenecektir.

4.4. Kullanıcılar İçin Eğitim

Bir SKA'da, ilk zamanlar kullanıcı ne söylemesi gerektiğini yada uygulamanın ne anlayabildiğini bilemez. Özellikle kullanıcı girdilerini sınırlandırdığımızı düşünürsek, kullanıcının uygulamanın anlayabildiği komut dizisini bilmesi gerekir. Bazı kullanıcılar, ne söylerlerse uygulamanın anlamasını beklerler, bazıları da ses algılama işlemi başladığı anda ne yapacaklarını şaşırırlar. Bu problemler GKA'da bulunmamaktadır. Çünkü GKA uygulamasında kullanıcı, grafiksel olarak her şeyi görebilmektedir ve sınırlarını bilmektedir. Kullanıcı en kötü ihtimalle deneme yanılma yolu ile uygulamanın ne yapabileceğini çözebilmektedir. Bu sebeplerden bir SKA uygulaması kullanıcıya ne söyleyebileceğini, sınırlarını aktarabilmesi için kullanıcıyı eğitmelidir. Bu durum özellikle grafiksel arayüzü hiç olmayan, eller ve gözler bağımsız amaçlı gerçekleştirilen SKA uygulamaları için çok büyük önem taşımaktadır. Bu amaçlı bir eğitim çeşitli yollarla yapılabilir.

Kullanıcıyı eğitmek amaçlı izlenebilecek yollardan birisi, kullanıcıya komutların gösterilebileceği yardım ekranlarıdır [Milette, 2012]. Bu ekranlar sayesinde kullanıcı hangi komutları kullanabildiğini istediği an görebilecektir. Diğer bir yöntem de, ses çıktıları ile kullanıcıyı yönlendirmektir. Uygulamanın herhangi bir aşamasında kullanıcıya verilecek bir çıktı ile kullanıcının sınırları bildirilmiş olunur. Örnek olarak, kullanıcıdan alınan girdinin teyit edilmesi amacı taşıyan aşamada kullanıcıya "Devam etmek için EVET, işlemi sonlandırmak için HAYIR demeniz yeterli" şeklindeki bir çıktı, kullanıcının "EVET" veya "HAYIR" seçeneklerinin olduğunu bildirir.

4.5. Başarının Teyidi ve Hatalardan Kurtulmak İçin Kullanıcıya Yardım Edilmesi

Ses girdilerini komut olarak algılayıp işlem yapmak riskli bir iştir çünkü çok farklı sebeplerden ses algılama başarısız olabilir. Bu sebeple GKA uygulamalarında hiç olmayacak olan hatalar SKA uygulamalarında gerçekleşebilir [Kortum, 2008]. Bir GKA uygulamasında, kullanıcı bir butona basmak isterse genelde başarılı olmaktadır. Örnek olarak, kullanıcıların GKA uygulamalarında "Tamam" butonu yerine "İptal" butonuna basmaları çok nadir görülen bir durumdur. Fakat buna

karşın SKA uygulamasında bu durum tersi olabilir. Ses algılamanın başarısız olması durumunda, yanlış seçimin yapılma ihtimali devamlı olarak var olmaktadır.

Buna ek olarak, GKA uygulamalarında başarılı olan işlemlerin sonucunda kullanıcı genelde bir görsel çıktı alabilmektedir. Fakat gözden bağımsız bir SKA uygulamasında, işlem sonucunu görebilmek için farklı yollar denenmelidir.

Yukarıda belirtilen iki sebep için, uygulama veya kullanıcı kaynaklı olan hataların düzeltilebilmesi için çözümler üretilmelidir. Bu çözümlerden birisi olan onay diyalogları kullanılırken dikkatli olunmalıdır. Kullanıcının devamlı çıkacak olan onay diyalogları ile sıkılmaması sağlanmalıdır. Kullanıcının ses komutlarının başarılı olduğunu belirtmek için kullanılacak olan onay diyalogları için birkaç öneri aşağıdaki gibidir.

- Görsel olarak onayların kullanılması

Eğer mümkünse, aksiyon sonucunun ekranda gösterilmesidir.

- Üstü kapalı onayların kullanılması

Kritik olmayan aksiyonlar için, uygulama ses algılamanın başarılı olduğunu varsaymalıdır. Sonucun onayını aksiyon yanıtında vermelidir. Örnek olarak, “Kimlik sorgula” komutu için “Kimlik sorgulama başladı” yanıtı, kullanıcıya uygulamanın “Kimlik” komutunu algıladığını gösterir. Eğer uygulama “Plaka sorgulama başladı” gibi bir yanıt verecek olursa, kullanıcı komutun yanlış algılandığını anlayacaktır. Eğer uygulama sadece “İşlem başladı” vb. gibi bir yanıt dönecek olursa, kullanıcı hangi işlemin başladığını bilemeyecektir.

- Belirgin onayların kullanılması

Belirgin onaylar her zaman gerekli değildir. Örnek olarak, kullanıcının “Kimlik sorgula” komutundan sonra uygulamanın “Kimlik sorgulama başlasın mı?” yanıtı gereksiz ve zaman kaybı olacaktır. Buna karşılık belirgin onaylar, bazı durumlarda kullanılabilirler. Geri alınması mümkün olmayan işlemler için veya uzun vakit alacak işlemler için belirgin onaylar kullanılabilirler. Örnek olarak, kimlik sorgu yapma işlemi uzun zaman alabilecek bir işlemdir. Kullanıcı kimlik sorgulama isteyip, kimlik numarasını söylemesi halinde ondan geri bildirim ile onay alınması buna bir örnektir.

- İşlemi telafi etmelere imkân sağlanması

Uygulama yanlış aksiyonu aldığı zaman, kullanıcıya işlemi geri alabilmesi için başka komutların sağlanması gerekmektedir. Bu kullanıcıya, uygulama veya komut algılama başarısız olduğu zaman işlemi telafi etme imkânı sağlayacaktır. Örnek olarak, kimlik sorgulama yapmak istendiği zaman kimlik numarasının yanlış algılandığı bir durumda, “Düzeltil” komutu ile kullanıcının uzun süren sorgulama işlemine başlamadan önce yanlış algılanan kimlik numarasını telafi etme imkânı olacaktır.

Kullanıcıya ses girdisi haricinde aynı işlemi yapabileceği farklı bir metot sunulması da telafi için başka ve fakat tercih edilmemesi gereken bir yöntemdir. Ses algılamanın devamlı olarak başarısız olduğu durumlar veya kullanıcının ses girdisi veremediği durumlar için uygulamanın aksiyonlarını ses girdisi yerine GKA ile yapılabilmesine olanak sağlamalıdır.

- Yardım için kademeli asistanlık kullanımı

Uygulama her başarısız olduğunda farklı, daha detaylı ve kademeli olarak hata mesajları çıkartılmalıdır. Örnek olarak, uygulama ilk başarısız olduğu zaman, “Anlayamadım, lütfen tekrar edin”, ve sonraki hata durumunda “Kusura bakmayın, kimlik sorgula veya iptal komutları kullanabilirsiniz” şeklinde cevap vererek kullanıcıya hem çıktı verilmiş olunur hem de yönlendirilmiş olunur.

4.6. Ses Algılamanın Yanlışlıkla Aktif Hale Getirilmesi

Bir SKA uygulaması ses girdisini algılayamadığı zaman uygulama başarısız olmuş olur, fakat aynı zamanda uygulama ses algılamanın yanlışlıkla aktif edilmesi ile de başarısız olabilir [Milette, 2012]. Özellikle ses algılamanın aktif edilme işlemi gelen ses girdileri ile yapılıyorsa, ses algılama yanlışlıkla aktif edilebilir ve ses algılama başarısız olabilir. Bu gibi durumlarda, uygulama belirlenmiş ses komutları geldiği zaman ses algılama işlemini başlatmış olur (veya kullanıcı yanlışlıkla “Başlat” düğmesi veya benzeri düğmeye basarak ses algılamayı aktif hale getirebilir).

Yanlışlıkla ses algılama aktif edildiği zaman, kullanıcının söyleyecek bir komutu olmadığı için garip bir durum oluşabilir. Bu gibi durumlarda, kullanıcının işlemi iptal edebilmesi için “İptal” gibi bir komutu olmalıdır. Ayrıca bu iptal

seçeneđi, kullanıcının yapmak istediđi fakat vazgeçtiđi işlem den geriye dönmesini de sağlamalıdır.

4.7. Kullanıcıya Net Konuşmanın Öğretilmesi

Kullanıcıların normalden farklı şekilde konuşmaları ses algılamının başarısını düşürmektedir [Yankelovich, 1998]. Bu tip durumlar ses algılama hataları arttıkça, kullanıcı sinirlenmeye başladığı için daha çok artmaktadır. Bu durumun ortadan kaldırılması için, kullanıcıya nasıl konuşulması konusunda yardımcı olunmalıdır.

- Normal konuşulması sağlanmalı

Kullanıcılar sinirlenmeye başladıkları zaman, telaffuzları da değişmektedir. Cümle içindeki sadece bir kelimeye baskı yapmaya başlayabilirler. Örnek olarak, “KİMLİK sorgula” cümlesinde “KİMLİK” kelimesini daha vurgulu söyleyebilirler veya “KİMMMMMLİK sorgula” cümlesinde olduğu gibi bir kelimeyi uzatarak söyleyebilirler. Bu gibi durumlar ses algılamayı daha da zorlaştırır. Ve ses algılama zorlaştıkça ses arayüzü uygulamasının başarısı da düşmektedir.

- Kelimeler arasında duraklama yapılması

Kelimeler arasında kısa duraklamalar yapılması ses algılamının başarısını arttırmaktadır. Bu duraklama, ses algılayıcının bir kelime bitip diğerinin başladığını düşünmesine yardımcı olur ve ayrıca iki kelimeyi peş peşe söylerken meydana gelen bozulmayı da ortadan kaldırmış olur.

