

2020학년도

광주과학고 기초자율연구 결과 보고서

시각장애인을 위한 애플리케이션 설계

Designing applications for the visually impaired

2020. 11. 18.

연 구 자 : 김동현(1604)
 문병석(1608)
 오정현(1612)
지 도 교 사 : 김은희 선생님



과학영재학교 광주과학고등학교

시각장애인을 위한 애플리케이션 설계

초록

요즘 정보화 시대가 도래되며 많은 기능이 우리가 가진 스마트폰 속으로 들어오게 되며 더욱 접근성이 높아지게 되었다. 이러한 정보화 시대를 바탕으로 TTS를 사용한 여러 가지 기능이 생겨났고, 이러한 기능들을 바탕으로 많은 이들이 삶에서의 편리함을 누리게 되었다. 하지만 많은 이들이 편리함을 누리는 와중에도 몇몇 장애인분은 이러한 편리함을 누리지 못하고 아직도 힘겹게 살아가는 경우가 많다.

우리는 이러한 문제점을 많이 발견하였고 이 부분에 대해 심각성을 느껴 장애인들을 위한 서비스가 필요함을 느꼈다. 또한, 조작의 주는 대부분 터치로 이루어지기 때문에 이러한 디스플레이를 볼 수 없는 시각장애인의 앱을 만들기로 하였다. 물론 우리가 일반적인 운영체제를 장애인들을 위해서 만들 수 있다면 정말 좋겠지만 현실적으로 우리에게 어렵게 느껴졌기 때문에 우리는 시각장애인의 불편함을 덜어줄 수 있는 시각장애인을 위한 애플리케이션을 만들기로 하였다.

가장 먼저 우리 또한 장애인들의 불편함을 깨달아야 하므로 우리도 애플리케이션을 사용해 보고 이를 분석하는 과정을 통해 장애인들에게 필요한 기능을 선정하고 이들을 위한 UI에 대한 구상까지 해보게 되었다. 또한, 현재 출시된 애플리케이션들의 특징과 불편한 점, 보완해야 할 점은 무엇인지 생각해 보았다. 이를 통해서, 물론 UI에 대한 부분에 대해서는 많은 개선이 필요하겠지만 기능적인 부분에 대해서는 가장 먼저 TTS와 OCR 등의 기능을 통해서 시각장애인들의 불편을 덜어줄 만한 기능이 필요함을 깨닫게 되었다.

우리가 실제로 이러한 앱을 개발해 내게 된다면 장애인들을 위한 여러 가지 앱에 많은 프로그래머가 관심을 끌게 될 좋은 기회가 되리라 생각한다. 또한, 앱의 개발을 통해 우리 또한 많은 장애인을 이해하고 이들을 더욱이 도와줄 수 있다고 생각한다. 어쩌면 우리가 만든 앱의 개발을 통해 실질적으로 이들에게 도움이 될 수도 있을 것이다.

현재는 그저 시각장애인들에게 도움이 되는 앱을 만드는 것에서 그쳤지만 나중에 기회가 된다면 직접 시각장애인들을 위한 UI를 바탕으로 한 새로운 OS 체계도 만들어 보고 싶다.

I. 서론

1. 연구 동기

음성합성(Text-To-Speech; TTS)이란 텍스트를 음성으로 반환한다. 이 기술은 시각장애인들 또는 저시력자들을 위한 유용한 기술이다. 광학 문자 인식(Optical Character Recognition; OCR)은 문자의 영상을 기계가 읽을 수 있는 문자로 변환하는 것이다. 이들은 시각장애인들이나 저시력자들이 컴퓨터, 스마트폰 등의 전자매체를 사용할 때 혹은 이들의 정보 접근성 보장을 위해 요즘은 필수적으로 사용된다. 이러한 이유에서 시각장애인들을 위한 보조공학기기가 보급되고 있고, 여러 가지 애플리케이션들이 많이 개발되고 있다. 하지만 시각장애인들의 관점에서는 실용성이 없는 애플리케이션들이 많았다. 대부분 애플리케이션이 버튼식 조작을 기반으로 하여 구동되고 있었다. 하지만 시각장애인들의 처지에서 실용

성의 관점에서 생각해 보면 버튼식의 UI(User Interface)는 시각장애인들의 편의성을 고려하지 못한 것이다. 예를 들어, 사진을 찍어 사진의 글자를 음성으로 반환해주는 기능을 가진 애플리케이션이 있었는데, 시각장애인은 사진을 찍기 어려울 뿐만이 아니라 이를 조작하기 힘들어 보였다. 시각장애인들을 위한 보조공학기기가 보급되고 있지만, 이들은 휴대하기 번거로워서 항상 편리하게 휴대할 수 있는 스마트폰에서 유용하게 쓰일 수 있는 시각장애인을 애플리케이션 설계에 관해 연구하였다.

2. 연구 목적

요즘 많은 기업의 경우 장애인을 위한 여러 부가적인 기능을 제공한다. 애플의 iOS의 경우에는 ‘손쉬운 사용’이라는 기능을 통하여 이들을 돕고, 구글의 안드로이드의 경우에는 ‘접근성’이라는 부분으로 이들을 돕는다. 하지만 이러한 기능들을 보면 일반인을 위한 UI에 장애인들을 도울만한 기능을 추가한 것이다. 즉, 장애인들의 입장에 대한 고려가 크게 이루어지지 않았다는 생각이 들었다. 따라서 휴대하기 쉬운 스마트폰에서 사용할 수 있는 시각장애인을 애플리케이션에 어떠한 기능이 필요하며, 기존의 애플리케이션들에는 없는 요구사항이 무엇이며, 시각장애인들이 사용할 때 불편함이 없는 UI 디자인은 무엇인지 파악한 뒤, 광학 문자 인식과 여러 가지 언어의 음성합성 기술을 이용하여 시각장애인들을 위한 애플리케이션을 설계하고자 하였다.

II. 이론적 배경

1. TTS

가. TTS란

TTS는 Text To Speech를 줄인 말로, 우리말로써 음성합성이라고 한다. TTS는 텍스트를 음성으로 변환한다. 이 TTS 프로그램은 특히 저시력자나 시각장애인에게 잘 활용될 수 있는 편리한 기술이다. 각각의 문장들을 직접 음성으로 출력하여 줄 수 있기에 여러 사용자도 많이 사용하고 있다.

나. TTS의 원리

TTS는 말소리를 기계가 자동으로 만들어내는 기술이다. 사람의 말소리를 녹음하여 일정한 음성 단위로 분할 후, 합성기에 입력하여 필요한 음성 단위를 합쳐 말소리를 만들어낸다. 음성의 분절음을 합성하게 되면 결과가 좋지 않으므로 앞 음성의 뒷부분, 뒤 음성의 앞부분을 함께 기록하여 음성합성을 하기도 하는데, 이를 di phone 처리라고 한다. 즉, TTS 기술은 문자가 입력되면 이 문자열을 여러 단위의 단어로 나눈다 그 뒤, 이 단어들을 다시 음소로 나누고, 이 음소들을 음절 단위로 읽는 방법을 찾아낸다. 마지막으로 음절과 운율 조각을 조합하여 음성파일을 만들어 이것을 출력하는 방식이다.

다. TTS의 활용

TTS의 경우 문장을 직접 소리로 변환해 주는 것인데, 우선 먼저 애플에서 만든 ‘손쉬운 사용’에서 Voice over 기능이다. 이는, 시각장애인들이 이 설정을 켜둔 뒤에 아무 곳이나 한번 누르면 그곳의 정보를 알려주고 두 번 터치하면 그 해당하는 기능을 수행할 수 있다. 두 번째로는 번역기 등을 사용 시에 사용자가 직접 발음을 듣기 위하여 TTS 기능을 통해 이러한 문장을 읽어주게 된다. 또한, 음성합성 기술은 문자를 읽기 어려운 장애인이나 문자를 읽을 수 없는 사람들을 위해서 화면을 읽어주는 기능으로, 말을 하기 힘든 사람이 발표 대체 수단으로 이용하는 예도 있다. 이렇게 여러 곳에 쓰이며 편리함을 도와주는 기능을 하는 역할이다.

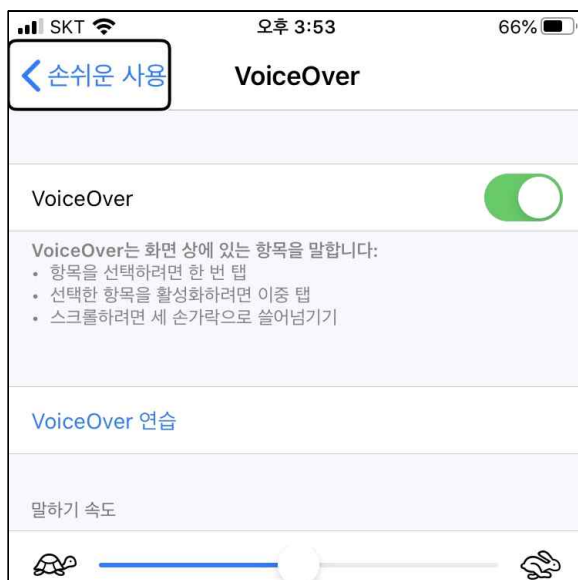


그림 1. Voice over에 대한 설명.

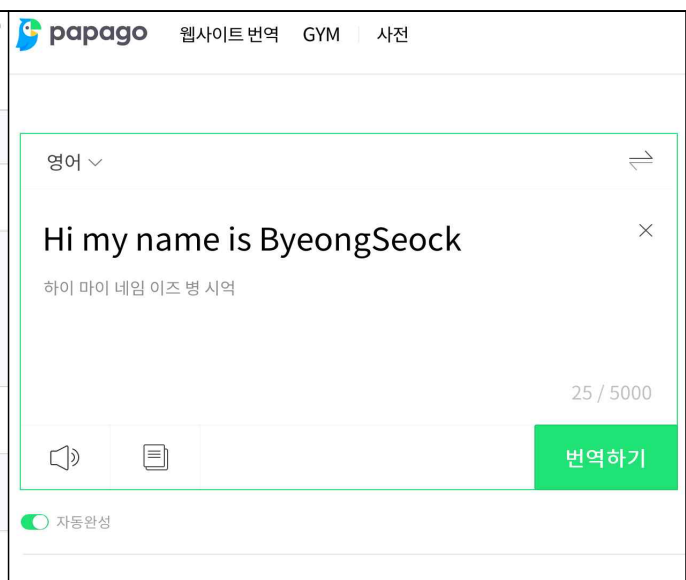


그림 2. 번역기에서의 TTS 기능.

2. API

가. API란

API는 Application Programming Interface를 줄인 말로써, 애플리케이션 소프트웨어를 구축하고 통합하기 위한 정의 및 프로토콜 세트로, 애플리케이션 프로그래밍 인터페이스를 나타낸다. API를 사용하면 구현 방식을 알지 못해도 애플리케이션 개발을 간소화하여 시간, 비용을 절약할 수 있다.

나. API의 원리

1) API의 원리

API는 사용자로부터 요청(Request)하면 결과값을 받아 Parsing 하면 된다. 요청 방법은 REST, SOAP, XML-RPC 등의 형식이 있다. 웹 API가 확산함에 따라, 정보 교환을 표준화하기 위해서 SOAP이라는 프로토콜 사양이 개발되었으나, 규정된 프로토콜보다는 훨씬 간단한 RESTFUL API가 더 많이 사용되고 있다. RESTFUL API란 Representational State Transfer로, REST 아키텍처의 제약 조건을 준수하는 웹 API이다. SOAP은 프로토콜이지만 REST는 아키텍처 스타일이라는 근본적인 차이가 있으므로 RESTFUL 웹 API에는 공식적인 표준이 없다. RESTFUL API는 RESTFUL 시스템의 다음 6가지 주요 제약 조건을 준수해야 한다.

2) RESTFUL 시스템의 6가지 주요 제약 조건

가) 클라이언트 서버 아키텍처

REST 아키텍처가 Client, Server, Resource로 구성되며 HTTP를 통해 요청을 처리한다.

나) 스테이트리스

요청이 통과하는 서버에는 클라이언트 콘텐츠가 저장되지 않으며 그 대신 세션 상태에 대한 정보가 클라이언트에 저장된다.

다) 캐시 가능성

캐싱으로 일부 클라이언트-서버 간의 상호 작용이 제거된다.

라) 계층화된 시스템

추가 계층으로 클라이언트-서버 간의 상호 작용을 조정할 수 있으며 이러한 계층은 로드 밸런싱, 공유 캐시 또는 보안과 같은 추가 기능을 제공할 수 있다.

마) 코드 온디맨드(옵션)

서버가 실행 가능한 코드를 전송하여 클라이언트의 기능을 확대할 수 있다.

바) 통합된 인터페이스

이 제약 조건은 RESTFUL API 설계의 핵심이며 다음과 같은 4가지 측면을 포함한다.

(1) 요청에서 Resource 식별

Resource가 요청에서 식별되며 클라이언트로 반환된 표현으로부터 분리된다.

(2) 표현을 통한 Resource 조작

클라이언트가 Resource를 나타내는 파일을 수신합니다. 이 표현에는 조작 또는 삭제를 허용할 수 있도록 충분한 양의 정보가 포함되어야 한다.

(3) 자기 기술적(Self-descriptive) 메시지

클라이언트에 반환되는 각 메시지에 클라이언트가 정보를 어떻게 처리해야 할지 설명하는 정보가 충분히 포함되어야 한다.