5. PROTOTİP SKA UYGULAMASI

Mobil cihazlar üzerinde SKA uygulaması geliştirme yöntemleri önceki bölümlerde belirtilmiştir. Bu bölümde Android işletim sistemi üzerinde geliştirilen prototip uygulamanın mimarisi ve geliştirme yapılırken izlenen yollar anlatılmıştır. Bölüm içerisinde uygulamanın amacı, uygulama için gerekli olan araçlar ve uygulama arayüzleri tanıtılmıştır.

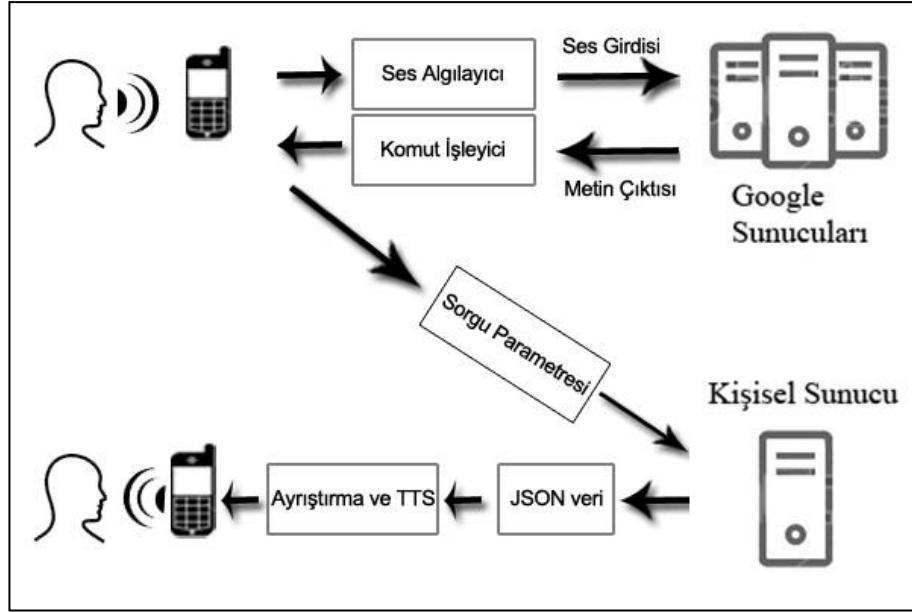
5.1. Uygulamanın Amacı

Geliştirilen Android SKA uygulamasının amacı, güvenlik görevlilerinin kimlik sorgulama için kullandıkları el terminallerinin yerine, elleri ve gözleri bağımsız bir şekilde kullanabilecekleri ve el terminalleri ile aynı görevleri yerine getirebilecek bir ses arayüzlü uygulamadır. Bu tarz bir uygulama ile benzer koşullar altında güvenlik görevlilerinin elleri ve gözleri cihaza bağımlı kalmayacağı için can güvenlikleri de artmış olacaktır [Krstovski, 2003]. Çünkü sorgulama işlemi yaparken, şüpheliyi gözleri ile takip edebilecekler ve elleri ile de önlem alabileceklerdir.

5.2. Uygulama Mimarisi

Geliştirilen uygulama Ses Arayüzü Uygulama Tasarım bölümünde anlatılan maddeler ışığında tasarlanmıştır. SKA tasarımı yapıldıktan sonra, önceden araştırılan ve karar verilen araçlar ile uygulamanın geliştirilmesi gerçekleştirilmiştir.

Uygulamanın mimarisinden genel olarak şu şekilde bahsedilebilir. Geliştirilen uygulama Android işletim sistemli cihazlar üzerinde çalışmakta ve Android ses uygulama arayüzlerini kullanmaktadır. Android uygulama arayüzleri sesten-metin çeviri işlemleri için Google sunucularını kullanmaktadır. Sesten-metin çeviri sonuçlarını Google sunucularından alarak işlem yapan uygulama son olarak kimlik ve plaka sorgulamalarını yapmak için kişisel sunucuyu kullanmakta ve sorgulama sonuçlarını buradan almaktadır. Şekil 5.1'de uygulamanın genel mimarisi gösterilmektedir.



Şekil 5.1: Uygulama Mimarisi

Geliştirilen SKA uygulamasının genel mimarisi yukarıda belirtildiği gibidir. Sonraki bölümlerde çalışmanın ayrıntıları ve çalışma mantığı anlatılmıştır.

5.2.1. Android İşletim Sistemi ve Ses Arayüzleri

SKA uygulaması geliştirirken ilk akla gelen soru, ses nasıl girdi olarak alınıp işlenebilir olmaktadır. Ses üzerinde çalışmak zaman ve maliyet isteyen bir işlem olduğundan dolayı, ses girdilerini uygulamaya komut olarak gönderebilmenin en kolay ve hızlı yolu, ses girdilerinin metne çevrilip, metin üzerinden işlemlerin sürdürülmesidir. Metne çevrilen ses girdileri, uygulamanın anlayabileceği komutlar olarak işlev görmektedir.

Android işletim sistemi üzerinde ses işlemlerinin yapılabilmesini sağlayan uygulama arayüzleri mevcuttur. Bu uygulama arayüzleri SKA uygulaması için gerekli olan sesten-metin ve metinden-ses çeviri işlemlerinin yapılabilmesini sağlamaktadır. Bölüm devamında ses uygulama arayüzleri hakkında bilgi verilmiştir.

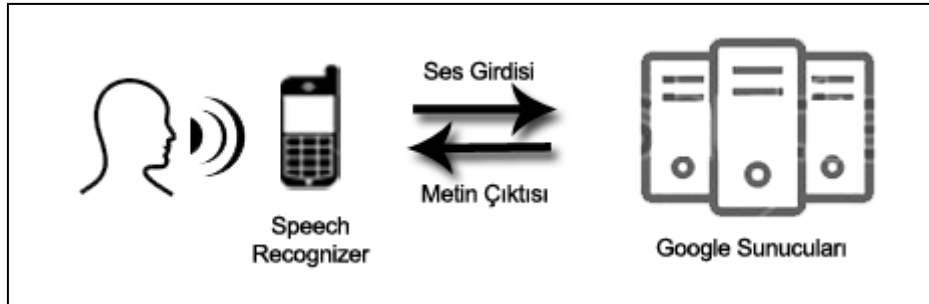
5.2.1.1 Ses Algılayıcı (Speech Recognizer)

Ses algılayıcı, Android işletim sisteminde gömülü olarak bulunan bir uygulama arayüzüdür. Bu arayüz sayesinde sesten-metin çevirisi yapılabilmektedir.

Ses algılayıcı uygulama arayüzünün çalışma mantığı şu şekildedir. Kullanıcıdan ses girdileri cihaz üzerindeki mikrofon sayesinde toplanır. Toplanan ses bilgileri metne çevrilmek için uygulama arayüzü tarafından geliştirici için nerede olduğu önemli olmayan Google sunucularına gönderilir. Google sunucularında işlenen ses girdileri, en yakın metin değerlerine çevrilirler. Çeviri sonucunda liste şeklinde metin olarak birden fazla sonuç döner. Örnek olarak, “Kimlik” kelimesi girdi olarak verildiğinde, bu sorgulama için metin olarak geriye dönen listenin ilk üç elemanı şu şekildedir.

kimlik	klinik	kimliği
--------	--------	---------

Elde edilen listenin ilk elemanı, ses girdisi ile eşleşme uyumluluğu en yüksek olan metin değeridir. Tez çalışmasında eşleşen diğer seçenekler daha az değerli olduğu ve kullanıcıyı çeviri sonuçları ile çok fazla meşgul etmemek için, çeviri sonuçlarından ilk eleman komut olarak kabul edilmiş ve işleme bu metin değerleri üzerinden devam edilmiştir. Sesten-metin çeviri işlemi Şekil 5.2: Ses Tanımlayıcı’da özetlenmiştir.



Şekil 5.2: Ses Tanımlayıcı

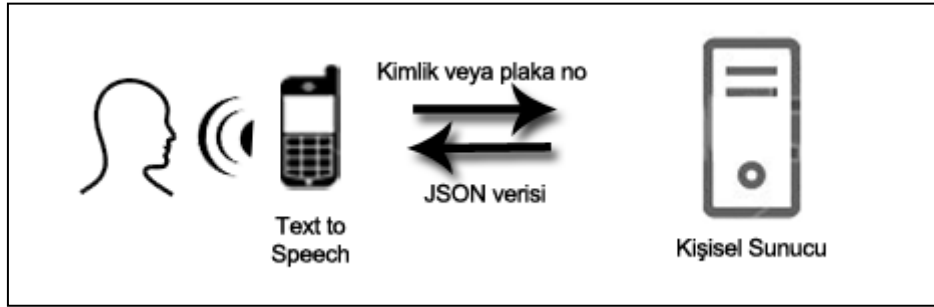
5.2.1.2 Metinden Ses (Text-to-Speech)

Metinden-Ses (Text-to-Speech : TTS), aynı ses algılayıcı gibi Android işletim sistemi üzerinde gömülü olarak gelen ve girdi olarak verilen bir metni ses çıktıları olarak üretilmesini sağlayan bir uygulama arayüzüdür. Metinden ses çeviri işlemini Android üzerinde *TextToSpeech* sınıfı yapmaktadır.

Metinden-Ses (MS) arayüzünü kullanmak basittir. Sadece çevrilmesi gereken metni girdi olarak uygulama arayüzünün *speak* metoduna parametre olarak vermek yeterli olacaktır. Fakat dikkat edilmesi gereken konu, mobil cihaz üzerindeki

Android işletim sisteminde yüklü olan MS dil paketidir. Android üzerinde gömülü olan MS sadece İngilizce çevirileri yapabilmektedir. Ve eğer *TextToSpeech* sınıfı Türkçe ile oluşturulmak istenirse, MS dil paketi İngilizce olduğundan çeviri yapılamayacak ve dil paketi yüklenmesi gerekecektir. Bu durum geliştirilen SKA uygulamasında grafiksel arayüz yardımı kullanıcıya uygun hata mesajı çıkartılarak kotarılmıştır.