(4) 애플리케이션 상태 엔진으로서의 하이퍼미디어

Resource를 할당한 후 REST 클라이언트가 하이퍼링크를 통해 현재 사용 가능한 기타 모든 작업을 찾을 수 있어야 한다.

3) OpenAPI 개발자들이 REST API 인터페이스를 구축하여 사용자가 별다른 추측 없이도 이를 사용할 수 있도록 한다.

다. API의 활용 방안

한 예로는 어떠한 앱에서 번역기를 돌린다고 한다면 그 앱에서는 API를 통하여 인터넷상에서 돌린 번역 결과를 재생하게 되는 것이다.

3. Android Studio

Android Studio는 안드로이드 앱 제작을 위한 통합 개발 환경이다.

Android Studio는 Kotlin과 JAVA의 언어로 사용할 수 있는데 Android Studio는 개발 사이트가 <https://developer.android.com/studio>로 있다. 그런데 사실상 Android Studio는 Kotlin을 매우 지향해준다. 그 이유는 Android Studio가 Jet Brains에서 만들어진 것이기 때문에 Kotlin을 도구적으로 매우 많이 지원해준다.

4. UI/UX

가. UI (USER INTERFACE)

UI는 사용자가 제품이나 서비스를 사용할 때, 보는 면이다. 즉, 사용자가 제품이나 서비스와 상호작용할 수 있도록 만들어진 매개체이다. 예를 들어, UI 디자인에는 폰트, 색, Layout과 같은 사용자의 디자인을 말한다.

나. UX (USER EXPERIENCE)

UX는 사용자 경험의 약자로, 사용자가 어떤 서비스나 제품을 직간접적으로 이용하며 느끼는 종합적인 만족을 의미한다. UX 디자인이란 사용자에게 만족스러운 경험을 제공하는 UI를 디자인하는 것이다. 즉, 사용자가 만족할 수 있도록 사용자 중심으로 설계된 디자인이라고 할 수 있다.

Ⅲ. 연구 방법 및 절차

1. 사전 조사

현재 출시된 시각장애인들을 위한 애플리케이션의 종류와 사용 방법을 조사하여 시각장애인들이 사용하는 데 어떤 불편한 점이 있는지 정리했다. 책을 읽어주는 기능, 점자 학습 기능 등을 가진 여러 가지 앱들이 있었지만, 연구하고자 하는 기능과 관련된 몇 가지 애플리케이션들을 더 자세히 조사했다.

먼저, ‘보이스 아이’는 인쇄물의 보이스 아이 코드를 인식하여 해당 내용을 음성으로 출력해주는 애플리케이션이다. 따라서, 시각장애인뿐만 아니라 노인과 저시력자에게도 좋은 기능이다. 보이스 아이 코드란? 보이스 코드는 순수 대한민국의 코드 기술인데, QR코드와 비슷한 모양으로, 다른 코드 대비 엄청난 데이터 밀도를 가지고 있어 현존 최고의 데이터 코드이다.

‘샤인 플러스’는 시각장애인을 위한 모바일 접근성 앱이다. 시각장애인, 저시력인 혹은 노안이 있는 사람까지도 이 앱을 통해 편리하게 글자를 읽을 수 있다. 스마트폰 화면에 나온 글을 소리 내 읽어주고, 문자를 확대해 주는 역할이 기본 기능이다. 또한, 음성 안내에 의한 프라이버시 노출을 방지하기 위해 컷속말, 이어 톡 기능이 탑재되어 있다. 이 앱은 저시력인, 노안이 있는 사람에게는 도움이 될 수 있지만, 시각장애인에게는 사용에 어려움이 있다고 생각한다. 왜냐하면, 시각장애인은 웹 서핑 자체에 어려움이 있기 때문이다.

‘탭탭시’는 시각장애인에게 음성으로 사물을 안내해주는 앱이다. 안내받고 싶은 사물에 카메라를 대고 화면을 두 번 두드리면 무슨 사물인지 음성으로 안내한다. 음성 안내를 위해서 ‘보이스 오버’를 따로 설치해야 한다. 또한, 자동초점 기능이 있어서 카메라가 사물을 제대로 식별했는지 소리로 구별할 수 있다. 이 앱에는 위에 Repeat, Library, Share라는 작은 버튼이 있다. 사물을 음성으로 안내받는 것까지는 유용할 수 있지만, 보호자가 없다면 버튼들을 통한 기능들은 이용하기 힘들 것이다. 이 외에도 아이 폴리라는 비슷한 앱이 있는데, 이 앱은 인식하지 못하는 물체를 가르치는 기능도 있어서 사용할수록 물체 인식 능력이 높아진다. 이는 인터넷에 연결하지 않고 사용할 수 있는 것이 장점이다.

‘비 마이 아이즈’는 시각장애인과 자원봉사자를 영상통화로 연결해준 뒤 시각장애인이 도움이 필요할 때 앱을 실행하면 그와 같은 언어를 쓰는 자원봉사자가 이에 응답할 수 있다. 그 후, 시각장애인이 앞에 보이는 장면을 스마트폰으로 촬영한 뒤 자원봉사자가 그 장면에 관해서 설명해준다. 2020년 11월 9일 기준 자원봉사자 4,317,091명, 시각장애인 264,619명이 등록되어있다. 사람이 직접 도와줄 수 있다는 점에서 장점이 있다. 또한, 그림 6을 보면 사용할 수 있는 언어를 선택할 수 있으므로 더욱더 정확하게 시각장애인을 도울 수 있다.

여러 애플리케이션을 조사해 본 결과, 시각장애인들을 위한 좋은 기능들이 많이 있지만, 몇몇 앱들은 버튼이 너무 작아서 시각장애인들이 사용하기 불편하다는 것을 느꼈다. 따라서, 애플리케이션을 설계할 때 시각장애인들의 편의를 위해서 디자인하는 것이 중요하다고 생각하였다.



그림 3. 보이스아이 앱 화면.



그림 4. Be my eyes 앱 첫 화면.