Android işletim sistemi üzerinde gömülü olarak gelen MS'in bir diğer özelliği de çevirileri yaparken robotik ses çıktıları üretmesidir. Kullanıcılarla yapılan deneyler sonucunda varılan sonuçlardan biri olarak, kullanıcıların robotik seslerden hoşlanmadığı ve insan sesine daha yakın ses çıktılarını istedikleri söylenebilir. Google tarafından MS için bir Türkçe dil paketi yayınlanmadığından ve robotik ses çıktıları istenmediğinden, çalışmada SWOX firmasının Türkçe dil paketi kullanılmıştır. Bu dil paketi ile metin-ses çevirileri hem daha insan sesine yakın çıktılara dönüştürülebilmekte hem de çevrimdışı çalışabilmektedir.



Şekil 5.3: Metinden Ses Çıktısı

5.2.2. PHP ve JSON

Bu bölümde Android işletim sistemi üzerindeki ses arayüzleri ile sesten-metin çevirisi yapılarak alınan girdilerle yapılan sorgulamanın teknik detayları verilmiştir.

Metin haline çevrilen sorgulama kıstasları, kişisel sunucudaki veritabanında bulunan tablolardan sorgulanarak sonuçlar kullanıcılara sesli olarak aktarılmaktadır. Sistemin sunucu tarafı PHP programlama dili ile geliştirilmiş olup, sunucuda bulunan veritabanı tablolarından kimlik veya plaka sorgulama işlemini gerçekleştirmektedir. Sunucudaki kimlik ve plaka tablolarının yapıları Tablo 5.1: Kimlik Tablosu ve Tablo 5.2 : Plaka Tablo'larında gösterilmiştir.

Tablo 5.1: Kimlik Tablosu

T.C. Kimlik No	Ad	Soyad	Baba Adı	Ana Adı	Doğum Yeri
----------------	----	-------	----------	---------	------------

Tablo 5.2 : Plaka Tablosu

Plaka	Araç sahibi adı	Araç sahibi soyadı	Markası	Modeli	Rengi	Üretim yılı
-------	-----------------	--------------------	---------	--------	-------	-------------

Sunucu tarafında sorgulama işlemi tamamlandıktan sonra elde edilen veriler JSON formatına çevrilip veri olarak istekte bulunan istemciye dönülür. Örnek bir kimlik sorgulamasından elde edilen JSON formatındaki veri şu şekildedir.

```
{"tckno":2147483647,"name":Ahmet,"surname":Kara,"father_name":Ali,"mother_name":Başak,"birth_place":İstanbul }
```

Yukarıdaki JSON verisi mobil cihaza sorgu sonucu ulaştığı zaman ayrıştırılarak metin haline çevrilir ve kullanıcıya ses çıktısı şeklinde iletilir. Örnek olarak, yukarıdaki kimlik sorgulama işleminin sonucu ayrıştırıldıktan sonra kullanıcıya aşağıdaki gibi sesli olarak aktarılmaktadır.

- Uygulama : Sorgulama tamamlandı.
- Uygulama : Kişi adı : Ahmet
- Uygulama : Kişi soyadı : Kara
- Uygulama :Baba adı : Ali
- Uygulama : Anne adı : Başak
- Uygulama : Doğum yeri : İstanbul

6. HİPOTEZ

Mobil cihazlar üzerinde bir SKA uygulaması geliştirirken uygulanabilecek en iyi yöntemlerin araştırıldığı bu çalışmada, öncelikle şimdiye kadar yapılan çalışmalardan yöntemler tespit edilmiştir. Belirlenen SKA geliştirme yöntemleri ile geliştirilen prototip uygulama kullanıcılarla yapılan deneyler ile test edilmiştir. Yapılan deneyler ile kullanıcılardan alınan geri bildirimler sayesinde kullanılabilirliği arttıran yeni yöntemler geliştirilmiştir. Geliştirilen yeni yöntemler denekler ile tekrar denenerek kullanılabilirliğe olan etkileri gözlemlenmiştir. Bu bölümde, çalışma kapsamında ortaya atılan, mobil cihazlar üzerinde SKA uygulamalarının kullanılabilirliğini arttıran geliştirme yöntemlerinden bahsedilecektir.

6.1. Dilbilgisi (Grammar) Yapısı

Konu hakkında yapılan araştırmalar gösteriyor ki, SKA uygulamaları geliştirilirken, uygulamaların işlemleri genelde menü yapısı ile gerçekleşmiştir. Menüler klasik GKA uygulamaları için çok yararlı bileşenler olmasına rağmen, SKA uygulamaları için bazen gereksiz olabiliyorlar. İçerik karmaşıklığı çok fazla olan bir SKA uygulamasında, menüler yardımı ile sonuca çok aşamalı olarak gidilebilecekken, dilbilgisi yapısı ile tek aşamalı olarak gidilebilir. Bu da kullanıcının uzun sürebilecek işlemini daha kısa sürede gerçekleştirmesine olanak sağlamak ve zaman kazandırabilmektedir.

Çalışma kapsamında dilbilgisi yapısı için, uygulamada belirlenen görevler ele alınmış ve bunlar için dilbilgisi yapısı kurulmuştur. Kullanıcılar, menü yardımı ile yaptıkları işlemleri, dilbilgisi yapısı ile tek bir cümle kurarak yapabiliyor hale gelmişlerdir.

Çalışma kapsamında geliştirilen dilbilgisi yapısı aşağıdaki gibi formül haline getirilebilmiştir.

“<komut kelimesi> <gerekli olan girdi> **”

Şekil 6.1 : Dilbilgisi Yapısı

Yukarıda belirtilen genellemeye göre, prototip uygulamada geçerli olan kimlik sorgulama görevi için dilbilgisi yapısı “KİMLİK 12345678912 *” şeklinde tasarlanmıştır. Bu dilbilgisi yapısında komut kelimesi cümlenin başındaki “KİMLİK” kelimesi ve ondan sonra gelen kelime de T.C. kimlik numarası olarak kabul edilmiş ve bu kurallara göre işlem yapılmıştır. Örnek olarak, uygulama ve kullanıcı konuşması aşağıdaki gibidir.

- Kullanıcı : Kimlik 12345678912 sorgulayalım.
- Uygulama : Sorgulamak istediğiniz kimlik numarası 12345678912. Devam etmek istiyor musunuz.
- Kullanıcı : Evet.
- Uygulama : Kimlik sorgulama yapılıyor. Lütfen bekleyin...

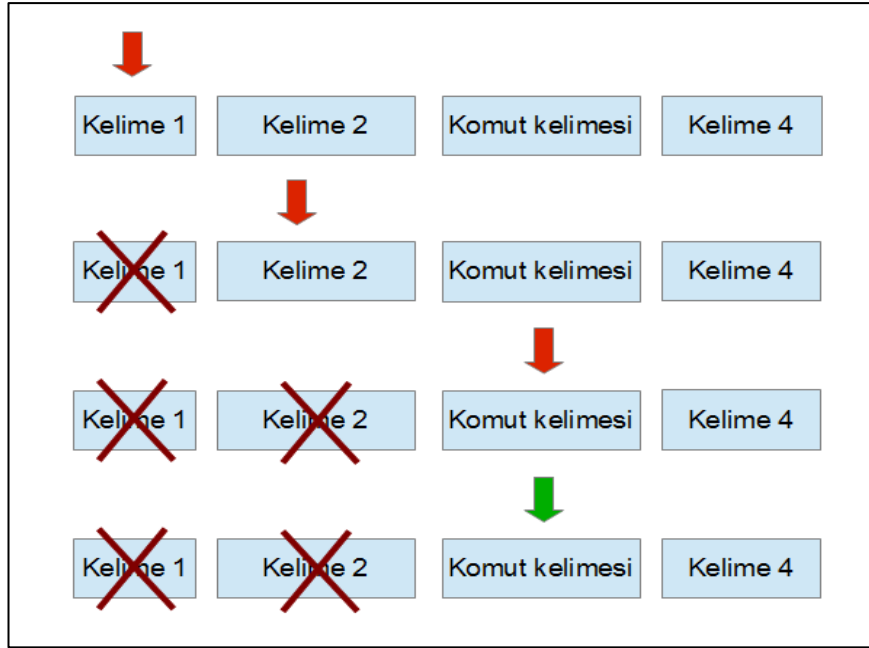
6.2. Komutların Cümle İçinde Aranması

Kullanılabilirlik prensiplerinden olan “Kullanıcının dilini konuşma” prensibinin, yapılan deneyler sonucunda kullanıcılar tarafından önemsendiği gözlemlenmiştir. Kullanıcılar, SKA uygulamaları ile günlük yaşantılarında konuştukları gibi konuşabilmeyi istemektedirler. Buradan yola çıkarak, kullanıcılardan girdi olarak sadece komut kelimelerini beklemektense, onların günlük yaşantılarındaki konuşmalarından komut kelimelerinin algılanabilmesi hedeflenmiştir.

Komutların belirli format olmadan kullanıcıdan alınabilmesi için, ses girdilerinden elde edilen her metin, cümle olarak kabul edilip işleme tabii tutulmuştur. Komut kelimeleri de, elde edilen metin içindeki kelimelerin her birinin kontrol edilmesi ile cümle içinde taranmıştır. Cümlenin kelimelerinin taranması sonucu kelimeler içinde komut kelimesinin bulunması durumunda, kullanıcının söylediği diğer kelimeler göz ardı edilip sadece komut kelimesini söylemiş gibi işlem yapılmıştır. Bahsedilen işlem bilgisayar bilimlerinde “Kelime Tanıma (Word Spotting)” algoritması olarak geçmektedir. Kelime tanıma algoritmasının uygulanması ile SKA uygulamasının kullanılabilirliğinin artması hedeflenmiştir.

Kelime algılama algoritması, cümle içinde her bir kelimeyi kontrol ederek komut kelimesini bulur ve bu sayede kullanıcıların sınırlandırılmasına gerek

kalmamış olunur. Algoritmanın çalışma mantığı Şekil 6.2 : Kelime Algılama Algoritması gösterilmiştir.



Şekil 6.2 : Kelime Algılama Algoritması

6.3. Kelimenin Kökünün Bulunması

SKA uygulamalarında, ses sentezi yapılarak elde edilen metin üzerinden işlem yapılır. Bu tarz geliştirilen uygulamalarda, kullanıcının önceden belirlenmiş olan komut kelimelerini girdi olarak uygulamaya sağlaması gerekmektedir. Örnek olarak, çalışma kapsamında önceden belirlenen komutlardan biri “KİMLİK” kelimesidir ve kullanıcıdan alınan girdi eğer bu komut ise kullanıcı kimlik sorgulama işlemine yönlendirilmektedir. Fakat bu tarz geliştirilen uygulama, kullanıcıdan sadece belirlenen komut kelimesini girdi olarak alır ve eğer komut kelimesi sağlanmadı ise işlem yapamaz.

Kullanıcının dilini konuşma adına, çalışma kapsamında girdi olarak alınan kelimeler, sadece sentezlenen halleri ile değerlendirilmemiştir. Girdi olarak alınan kelimeler kökleri bulunarak işleme tabii tutulmuştur. Örnek olarak kullanıcı, “KİMLİK” yerine “KİMLİĞİ” olarak girdi verse bile doğru komut olarak algılanmış ve kullanıcının işlemi yapılmıştır.

7. YÖNTEM

Hipotezlerin sınanması için kullanıcı deneyleri yöntemi benimsenmiştir. Deneylerde kullanıcılara, üzerinde geliştirilen uygulama yüklü olan mobil cihaz verilmiş, belirlenen görevleri yerine getirmeleri istenmiş ve deneklerin hangi yöntemlere nasıl tepki verdiklerinin gözlemlenmesine karar verilmiştir. İlk adım olarak çalışmanın hipotezlerinin ayrı ayrı sınanabilmesi için, geliştirilen uygulamada özelleştirmeler yapılmış ve her bir hipotezin çalışmasının başlatılmasına veya iptal edilmesine olanak verir hale getirilmiştir. Deneklerden görevleri öncelikle hipotez yöntemleri olmadan yerine getirmeleri istenmiştir ve sonra da hipotez yöntemlerini denemişlerdir. Bu şekilde deneylerle yöntemleri karşılaştırma imkânı elde edilmiş olunmuştur.

7.1. Görevlerin Seçilmesi

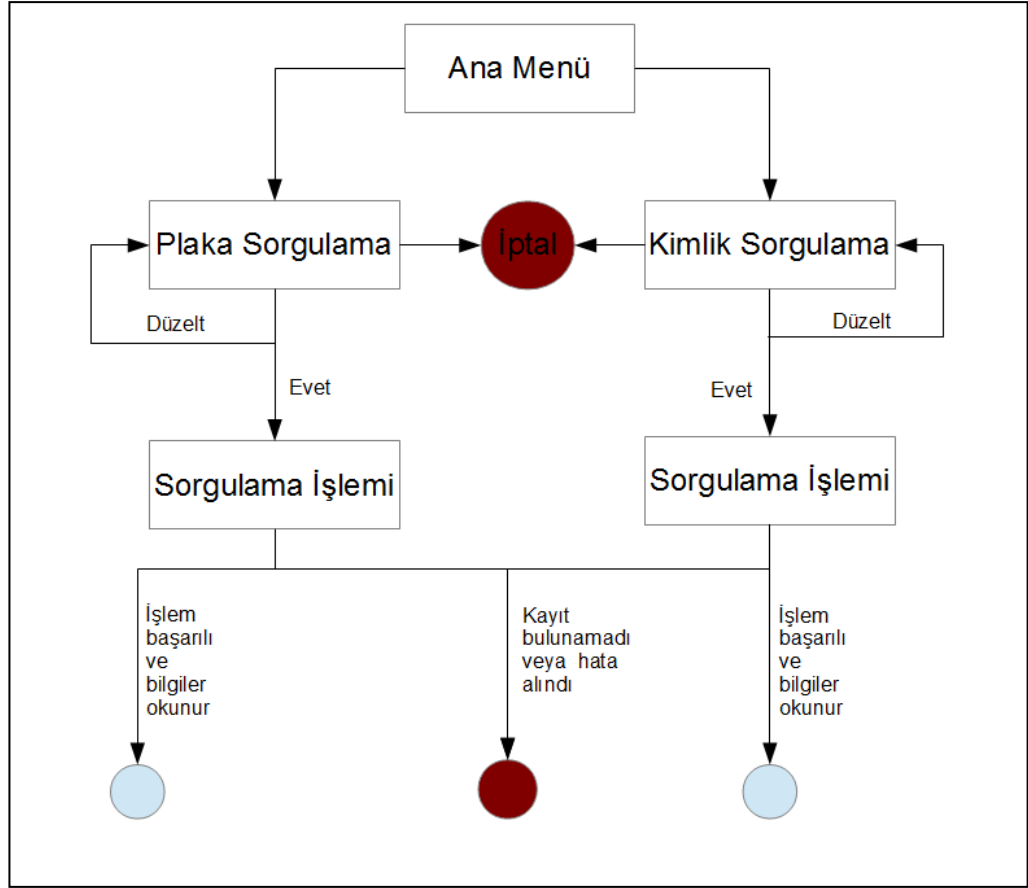
Deneklerin yerine getirmeleri istenen görevler, çalışmada geliştirilen SKA uygulaması güvenlik görevlileri için olduğundan, uygulamanın doğası gereği, T.C kimlik numarası sorgulama ve plaka sorgulama işlemleridir.

7.2. Görev Yöntemlerinin Seçilmesi

Çalışmada, öncelikle Ses Arayüzü Uygulama Tasarım Yöntemleri bölümünde belirtilen yöntemlere göre Android işletim sistemi üzerinde çalışacak bir uygulama geliştirilmiştir. Deney aşamasında deneklere sunulan görevler ilk önce hipotez yöntemleri olmadan, klasik yöntemlerle geliştirilen uygulama üzerinde deneklere sunulmuş ve onlardan geri bildirimler alınmıştır. Bu işlem için belirlenen görevler kimlik ve plaka sorgulama işlemleridir ve bu görevler için yapılan çalışmalar aşağıdaki gibidir.

7.2.1. Menü Uygulaması

Bu aşama için T.C. kimlik sorgulama ve plaka sorgulama görevleri, klasik menü uygulaması ile gerçekleştirilmiş ve kullanıcılara bu şekilde sunulmuştur. Uygulamanın bu hali Şekil 7.1'den kolayca anlaşılabilir.



Şekil 7.1 : Menü Uygulama Mimarisi

Deneklere öncelikle Şekil 7.1 : Menü Uygulama Mimarisi'ndeki gibi işleyen SKA uygulaması sunulmuştur. Bu kısımda, şekilden de anlaşılacağı gibi kullanıcı, uygulama ile devamlı olarak iletişim içindedir ve bir işlemin bitmesi için birkaç defa ses girdisi alınıp ses çıktısı üretilmektedir.

Menü işlemlerinin kısaltılması için hipotez olarak dilbilgisi yapısı ortaya sürülmüş ve dilbilgisi yapısı ile menü hiyerarşisi SKA görevleri için karşılaştırılmıştır.

7.2.2. Sadece Komut Kelimesinin Beklenmesi

SKA uygulaması geliştirirken komut kelimeleri üzerinden işlem yapılmaktadır. Kullanıcıdan alınan ses komutları, uygulamanın nasıl davranacağına karar vermek için kullanılır. Bununla birlikte SKA uygulamasının ses komutlarını kullanıcıdan nasıl aldığı kullanılabilirlik açısından önem taşımaktadır.

Çalışmanın bu aşamasında SKA uygulaması, kullanıcıdan sadece komut kelimelerini, yanında başka kelime olmadan beklemektedir. Örnek olarak, kimlik sorgulama işlemi için kullanıcı sadece ve sadece “KİMLİK” kelimesini söylemelidir. Bunun dışındaki girdiler hata olarak algılanmaktadır. Örnek olarak “Kimlik Sorgulama Yapalım” cümlesi, kullanıcı tarafından komut kelimesi direk olarak verilmediği için hataya sebep olacaktır. Aslında bu durum, kullanılabilirlik ilkelerinden “Kullanıcının dilini konuşma” ilkesine pek de uymayan bir yöntemdir ve buna karşın kurulan hipotez yöntemi Kelime Tanıma Algoritması ile gerçekleşmiştir. Hipotez yöntemi hakkında detaylı bilgi Bölüm 6.2’de verilmiştir.

7.2.3. Komut Kelimesinin Köküne İnilmeden Beklenmesi

SKA uygulamasının başarısını etkileyecek başka bir yöntem de komut kelimelerinin algılanmasıdır. Bu aşamada kullanıcının komut kelimesini saf hali ile girdi olarak vermesi beklenmektedir. Örnek olarak yine kimlik sorgulama görevi üzerinden gidilecek olunursa, kullanıcının kimlik sorgulama işlemi için uygulamaya sadece "KİMLİK" kelimesini girdi olarak vermesi gerekmektedir. "KİMLİĞİ", "KİMLİĞE" vb. gibi girdiler uygulamanın hataya düşmesine sebep olacaktır çünkü komut kelimesi söylenmemiş kabul edilecektir. Bu duruma karşıt sunulan hipotez yöntemi ise kelimenin köküne göre işlem yapılmasını önermektedir ve 6.3’te anlatılmıştır.

Çalışmanın bu aşamasında SKA uygulaması, Ses Arayüzü Uygulama Tasarım Yöntemleri bölümünde belirtilen yöntemler ışığında, kullanıcıdan sadece komut kelimelerini girdi olarak vermesini beklenmiştir, beklenen format dışındaki girdiler hata olarak nitelendirilmiştir.