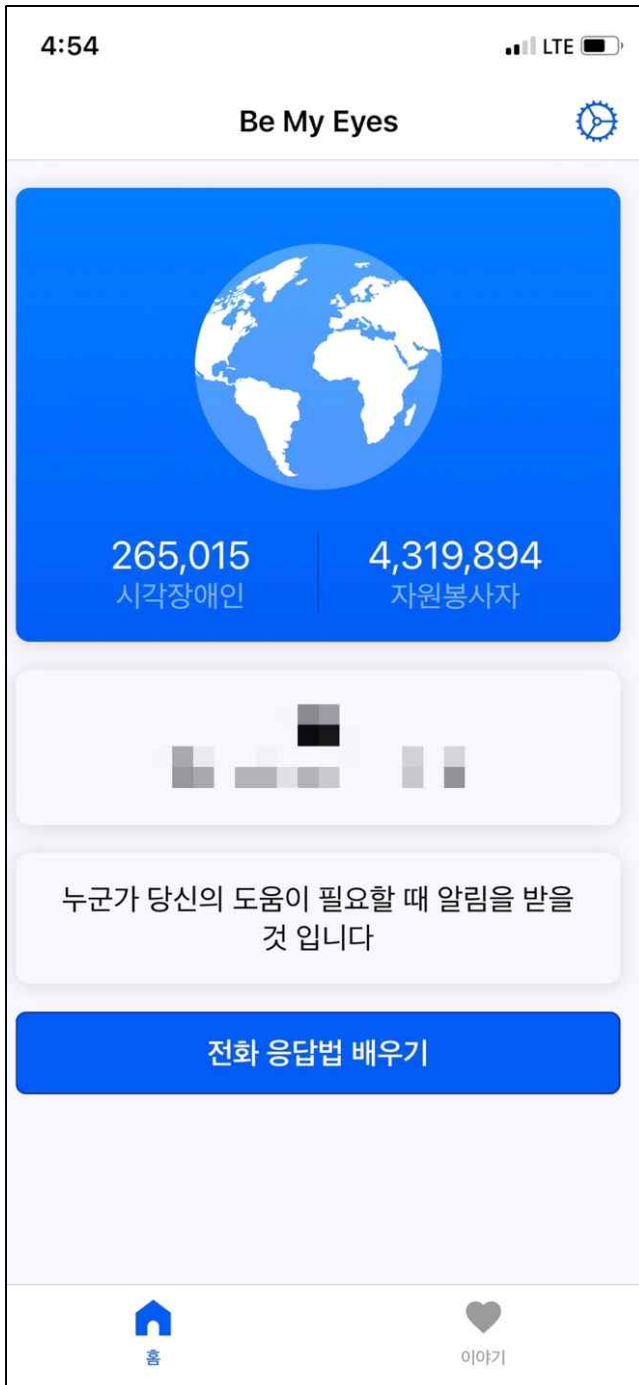


그림 5. Be my eyes 자원봉사자 화면.

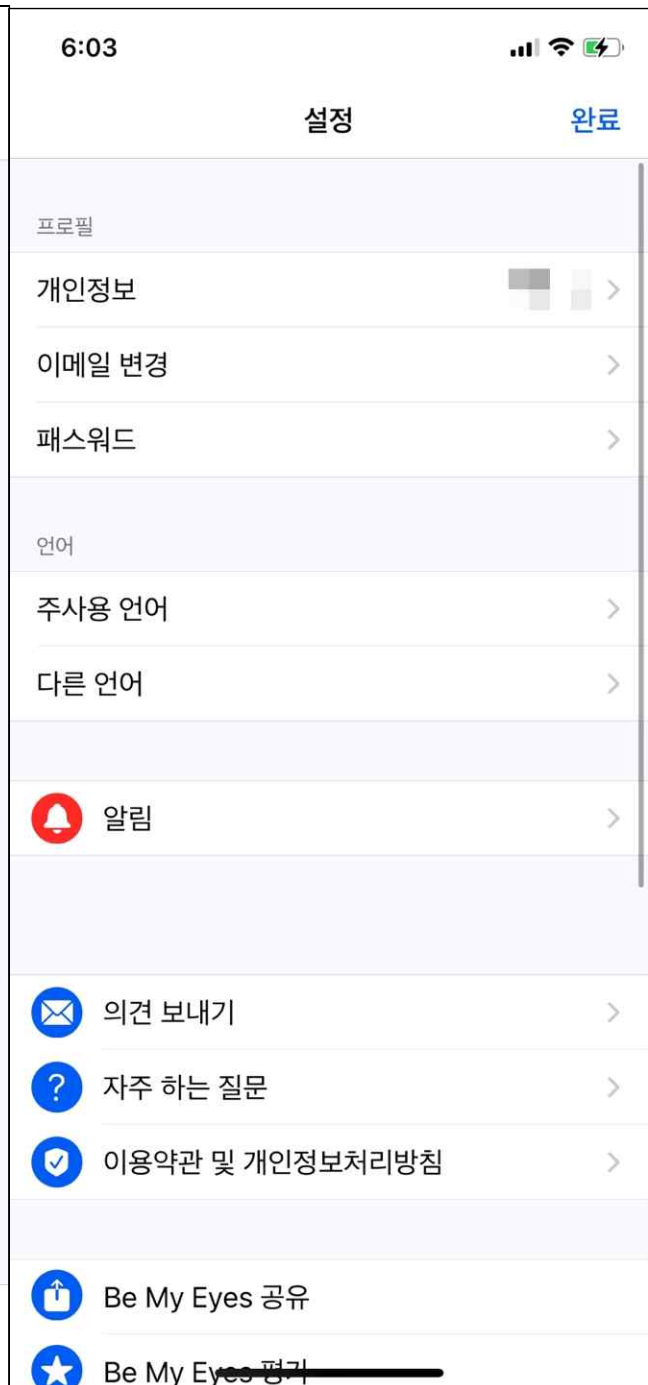


그림 6. Be my eyes 설정 화면.

2. 문헌 조사

시각장애인들을 위한 애플리케이션 개발에 관한 논문을 찾아 정리했다. 스마트폰에서의 시각장애인을 위한 음성 인터페이스-TTS (Text to Speech) 요구사항과 기능에 관한 연구, Open CV와 TTS를 활용한 시각장애인의 키오스크 이용 보조 애플리케이션 설계, 영상 처리 모듈을 이용한 시각보조 시스템의 구현, 시각장애인을 위한 인식 및 보행 도움 안드로이드 애플리케이션에 관련된 논문을 찾아본 뒤, 시각장애인에게 필요한 기능, 현재 출시되어있는 애플리케이션에서 보완할 점에 대해 알아보았다.