7.3. Karşılaştırılacak Görev Yöntemlerinin Seçilmesi

Bu bölümde, Ses Arayüzü Uygulama Tasarım Yöntemleri bölümünde belirtilen yöntemlere göre geliştirilen uygulamanın kullanılabilirlik açısından daha iyi olması için Hipotez bölümünde belirtilen yöntemlerle birlikte gerçekleştirildi. Önceki deneylerde deneklere sunulan görevler bu kez hipotez yöntemleri ile gerçekleştirilmiş şekilde deneklere sunulmuş ve onlardan geri bildirimler alınmıştır. Bu işlem için yapılan aşamalar sonraki bölümlerde anlatılmıştır.

7.3.1. Dilbilgisi Yapısı Uygulaması

Deneklerle yapılan deneyler sonucunda ortak olarak alınan geri bildirimlerden en önemli olanı, menü uygulamasının gereğinden fazla vakit kaybettirdiği ve bunun yerine başka bir yöntem kullanılmasının daha iyi olacağı konusunda olmuştur. Alınan bu yararlı geri bildirim sonrasında, uygulama üzerinde bir dilbilgisi çalışması gerçekleştirilmiştir ve deneklere dilbilgisi yöntemi de sunulmuştur.

Dilbilgisi yöntemi, menü yöntemi ile en az üç aşamada sonlandırılabilen bir görevin, bir aşamada yapılabilmesine olanak sağlamıştır. Şekil 7.1 : Menü Uygulama Mimarisi şeklinden de anlaşılabilceği gibi, kimlik sorgulama veya plaka sorgulama işlemini tamamlayabilmek için en az üç adım işlem yapılması gerekmektedir. Dilbilgisi yönteminde kullanıcı, tarafımızdan belirlenen bir kural halinde girdileri uygulamaya vermektedir. Ve bunun sonucunda, kullanıcı işlem sonucunu tek bir adımda alabilmektedir.

Dilbilgisi yöntemi için geliştirilen kural ise yine kullanıcılardan alınan geri bildirimlerden yararlanılarak oluşturulmuştur. Kullanıcılardan alınan geri bildirimlerin çoğunluğu bu yöntem için, komut kelimesinin cümlede başında, girdi olarak uygulamaya verilmesi gereken kimlik numarası veya plaka numarasının da komut kelimesinden hemen sonra gelecek şekilde olmuştur. Alınan bu geri bildirimler ışığında oluşturulan dilbilgisi kuralı şekilde verilmiştir.

7.3.2. Kelime Algılama Algoritması

Kullanıcılardan alınan diğer önemli geri bildirim ise, uygulama ile günlük hayattaki gibi konuşabilme arzusudur. Bu geri bildirim doğrultusunda, kullanıcılardan komut kelimeleri girdi olarak alınırken, kelime algılama algoritması uygulanmıştır. Bu yöntem ile kullanıcıdan sadece komut kelimesinin beklenmesine gerek kalmamıştır, komut kelimesinin geçtiği bir cümleyi girdi olarak uygulamaya vermesi yeterli olmuştur. Bu şekilde kullanıcılar, günlük hayatta konuştuğu gibi uygulama ile iletişim kurabilmişlerdir.

Tez çalışmasında kelime algılama algoritmasının prototip SKA uygulamasında kullanılmasında ortaya çıkan sonuç aşağıda gösterilmiştir.

- Uygulama : Kimlik sorgulama için KİMLİK, plaka sorgulama için PLAKA komutlarını kullanabilirsiniz.
- Kullanıcı : Bir kimlik sorgulaması yapalım.
- Uygulama : Lütfen 11 haneli T.C. kimlik numarasını söyleyin.

Yukarıdaki örnekte de görüldüğü gibi kelime algılama algoritması ile uygulama, “Bir kimlik sorgulaması yapalım” cümlesindeki “KİMLİK” komutunu algılar ve bu sayede kimlik sorgulama işlemi yapılır.

7.3.3. Kelimenin Kökünün İncelenmesi

Hipotez yöntemlerinden son olarak sunulan kelimenin köküne göre işlem yapmak, yine kullanıcının dilini konuşmak açısından, yani komut kelimelerinin algılanabilirliğini arttırmak için geliştirilen bir yöntemdir. Bu yöntem ile kullanıcı, komut kelimesini harfi harfine söylemek zorunda kalmamış, günlük yaşantısındaki gibi eklerle bile konuşup uygulamaya girdi olarak verebilmiştir.

Kelimenin köküne bakılması sadece komut kelimesi arandığında çalıştırılmıştır. Bunun dışında kullanıcıdan girdi olarak alınması durumunda ise kelimenin köküne inilme işlemi yapılmamıştır. Komutların algılanması işleminde kelimenin köküne inilmesi, uygulamanın kullanılabilirliğini olumlu yönde etkilemiştir. Bu işlem de örnekle birlikte şu şekilde anlatılabilir.

- Uygulama : Kimlik sorgulama için KİMLİK, plaka sorgulama için PLAKA komutlarını kullanabilirsiniz.
- Kullanıcı : Bir aracın plakasını sorgulayalım.

Örnek diyalogda görülebileceği gibi, “PLAKA” komut kelimesi eklerle birlikte uygulamaya girdi olarak verilmiştir. Kelimenin köküne göre işlem yapılmadığı durumlarda, uygulama komut kelimesini verilen girdide bulamayacak ve algılama yapamayarak kullanıcıya hata bildiriminde bulunacaktı. Fakat bu gibi durumlarda deneylerden gözlemlediğimize göre kullanıcılar, girdi olarak komut kelimesini verdiklerini ve hata alınmaması gerektiğini savunmaktadırlar. Kullanıcılardan alınan bu geri bildirim için de sunulan kelimenin köküne göre işlem yapma yöntemi, yine SKA uygulamasının kullanılabilirliğine olumlu yönde etki etmiştir.

8. SKA’LARDA KULLANILABİLİRLİK DEĞERLENDİRMESİ

SKA’ların teknik olarak değerlendirilmesi, daha sık yapılan ve daha somut sonuçları olan bir çalışmadır. Diğer taraftan, kullanılabilirlik açısından değerlendirilmesi, çok gerekli olmakla birlikte, henüz oturmuş yöntemler bulunmadığı için teknik değerlendirmeye kıyasla daha zordur.

SKA’ların kullanıcı merkezli bir şekilde tasarımı için ilk defa kullanıma sunulmadan önce mutlaka ayrıntılarıyla değerlendirilmesi ve elde edilen bulgulara göre adımlar atılması gerekmektedir. Ayrıca SKA’lar üzerinde yapılacak değişikliklerin yayınlanmadan önce de, değişikliklerin kullanılabilirliğe zarar vermediğinden emin olmak için değerlendirme yapılmalıdır.

8.1. Sezgisel Değerlendirme

Sezgisel değerlendirme, kullanıcı arayüzünün tasarımındaki problemleri ortaya çıkarmak için kullanılan geleneksel yöntemlerden biridir. Kullanıcı arayüzü, uzman bir değerlendirici tarafından, belirli bir kıstas kümesine göre değerlendirilerek o kriterlere göre uyum ölçülür. Sezgisel değerlendirme yöntemi, ses kullanıcı arayüzlerinin değerlendirilmesinde de kullanılabilir. Bu çalışmada SKA’ların sezgisel değerlendirilmesinde kullanılmak üzere, Ses Arayüzü Uygulama Tasarım Yöntemleri bölümünde anlatılan faktörlere göre yapılması önerilmektedir.

8.2. Kullanıcı Deneyleri

İnsan-bilgisayar etkileşimi hedeflerini yerine getirilebilmesi için dikkat edilmesi gereken en önemli nokta, kullanıcının zihinsel modelinin doğru anlaşılmasıdır. Bu amaçla, kullanıcı ile uygun olan ortamlarda bazı deneyler yapılır. Denekler deneyin amacına ulaşabilmesi için daha önce SKA alanında çalışmamış ve tamamen son kullanıcı gözünden işlem yapabilecek kişiler arasından seçilmelidir. Bu çalışmalar 3 aşamalı olarak tasarlanmıştır.

8.2.1. Sesli Düşünme

Sesli düşünme protokolü (“think aloud protocol”), kullanıcı grafik arayüzlerinin kullanılabilirlik açısından değerlendirilmeleri için yapılan deneyler sırasında, kullanıcıdan doğru veriyi toplayabilmek için sıkça başvurulan bir yöntemdir. Kullanıcı kendisine verilen görevleri gerçeklerken aklından geçenleri sesli ifade ederek, zihinsel modelindeki ürün ile değerlendirilen ürün arasındaki farkı ortaya çıkarır.

Sesli düşünme yöntemi, GKA uygulamalarının değerlendirilmesi için kullanılabilirdiği gibi SKA uygulamalarının değerlendirilmeleri için de kullanılabilir. Bu kapsamda, her bir deneğe farklı görevler verilir ve bu görevleri gerçeklemeleri istenir. Fakat görev gerçekleştirme aşamasından önce, deneklere uygulamanın içeriği anlatılır ve sadece sesli olarak anlatılan uygulama hakkında düşünce ve yorumları alınır. Uygulamayı görmeden ve kullanmadan alınan bu geri bildirimler, aslında ilk beklentilere denk düştüğü için çok değerlidirler. Böylece, kullanıcı beklentileri ilk defa yazılı olarak ifade edilmiş olur. Bu bilgilerle SKA'nın kullanıcının zihinsel modeline ne kadar uyduğunun değerlendirmesi yapılır. Yazılı ifadenin ardından, görev gerçekleştirme aşamasına geçilir ve deneklerin her birine görevler verilir ve bu görevleri yerine getirmeleri istenir. Görevler gerçekleştirilirken sesli düşünme protokolü uygulanır. Kullanıcı, karşılaştığı problemleri ya da düşünmekte olduğu şeyleri kendi cümleleri ile ifade eder.