3. 애플리케이션 설계를 위한 시스템 설계도 구성

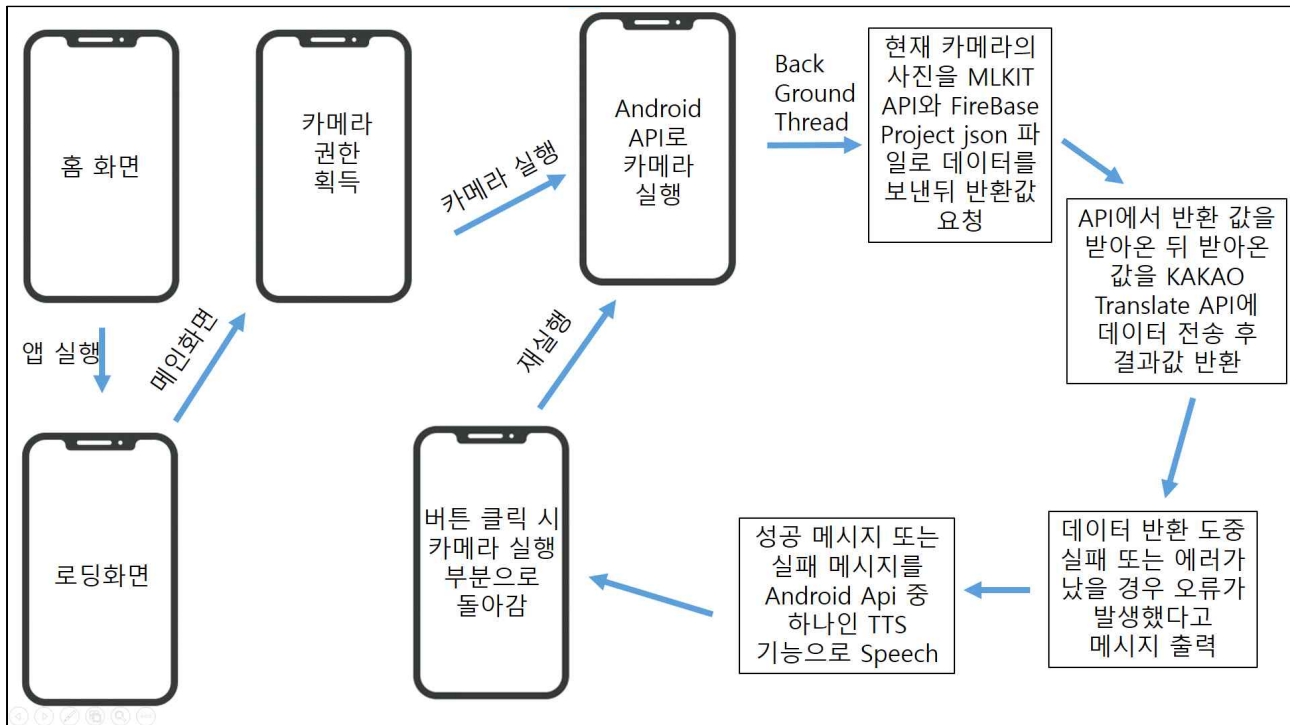


그림 7. 시스템 설계도

조사한 내용을 기반으로 시각장애인을 위한 애플리케이션 설계 요소를 추출하고 다양한 관점에서 시각장애인들의 편의성을 고려한 인공지능 기술 및 사용자 편의성을 고려한 UI를 고안하여, 이를 통해서 시각장애인들이 평소 생활의 불편함을 덜어줄 수 있는 음성 서비스 애플리케이션을 설계할 수 있는 설계도를 구성했다.

첫 번째로, 시각장애인들이 실질적으로 애플리케이션을 사용하기 위해서는 간단한 UI, 즉, UX 디자인이 필요할 것이다. 즉, 사용자가 애플리케이션을 사용할 때, 편리함과 만족감을 고려할 수 있는 애플리케이션 디자인을 고안했다. 시각장애인이 이용하려면 화면의 버튼 위치를 파악하고 있어야 하고, 누를 때 실수가 생길 수 있으므로 버튼의 크기를 크게 해야 했다. 애플리케이션에 들어가게 되면 두 가지 화면으로 나뉘어 있다. 애플리케이션 사용자가 사용할 수 있는 기능은 두 가지가 있는데, 첫 번째로는 위쪽 화면을 누르게 되면 카메라를 실행시킨 뒤에 앞의 물체를 인식하여 그 물체의 정보를 음성으로 제공해주는 기능이다. 두 번째로는 긴급 상황에 보호자에게 전화를 걸고 자신의 위치를 알리는 기능이다.

두 번째로, 시각장애인들에게 필요한 기능에 대해서 고안해보았다. 시각장애인들의 가장 큰 불편 중 하나는, 앞에 있는 정보를 받아들일 수 없는 것과 위급한 상황이나 보호자가 필요한 상황에 연락을 바로 할 수 없다는 점이라고 생각했다. 그 결과, 이 애플리케이션의 기능을 두 가지로 생각했다. 첫 번째 기능은, 카메라를 통해서 앞에 있는 사물의 사진을 API로 전송하여 정보를 받아와 받은 텍스트를 TTS를 이용하여 음성으로 제공하는 것이다.

4. 애플리케이션 개발

안드로이드 스튜디오를 이용해서 애플리케이션 프로토타입을 제작하였다.

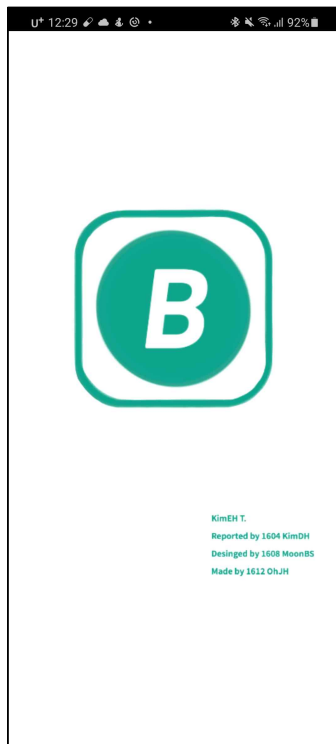


그림 8. 로딩 화면.



그림 9. 테스트 실행화면.

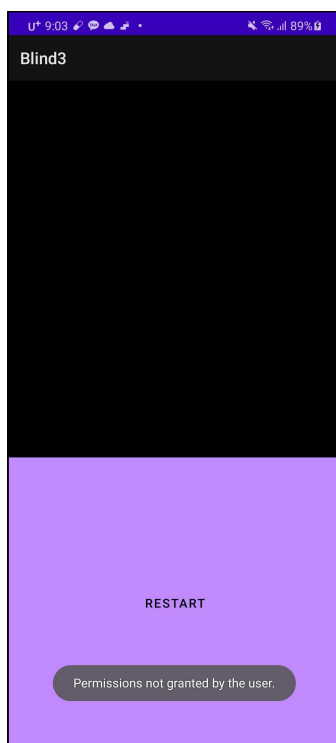


그림 10. 카메라 권한을 부여받지 못한 경우.

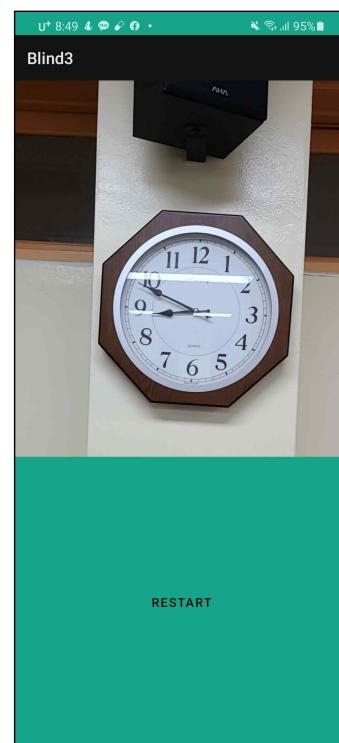


그림 11. 실행화면.