8.2.2. Görev Gerçekleme

Bu aşamada deneklerden prototip uygulama için geçerli olan kimlik ve plaka sorgulama görevlerini, SKA uygulaması üzerinde kullanarak gerçeklemeleri istenir. Görev gerçekleştirme aşaması boyunca kamera ile kayıt yapılır. Bu yöntemle aşağıdaki veriler toplanabilir [Shneiderman, 1997].

- Öğrenme Zamanı

Kullanıcıların uygulamayı kullanmayı öğrenebilmeleri için geçen zamandır. Kullanıcılar uygulamayı ilk kullanmaya başladıklarında, uygulamaya girdi olarak

vermeleri gerekenleri ve uygulamanın karřıt davranıřlarını bilmemektedirler. Uygulamayı kullanmaya bařladıklarında öğrenme ařamaları bařlamaktadır.

- İşlem Hızı

Kullanıcıların verilen görevleri yerine getirmek için harcadıkları zaman. Her bir görev için farklı yöntemlerle yerine getirirken harcadıkları zamandır.

- Hata Sayısı

Kullanıcıların verilen görevleri yerine getirirken yaptıkları hataların türleri ve sayısı ile SKA uygulamasının hataya düşme sayısı (Kullanıcılar hata yaparken işlem hızları da olumsuz etkilenecektir. Fakat kullanıcıların yaptıkları hatalar, kullanılabilirlik problemlerine doğrudan etki ettiği için hataların da ayrıca değerlendirilmesi gerekmektedir).

Deneylerin tamamlanmasıyla elde edilen veriler, yöntemler arasında karşılaştırma yapılabilmesine olanak sağlamıştır.

9. DENEY DEĞERLENDİRMESİ

Bu bölümde, yöntem kısmında tanımlandığı şekilde oluşturulan SKA uygulama geliştirme yöntemlerinin, değerlendirilmesine ait detaylar anlatılmaktadır.

9.1. Örneklem

Her bir hipotez yöntemi için 10'ar deney yapılmıştır. Tüm deneyler için, deney örneklemini 10 kişiden oluşmaktadır. Her katılımcı uygulamanın kimlik ve plaka sorgulama görevlerini birbirlerinden bağımsız bir şekilde gerçekleştirmişlerdir. Deneylerin tamamı Türkiye'de gerçekleştirilmiştir. Tüm kullanıcıların anadili Türkçedir.

Kullanıcılar, farklı görüş açılarının alınabilmesi için farklı alanlardan bulunmuştur. Kullanıcılardan 7'si mühendislik alanında yüksek lisans veya doktoralarına devam eden ve aynı zamanda aktif olarak çalışan kişilerden seçilmiştir. Diğer 3 kişi de bir kurumda güvenlik görevlisi olarak çalışanlardan kişilerden seçilmiştir. Deneklerin yaşları 28 ile 32 arasında değişmektedir ve yaş ortalamaları 29,72'dir.

Katılımcıların hiçbiri, şimdiye kadar herhangi bir ses kullanıcı arayüzü ile geliştirilmiş bir uygulama kullanmamışlardır.

Deneklere, deney başlamadan önce uygulamanın amacı anlatılmış ve uygulamayı kullanmadan önce uygulama hakkında kendi düşüncelerinin oluşması hedeflenmiştir.

9.2. Deney Ortamı

Deneyler, kullanıcılara uygun olan çeşitli mekânlarda gerçekleştirilmiştir. Kimi deneyler seçilen deneklerin çalıştıkları kurumda gerçekleştirilmiş kimi ise de eğitim aldıkları üniversitede gerçekleştirilmiştir.

Çalışmamız mobil cihazlar üzerinde çalışan SKA uygulaması olduğu için, deneyleri birden fazla mekânda gerçekleştirmek sorun teşkil etmemiştir. Deneyler sırasında uygulamanın yüklü olduğu ve deneklerin görevleri yerine getirmek için kullandıkları cihaz ASUS Nexus marka tablet cihaz olmuştur. Cihaza ait donanım bilgisi Tablo 9.1'de verilmiştir. Bununla birlikte video kayıt işlemi için çeşitli

cihazlar kullanılmıştır. Yapılan kayıtlar, ölçümlerin kontrol edilmesini ve eksik olanların kayıtlardan toplanmasını sağlamıştır.

Tablo 9.1 : Donanım Bilgisi

İşlemci Bilgisi	NVIDIA® Tegra® 3 dört çekirdekli işlemci
Bellek Bilgisi	1.00 GB
İşletim Sistemi Bilgisi	Android4.2 (JellyBean)

Kullanıcıların kamera önünde rahat etmelerini sağlamak amacı ile kayıtların 3. kişiler ile paylaşılmayacağı ifade edilmiştir.

Kullanıcılar görevlerini yerine getirirken, tez yazarı gözetmen olarak ilgili ölçümleri yapmış ve bazı notlar almıştır. Kullanıcılar istedikleri zaman soru sorabilmişlerdir.

Kullanıcıların verilen görevleri gerçeklemeleri için herhangi bir zaman kısıtı konmamıştır. Kullanıcıya öncelikle görevler anlatılır ve görevleri gerçeklemesi istenir. Kullanıcının deney sırasında SKA yöntemleri arasında geçişlerde zaman kaybetmemesi için, SKA uygulamasının üzerinde yöntemler arası geçişleri sağlayacak grafiksel arayüz bileşenleri konmuştur. Bu arayüz bileşenleri sayesinde istenen yöntem kolaylıkla test edilebilir hale getirilmiştir.

9.3. Deney Aşamaları

Katılımcılar seçildikten ve ortam ayarlandıktan sonra deneyler yapılmıştır. Çalışma için yapılan deney iki aşamadan oluşmaktadır:

- SKA uygulamasının ve görevlerin anlatılması
- Görevlerin uygulama üzerinde gerçekleşmesi

9.4. Ölçüm

Deneyler sırasında gözetmen tarafından bazı ölçümler toplanmıştır. Gerek duyulan kısımlarda toplanan ölçümlerin kontrolü için kaydedilen görüntüler bir kez

daha izlenmiş ve gerekli yerlerde düzeltmeler yapılmıştır. Deneylerde her görev için toplanan sayısal metrikler şunlardır:

- Görevlerin tamamlanma süreleri (saniye)

Kullanıcılara verilen görevlerin tamamlanma süreleri.

- Görevlerde hataya düşme sayıları

Kullanıcıların görevleri yerine getirirken, SKA uygulamasının her bir yöntemde hataya düşme sayısı.

- Görevler hakkında sorulan soru sayıları

Kullanıcıların görevleri yerine getirirken görevler hakkında ve SKA uygulamasının kullanılması konusunda sordukları soru sayıları.

- Kullanılan yöntemlerden hangisinin tercih edildiği

Kullanıcılarla yapılan deneylerde yöntemler de karşılaştırılmıştır ve deney sonucunda kullanıcıların hangi yöntemi daha çok kullanılabiliyor buldukları sorulmuştur.

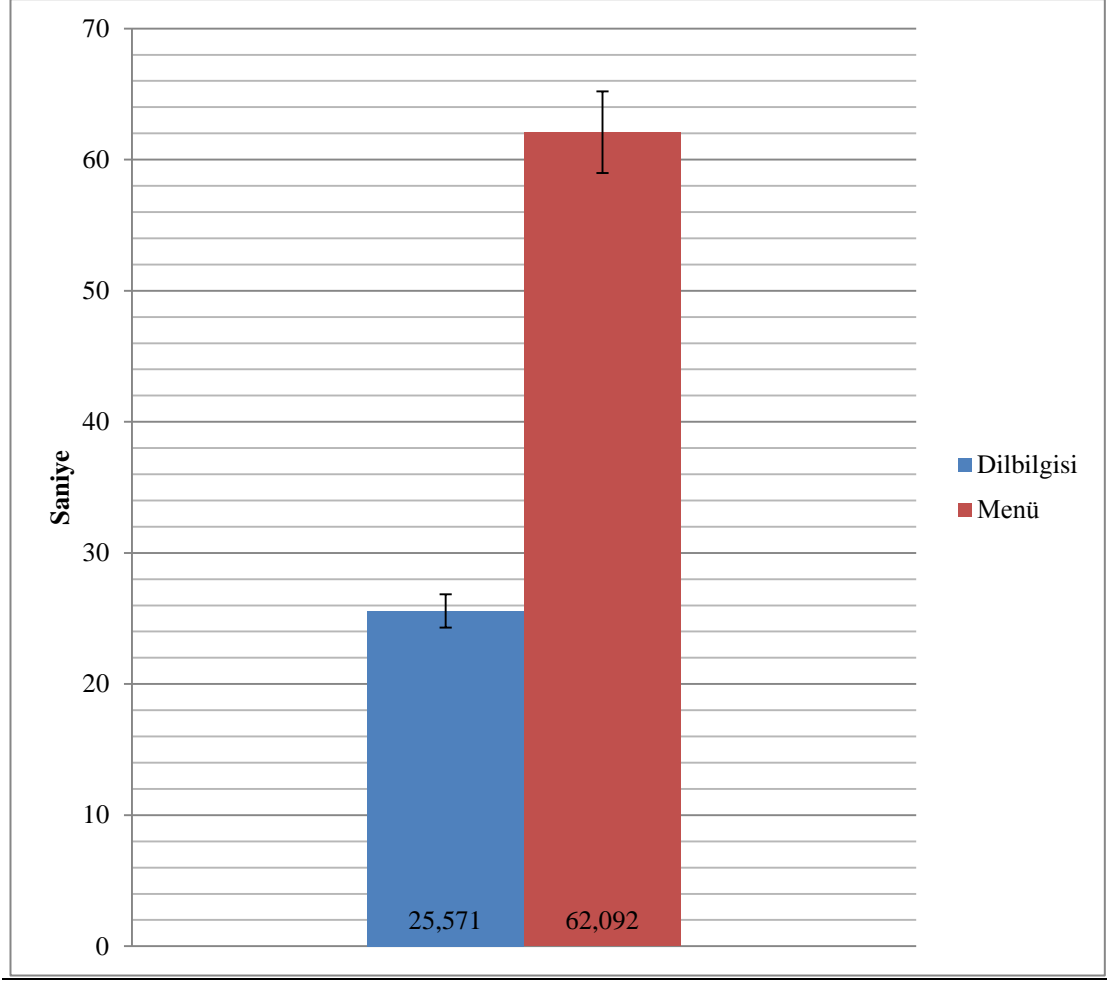
10. SONUÇLAR

Bu bölümde çalışma kapsamında uygulanan deney yöntemleri ve deney sonuçları anlatılacaktır. Deney örnekleme ve deney ortamı konuları Deney Değerlendirmesi bölümünde anlatılmıştır.