표 1. ML KIT API를 받아오고 정확도를 확인하여 검출

```
private class YourImageAnalyzer(private val listener: MyListener) :
    ImageAnalysis.Analyzer {
    @SuppressWarnings("UnsafeExperimentalUsageError")
    override fun analyze(imageProxy: ImageProxy) {
        val mediaImage = imageProxy.image
        if (mediaImage != null) {
            val image = fromMediaImage(mediaImage,
imageProxy.imageInfo.rotationDegrees)
            // Pass image to an ML Kit Vision API
            val labeler = ImageLabeling.getClient(ImageLabelerOptions.DEFAULT_OPTIONS)
            labeler.process(image)
            .addOnSuccessListener { labels ->
                for (label in labels) {
                    val text = label.text
                    val confidence = Math.round((label.confidence) * 10000)/100
                    if(confidence>70)
                        listener(text)
                }
            }
        }
    }
}
```

표 2. 카메라 미리 보기와 카메라의 사진을 ML KIT로 전송 및 번역 후 TTS

```
private fun startCamera() {
    ... (중략)

    val imageAnalyzer = ImageAnalysis.Builder().build()
    .also {
        it.setAnalyzer(cameraExecutor, YourImageAnalyzer { textall: String ->
            Log.d(TAG, "TEST Image Labeling: $textall")
            ttsGreater21(textall)
        })
    }
    ...
}
```

표 3. TTS function

```
@TargetApi(Build.VERSION_CODES.LOLLIPOP)
private fun ttsGreater21(text: String) {
    var textko: String? = ""
    Thread {
        if(!text.startsWith("//")) {
            val arrtext = text.split("/").toTypedArray()
            textko="앞에 "+translation("There is a "+arrtext[0])
            .replace("있다", "있어요").replace("있습니다", "있어요")
        } else textko = translation(text.replace("//", ""))
        runOnUiThread {
            val utteranceId = this.hashCode().toString() + ""
            mtts.speak(textko, TextToSpeech.QUEUE_FLUSH, null, utteranceId)
            Thread.sleep(2000)
        }
    }.start()
}
```

표 4. ML KIT에 API를 보냈을 시 받아오는 반환 값인 영어를 KAKAO 번역 API를 사용해 한국어로 반환

```
private fun translation(q: String): String {
    //kakao translate
    var jsonString = ""
    var textko = ""
    try {
        jsonString = Jsoup.connect("https://dapi.kakao.com/v2/translation/translate")
            .header("Authorization", "KakaoAK " + key).data("query", q)
            .data("src_lang", "en").data("target_lang", "kr")
            .ignoreContentType(true).post().body().text()
        val jobject = JSONObject(jsonString)
        val translatedarray = jobject.getJSONArray("translated_text")
        val jarray = translatedarray.getJSONArray(0)
        textko = jarray.getString(0)
    } catch (e: IOException) {
        e.printStackTrace()
        textko="번역 도중 오류가 발생하였습니다."
    }
    return textko
}
```

5. 애플리케이션 사용을 위한 보조 장치 제작

우리는 앱의 편의성을 늘리기 위해서 앱을 보조할 도구를 만들어 보기로 생각하였다.

가장 먼저 고안해본 것은 사진처럼 케이스에 블록을 두어서 다음과 같이 만들었다. 구멍을 두어서 시각장애인이 스스로 조립할 때의 편의성을 고려하였다. 또한, 시각장애인이 스마트폰의 위아래와 오른쪽 왼쪽이 헷갈릴 것을 고려해서 왼쪽과 오른쪽에 구멍을 두고 안 두고에 차이를 뒤서 방향을 구분할 수 있도록 하였다. 또한, 그림에는 모든 글이 다 한글로 표기되어있지만, 이를 점자로 표시하여 시각장애인들이 정보를 얻을 수 있도록 하였다.

두 번째로는, 슬라이딩 방식으로 앞선 끼우는 방식을 바꾼 것이다. 끼우는 방식에서 튀어나온 부분이 부러질 수 있다는 생각이 들었고, 이를 보완하기 위해서 만들게 되었다. 하지만 이 또한 끼우는 방식이다 보니 끼우는 것 자체가 어려울 수 있다는 단점이 있다.

세 번째로 생각한 방식은 옆에 구멍을 두는 것이다. 앞서 나온 방식들이 겹으로 튀어나와 있어서 다 칠 수 있고, 끼우는 것이 어렵다는 점을 보완하였다.

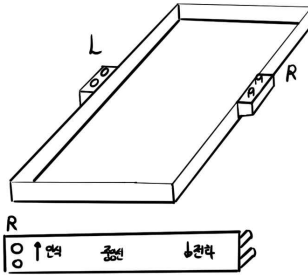
네 번째에서는 끼우는 방식을 보완하여 자석을 통해 붙이는 방식을 생각하였다. 이 방식의 단점은 어쨌면 휴대전화에 좋지 않은 영향을 줄 수 있다는 것이다.

다섯 번째에서는 붙이는 곳을 보완하였고, 여섯 번째에서는 휴대폰케이스를 만드는 것보다 붙이는 방식이 더 나을 것 같다고 생각해서 붙이는 방식으로 보완하려 하였다.

휴대전화에 무언가를 붙이는 방식 중에 요즘 사용하는 휴대폰케이스 중 뒤에 고무줄과 같은 것으로 잡을 수 있는 공간을 제공하는 케이스가 있는데, 이를 응용하여 일곱 번째 방식을 고안했다. 일곱 번째 방법은 고무줄 밴드를 그림과 같이 사용해서 표현하도록 하였다. 또한, 이 위에 점자를 써놓게 된다면 정보 전달까지도 가능하리라 생각했다. 좌우 구분 또한 고무 밴드에 점 하나를 두어서 좌우 구분을 시킬 수 있다고 생각하였다. 위아래의 경우 점자의 위치로 충분히 구분할 수 있으므로 보조 장치 제작을 이와 같은 방식으로 구체화 시키기로 하였다.

결론적으로, 애플리케이션 보조 장치 제작 결과는 휴대폰케이스를 만들기로 하였다. 휴대폰케이스를 통해서 휴대전화의 위, 아래와 왼쪽, 오른쪽을 알 수 있다. 이를 통해서 시각장애인이 휴대전화와 애플리케이션을 이용할 때 방향을 몰라서 생기는 불편함을 해결할 수 있다.

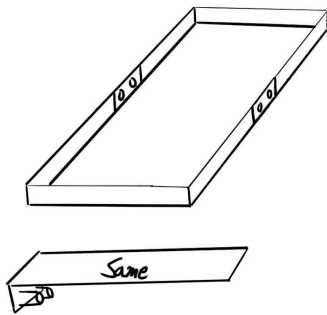
1. Normal



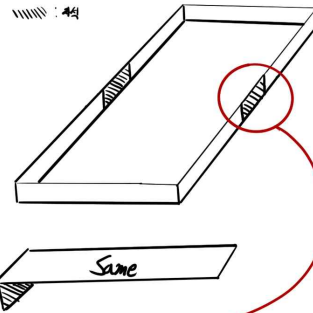
2. Sliding



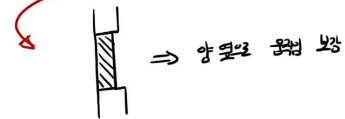
3. Slide



4. mag 1



5. mag 2



6. Sticker

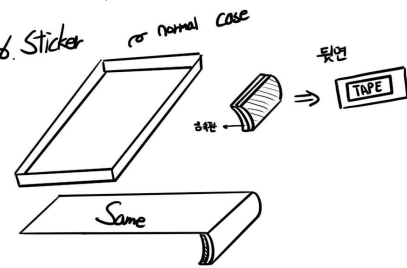


그림 12. 애플리케이션 보조 장치 제작 과정 1~6.

7. BAND

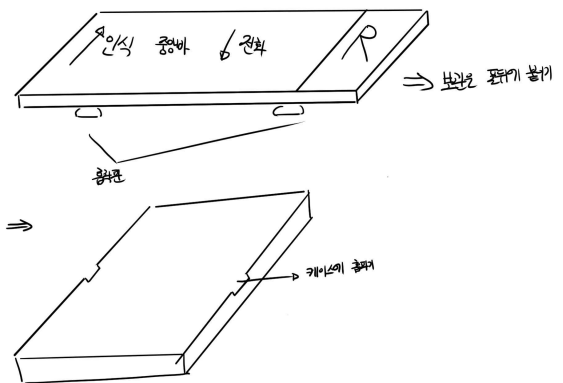
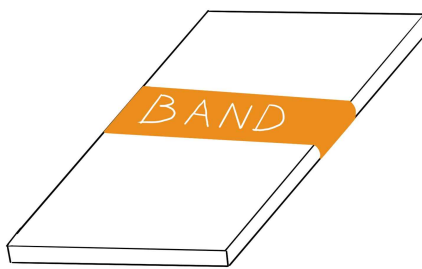


그림 13. 애플리케이션 보조 장치 제작 과정 7. 그림 14. 애플리케이션 보조 장치 제작 과정 8.



그림 15. 애플리케이션 보조 장치 제작 결과.

IV. 연구 결과

사전 조사를 통해서 보이스아이, 샤인플러스 등 여러 애플리케이션의 종류와 사용 방법을 알아봄으로써 시각장애인을 위한 앱에서 필요한 기능이 무엇인지, 어떤 점이 불편한지에 대해서 알아본 후 애플리케이션 설계 시 필요한 점들과 불필요한 점들을 탐색하였다. 모든 앱이 버튼을 기반으로 하는 UI를 지니고 있어서 옆에서 대신 버튼을 눌러줄 만한 사람이 없다면 시각장애인이 스스로 할 수 있어야 한다는 취지에 맞지 않기에 적합하지 않다고 생각했다. 그 결과, 시각장애인들의 편의를 위해서는 시각장애인들이 화면을 볼 수 없는 점을 고려해서 버튼의 개수를 최소화하고, 버튼의 위치를 촉각으로 알 수 있도록 해야 하고, 시각장애인이 스스로 할 수 있는 것을 조금이라도 눌러줄 수 있도록 앞의 정보를 전달해줄 기능과 위험 상황에 대비해서 보호자와 빠르게 연락이 가능한 기능이 필요하다는 결론이 나왔다.

연구 주제는 시각장애인을 위한 애플리케이션 설계이다. 주제를 이뤄내기 위해서 실제로 코드를 작성해서 간단한 앱 또한 만들어 보았다. 우선 앱 만드는 것을 진행하기에 앞서 “Oven”이라는 앱 디자인 웹을 통해서 앱의 기본적인 모양을 만들어 보았었는데 이와 최대한 유사하게 만들기 위해서 노력을 하였다. 하지만 중간에 개발 과정에서의 어려움으로 계획했던 부분이 변경되었지만, 최대한 유사하게 만들기 위해서 노력하였다. 우선 우리가 초기에 설정한 시뮬레이터와의 차이점은 로그인 창이 없어진 것이다. 로그인 창과 이를 받아들이는 서버를 만들지 않은 이유는 굳이 이 앱을 사용하면서 계정을 만들어서 관리하는 것의 필요성이 없을 것으로 생각했기 때문이다. 또한, 우리가 먼저 생각했던 소셜 등과 연동 또한 하지 않았다. 하지만 우리가 가장 중요하게 생각한 UI/UX 부분은 가장 비슷하게 만들도록 하였다. 우리가 만든 앱의 결과물을 보면 가장 먼저 “BLIND”라는 앱을 누르면 정말 간단하게 1개의 버튼만이 존재하게 된다. 위의 버튼을 누르게 되면 자신의 앞에 있는 물체들을 읽어주게 된다. 개발 초기에는 두 개의 버튼으로 기능을 나누려고 하였지만, 개발 과정에 있어서 어려움이 발생해 한 개의 버튼으로만 하기로 하였다. 애플리케이션의 디자인적인 부분을 보면 블라인드라는 앱의 시작화면은 우리의 앱의 아이콘을 두었고, 우리의 앱의 보조도구인 케이스에는 뒷면에 블라인드에 대한 디자인을 점자와 결합해서 만들게 되었다.

초기에는 앱의 기능을 두 가지로 만들기로 계획하였다. 첫 번째는 현재 앱의 기능인 물체의 정보를 음성으로 제공하는 것이고, 두 번째는 버튼을 누르면 바로 보호자에게 연락할 수 있는 기능이다. 시각장애인이 긴급한 상황에 놓였을 때 꼭 필요한 기능이라고 생각되어 만들기로 하였으나, 애플의 시리, 삼성의 빅스비 등에 이미 있는 기능이므로 앱에 기능이 없더라도 충분히 이용할 수 있다고 생각하여 첫 번째 기능만 사용하기로 하였다.

따라서 앱을 실행시키면 로딩 화면 후 카메라가 실행된다. 시각장애인이 알고 싶은 물체의 방향으로 카메라를 가져다 대면 앞에 무슨 물체가 있는지 음성으로 알려준다. 앱을 실행시키고 로고가 나온 뒤 Main Activity가 호출되는데, Main Activity가 호출되는 동시에 물체를 인식하는 API가 실행되면서 음성으로 정보를 제공하는데, 처음 한 번만 실행되므로, 아래의 버튼을 누르면 Main Activity가 다시 호출되어 반복해서 기능을 사용할 수 있도록 하였다.