10.1. Dilbilgisi ve Menü Yapısı

Deneklerle yapılan ilk deneyler, dilbilgisi ve menü yapısı karşılaştırması için olmuştur. Kullanıcılardan öncelikle kimlik ve plaka sorgulama görevlerini menü yapısındaki uygulama üzerinden gerçekleştirmeleri istenmiş ve aynı görevleri tekrar dilbilgisi yapısındaki uygulama üzerinden gerçekleştirmeleri istenmiştir.

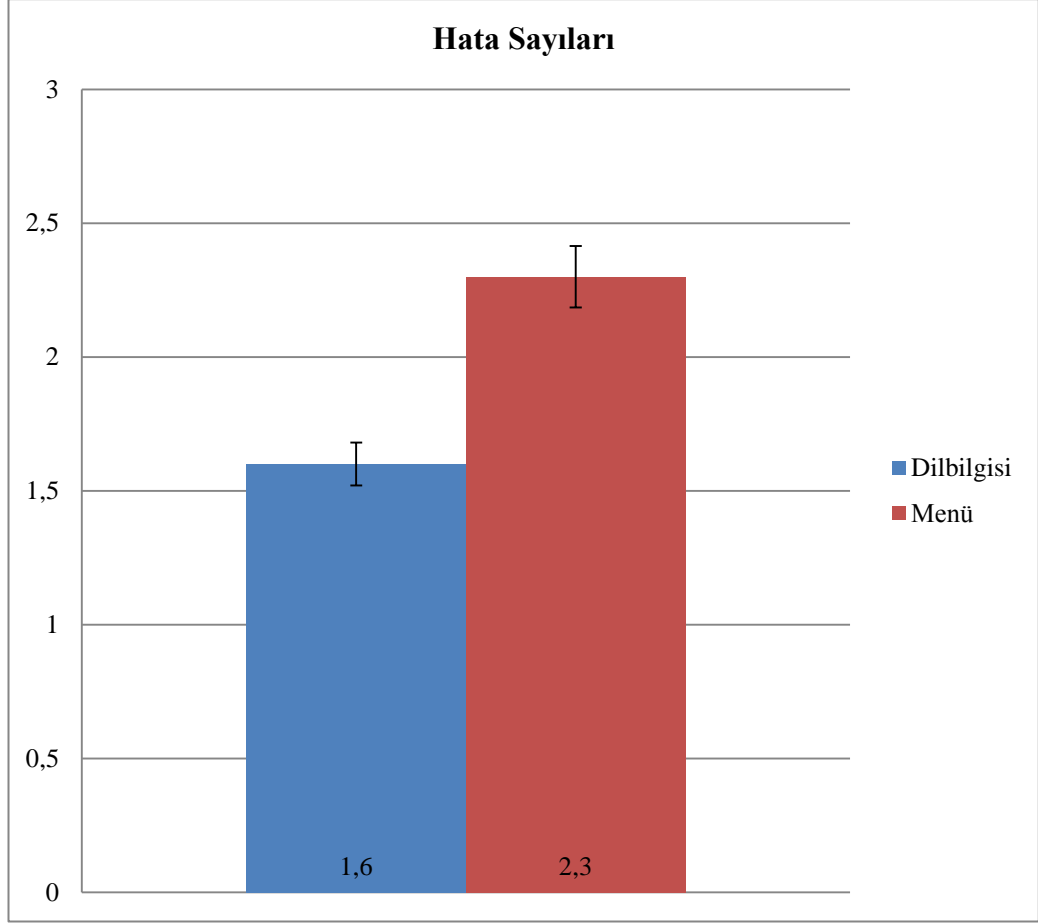
Denekler üzerinde yapılan ilk gözlem, görevlerin yerine getirilme zamanlarıdır. Görevlerin ortalama yerine getirilme zamanları dilbilgisi için 25,57 saniye olurken (1,80s standart sapma), menü yapısı için 62,09 saniye (2,55s standart sapma) olmuştur. Dilbilgisi ve menü yapısının test edildiği deneylerden elde edilen zaman sonuçları Şekil 10.1’de gösterildiği gibidir.



Şekil 10.1 : Görevlerin tamamlanma zamanları

Deneklerin görevleri tamamlama zamanlarına bakılarak, menü yapısının kullanıcılara daha çok zaman kaybettiği gözlemlenmiştir. Kullanıcıların bir işlemi en kısa sürede gerçekleştirmek istedikleri göz önünde bulundurulduğunda, dilbilgisi yapısı SKA uygulamaları için menü yapısından daha kullanılabilir sonucu çıkartılabilmektedir.

Tamamlanma zamanları ölçülen yöntemler, uygulamanın hataya düşmesi bakımından da test edilmiştir. Kullanıcılar ile yapılan deneyler esnasında ortaya çıkan gözlemlerden biri de, SKA uygulamalarında görevlerin tamamlanma aşamaları ne kadar uzun olursa, uygulamanın hataya düşme ihtimalinin o kadar fazla olduğudur. Bunun en önemli sebebi, sesten-metin çevirisi sırasında meydana gelen hatalardır. Dilbilgisi yapısında işlem aşamaları kısa olduğu için, kullanıcıların menü yapısına göre daha az hataya düştüğü gözlemlenmiştir. Bu gözlemlerle ilgili deney sonuçları Şekil 10.2’de verildiği gibidir.



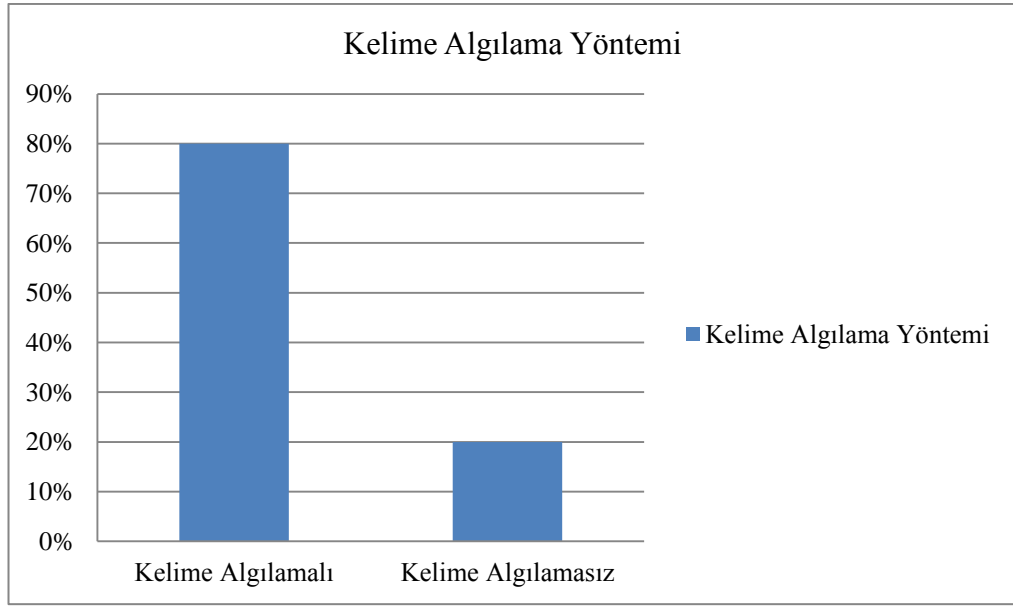
Şekil 10.2 : Görevlerde hataya düşme sayısı

10.2. Kelime Algılama Yöntemi

Dilbilgisi ve menü yapısının karşılaştırılmasının yanında, hipotez yöntemlerinde sunulan kelime algılama algoritması ve kelimenin kökünün alınarak komut kelimelerinin aranması yöntemleri için de deneyler yapılmıştır.

Kelime algılama algoritması yönteminin deneyleri için ilk önce kelime algılama algoritması olmadan kullanıcıların uygulamayı test etmeleri istenmiştir. Bir sonraki adımda kullanıcıların aynı görevleri, kelime algılama algoritması ile birlikte yapmaları istenmiş ve hangi yöntemi daha çok tercih ettikleri sorulmuştur. Genel olarak alınan cevaplar, bahsedilen yöntemin kullanılabilirliğinin daha iyi olduğu yönünde olmuştur. Fakat bazı kullanıcıların(güvenlik görevlisi olarak görev yapan denekler) davranışları izlendiği zaman, özellikle ilk kullanımlarında kullanıcıların uygulama için gerekli olan komut kelimelerini birebir söyledikleri gözlemlenmiştir. Bu gözleme dayanarak aslında yöntemin gerekliliği sorgulanabilirdi fakat kullanıcılar uygulamayı kullanmaya devam ettiklerinde, alışma evresinden sonra