V. 결론 및 고찰

다음과 같이 시각장애인을 위한 애플리케이션을 제작하였다. BLIND 앱을 통해서 시각장애인이 얻을 수 없는 시각 정보를 청각 정보로 바꿔서 제공할 수 있다. 시각장애인이 앞에 있는 물체에 대한 정보를 알고 싶을 때, 앱을 실행시키면 카메라가 실행되고, 카메라를 원하는 방향을 가리키게 하면, 앞에 무엇이 있는지 알려주게 된다. 이를 통해서 시각장애인이 조금 더 스스로 생활할 수 있도록 기회를 줄 수 있을 것이다. 시각장애인에겐 보호자가 꼭 필요하지만, 보호자가 잠깐 도와줄 수 없는 상황에 놓인다면 이 애플리케이션이 시각장애인에겐 큰 도움이 될 것이다.

많은 앱의 문제점이라면 비장애인들이 만들기 때문에 아무리 시각장애인에게 편한 앱을 만든다고 하더라도 불편한 점이 생길 수밖에 없다. 현재 앱의 UI는 버튼의 크기가 크고, 개수가 적어서 시각장애인이 충분히 기억하고 사용할 수 있지만, 이도 불편한 점이 존재하기에, 장애인용 UI를 개발하는 것이 절실히 필요함을 느꼈다. 따라서 추후 시각장애인이 편리하게 사용할 수 있는 애플리케이션 UI에 관한 연구도 가능할 것이다.

추가로 사용자의 편의를 위해 Blind 앱을 통해 API를 연동하여 Bixby, Siri, google assistant 등의 인공지능 기반의 프로그램 등을 연동해보고 싶었으나 시간도 부족했고, API 지원을 하려면 돈을 내거나 하는 경우가 많아 못했다. 현재 시각장애인을 위한 애플리케이션이지만, 기능이 부족하다고 생각한다. 이후에 더 많은 연구를 통해서 시각장애인을 위한 더 많은 기능에 대해서 고민해보고, 이를 구현시킴에 따라 더 나은 앱을 개발하고, 시각장애인이 BLIND 앱을 가지고 스스로 할 수 있는 것의 범위를 조금 더 늘림으로써 시각장애인의 불편함을 해소하여 시각장애인의 삶의 질을 조금 더 높일 수 있을 것이다.

디자인하는 중 생각해서 만들어낸 부분인데, 휴대전화만 있으면 시각장애인인 주 사용자가 어디에 무슨 버튼이 있는지 잘 판단하지 못하거나 휴대전화가 뒤집힌 것 등을 잘 판단하기 힘들어서 점자를 넣어 상하를 구분할 수 있도록 케이스를 만들었다. 그러나 테스트 핸드폰이었던 Galaxy S20의 3D Printer 도면이 없거나 유료였던 관계로 팀원의 핸드폰인 iPhone 6, 6S의 도면을 전제로 하여 iPhone의 케이스를 만들었다.

아쉬운 점은 첫 번째로 바로 위에 있는 케이스 부분에서, 테스트 핸드폰에 케이스를 이용해서 앱을 실행시켜보지 못했던 점이다. iPhone은 IOS 기반으로, Android 기반이 아니므로 Android Studio에서 만든 앱을 사용하지 못해 케이스와 앱을 같이 사용해 보지 못해 아쉬웠다. 두 번째로 앱이 사용하는 API인 GOOGLE의 ML KIT API의 정확도가 예상보다 많이 떨어진다는 것이었다. 가능하다면 머신러닝을 통해 직접 데이터값들을 입력해서 좀 더 일상적인 곳에 편리할 수 있도록 정확도 및 일상생활에 필요한 단어들만 검출해 내는 것을 해보고 싶다. 기존에 사용하려 했던 KAKAO의 API인 MULTI TAG 기능은 상품 검출 같은 것을 잘 해내는 편인 것 같아 일상생활에 유용할 것 같다고 생각하였으나, 변수값인 image_url String 값과 image Binary 값 중 String인 image_url 값은 잘 넘길 수 있지만, Binary인 image 값은 넘기는데 코드를 어떻게 짜야 할지 몰라 여러 번 찾아보았지만, 애초에 KAKAO의 API Vision이 Beta 버전이며, API의 예제코드들이 Python으로 작성되어 있어 안드로이드 개발에 유용하지 못했다. 시간이 된다면 조금 더 연구해봐서 KAKAO API도 사용해 보고, ML KIT와의 정확도 등을 직접 비교해 볼 수 있으면 좋겠다. 마지막으로 연구 거의 마지막에 JAVA로 구현하기 힘들어 JAVA 언어에서 Kotlin언어로 급하게 바뀌 마지막에 코드를 짜는 것이 시간이 촉박했기에, 더 해볼 수 있을 것 같음에도 하지 못했던 것이 있어 아쉬움이 많이 남는다.

VI. 참고 문헌

- [1] 이선화, 박예경, 김진실. (2019). 스마트폰에서의 시각장애인을 위한 음성 인터페이스-TTS (Text to Speech) 요구사항과 기능에 대한 연구. 한국HCI학회 학술대회, (), 1243-1245.
- [2] 신현우, 이재용, 채지환, 이광재. (2018). OpenCV와 TTS를 활용한 시각장애인의 키오스크 이용 보조 애플리케이션 설계. 대한전기학회 학술대회 논문집, (), 27-32.
- [3] 김예진, 임창훈. (2018). OCR과 TTS, 영상 처리 모듈을 이용한 시각보조 시스템의 구현. 대한전자 공학회 학술대회, (), 579-582.
- [4] 손예지, 서혜인, 이수정, 하동휘, 김형신. (2019). 시각 장애인을 위한 인식 및 보행 도움 안드로이드 애플리케이션. 한국정보과학회 학술발표논문집, (), 1321-1323.
- [5] Wikipeda: 음성합성. _음성합성, 광학 문자 인식, 음성 인식,. <https://ko.wikipedia.org/>
- [6] Red Hat. API(애플리케이션 프로그래밍 인터페이스)란 - 개념, 기능, 장점
- [7] 장애인을 위한 앱 31가지 <https://1boon.kakao.com/bloter/307031>
- [8] UI/UX 디자인 뜻, 차이점은 무엇일까? [http://blog.wishket.com/uiux-디자인-뜻-차이점은-무엇](http://blog.wishket.com/uiux-디자인-뜻-차이점은-무엇일까)
[일까](http://blog.wishket.com/uiux-디자인-뜻-차이점은-무엇일까)