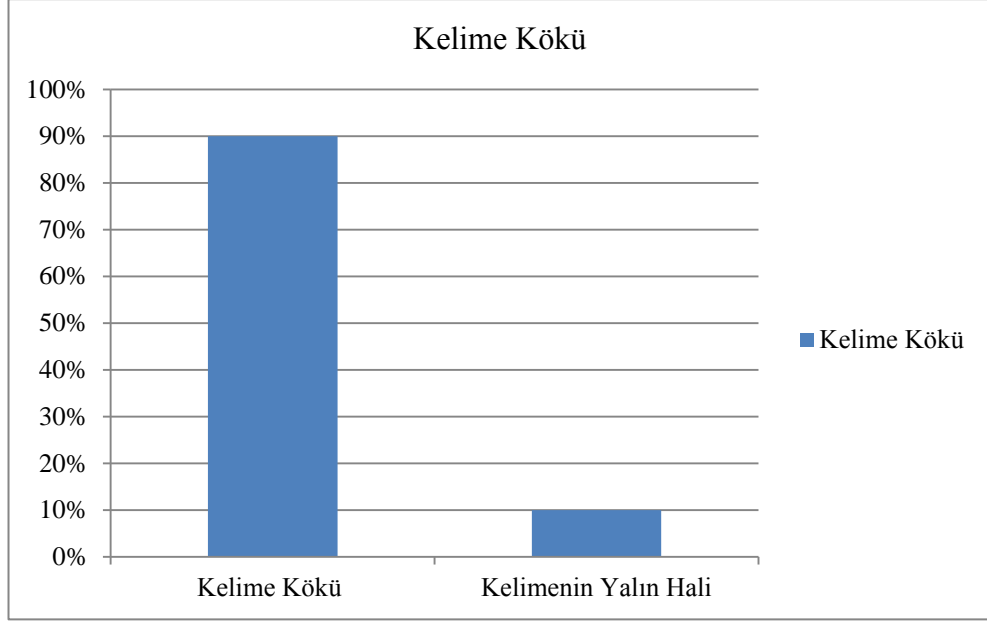
daha rahat davranışlar sergilemişlerdir. Kullanıcılardan alınan geri bildirimlere göre, kelime algılama algoritmasının kullanılabilirliğe etkisi Şekil 10.3'te gösterilmiştir.



Şekil 10.3 : Kelime algılama yöntemi

10.3. Kelimenin Köküne Göre İşlem Yapılması

Son olarak kelimenin köküne göre komutların aranması için ortaya koyulan hipotez yöntemi için de deneyler gerçekleştirilmiştir. Kelime algılama yöntemi için gerçekleştirilen deneylerin aynıları, bu yöntem için de gerçekleştirilmiştir. Kullanıcılardan alınan geri bildirimlerin sonucuna göre, komutlar için kelimenin kökü ile işlem yapmak kullanılabilirliğe pozitif yönde etmektedir. Kullanıcılar uygulamaya komut kelimelerini birebir vermek zorunda olmadıkları için, bu yöntemin yararlı olduğunu belirtmişlerdir. Bunun yanında, bazı kullanıcılar (mühendis denekler), prototip SKA uygulaması için kelimenin köküne göre işlem yapılmasının çok yararlı olmadığı kanısında olmuşlar ve bu yönde geri bildirim vermişlerdir. Bunun sebebi olarak, uygulama için geçerli olan “KİMLİK” ve “PLAKA” komutlarını cümle içinde kullandıklarında genelde yalın halde kullandıklarını ve bu sebepten bu komutlar için kelimenin köküne göre işlem yapmanın kullanılabilirliği çok etkilemediğini savunmuşlar, fakat genel SKA yöntemi olarak kullanılabilirliği pozitif yönde arttırdığını belirtmişlerdir. Bu yöntem için yapılan deney sonuçları Şekil 10.4’te gösterilmiştir.



Şekil 10.4 : Kelimenin köküne göre işlem

10.4. Uygulama Aktifleştiriciler

Kullanıcılar ile yapılan deneylerin son aşaması da uygulama aktifleştiriciler konusunda olmuştur. Uygulama aktifleştiriciler, ses uygulamasının tamamen eller ve gözlerden bağımsız çalışabilmesi amacıyla gerçekleştirilmiştir. Eğer uygulama aktifleştiriciler olmazsa, SKA uygulamasının aktif edilebilmesi için kullanıcı mobil cihaz ekranından bir grafik arayüzü bileşeni ile uygulamayı aktif etmek durumunda kalacaktır. Bu durum ise, SKA amacına hizmet etmemektedir.

Kullanıcılara tamamen eller gözlerden bağımsız bir SKA uygulaması sunabilmek amacıyla tez çalışmasında uygulama aktifleştirici olarak Ses Aktifleştirici gerçekleştirilmiş ve test edilmiştir.

Ses aktifleştirici, Android işletim sisteminin servisleri kullanılarak gerçekleştirilmiş bir arka plan işlemidir. Kullanıcı sorgulama işlemi yapmadığı zamanlarda bile kullanıcıyı dinleyen bu servis özel bir komut kelimesi beklemektedir. Bu özel komut kelimesi tez çalışması için “ASİSTAN” olarak belirlenmiştir. Ses aktifleştirici devamlı olarak kullanıcıyı dinler. Kullanıcı cümle içinde veya doğrudan “ASİSTAN” kelimesini söylediği anda, SKA uygulamasının çalışmasını sağlar ve sorgulama işlemi başlar. Bu şekilde kullanıcı bir sorgulama işlemini bitirdikten sonra başka bir sorgulama işlemini yapmak istediği zaman sadece

“ASİSTAN” demesi yeterli olacaktır ve hiçbir şekilde cihaz ekranında bir grafik arayüz bileşenine dokunmasına gerek kalmayacaktır.

Ses aktifleştirici konusunda kullanıcılar ile deneyler yapılmıştır. Deneklere görevlere başlamadan önce ses aktifleştirici olmanda, sorgulama yapmak istediklerinde ekranda bulunan “BAŞLAT” düğmesine basmaları istenmiştir. Buna karşıt olarak bir sonraki deneylerde deneklerden görevlere ses aktifleştirici ile başlamaları istenmiş ve ses aktifleştiricinin yararının kullanıcılar tarafından fark edilmesi planlanmıştır. Görevlerin bitiminden sonra kullanıcılara hangi yöntemi daha kullanılabilir buldukları sorulmuştur. Deneklerden 7’si ses aktifleştiricinin daha kullanılabilir olduğu konusunda geri bildirim verirken, 3’ü grafik arayüz bileşenin daha kullanılabilir olduğunu söylemişlerdir. Bu deneyle ilgili

Ses aktifleştiricinin kullanıcıların %30’u tarafından kullanılabilir bulunmamıştır ve bu konuda da bir araştırma yapılmıştır. Bu araştırmanın sonucunda ses aktifleştiricinin kullanılabilir seçilmemesinin sebepleri aşağıdaki gibidir.

- Olumsuz yanıt veren deneklerin teknolojiye uzak olmaları (Güvenlik görevlileri)
- Olumsuz yanıt veren deneklerin meslek ve yaş grupları (Güvenlik görevlileri)
- Android işletim sistemindeki ses algılayıcısında bulunan hata (Android ses algılayıcı 5 saniye girdi alamazsa dinlemeyi bırakmaktadır ve tekrar başlatılmak istendiğinde düşük oranlarda da olsa başarısız olunması)

KAYNAKLAR

Azom E. E., (2010), "Mobile Phones Interaction Techniques for Second Economy People", Doctoral Thesis, University of Fort Hare.

Carroll J. M., (1987), "Interfacing thought: Cognitive aspects of human computer interaction", 1st Edition, Bradford Books.

Milette G., (2012), "Professional Android Sensor Programming", 1st Edition, John Wiley & Sons.

Omoregbe I. N., (2010). "A Voice-based Mobile Prescription Application for Healthcare Services (VBMOPA)", International Journal of Electrical & Computer Sciences IJECS-IJENS 10 (02), 69-72.

Kortum P., (2008), "HCI Beyond the GUI", 1st Edition, Morgan Kaufmann Publishers.

Krstovski K., (2003), "Handheld Project54 Speech User Interface", Doctoral Thesis, University of New Hampshire.

Lawrence J. N., Jennifer J. O., Thompson J. C., (1998), "User Interface Design Guidelines for Speech Recognition Applications", IEEE VRAIS, 28-45, Atlanta-USA , 14-18 March.

Lewis C. P., (1990), "Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces", Proceedings of the SIGCHI conference on Human factors in computing systems, 235-242, April.

Lewis J. R., (2011), "Practical Speech User Interface Design (Human Factors and Ergonomics)", 1st Edition, CRC Press.

Cohen M., (2004), "Voice User Interface Design", Addison-Wesley Professional.

Yankelovich N., (1998), "Designing speech user interfaces", CHI 98 Conference Summary on Human Factors in Computing Systems , 131-132.

Nielsen J. ve Molich R., (1990), "Heuristic evaluation of user interfaces", Proc. ACM CHI'90 Conf., 249-256.

Olive J., (1999), "The voice user interface", Global Telecommunications Conference.

Preece J., (1994), "Human Computer Interaction", 1st Edition, Addison Wesley.

Shneiderman B., (1997), "Designing the User Interface", 1st Edition, Addison Wesley.

Welie M., (2001), “Task based user interface design”, Doctoral Thesis, Vrije Universiteit Amsterdam

Web 1, (2013), https://en.wikipedia.org/wiki/Voice_user_interface, (Eriřim Tarihi: 14/09/2012).

Yařa Y., Gktrk. M., (2013), “Gvenlik Grevlileri İin Bulut Biliřim Destekli, Ses Arayz

İle alıřan Asistan Sistemi”, Akademik Biliřim, 907-912, Antalya-Trkiye, 23-25 Ocak.

Neustein A., (2010), “Advances in Speech Recognition”, Springer, 61-91.

DENG L., YU D., (2005), “A Speech-Centric Perspective for Human-Computer Interface : A Case Study”, Journal of VLSI Signal Processing.

ÖZGEÇMİŞ

Yaşar Yaşa 1987 yılında Kocaeli’de doğdu. 2004 yılında başladığı İzmir Yüksek Teknoloji Enstitüsü (İYTE), Mühendislik ve Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Bölümü’nü 2009 yılında başarıyla tamamlayarak, aynı yıl çalışma hayatına başladı. 2010 yılında GTU Bilgisayar Mühendisliği Anabilim Dalında yüksek lisans eğitimine başladı. 2010 yılından bu yana TÜBİTAK BİLGEM’de iş hayatına devam etmektedir.