시각장애인을 위한 길 안내 서비스

2022.12.12.

김겸호(B671003)

이휘경(B711159)

금예인(B911012)

1. 서론

자유롭게 이동할 수 있는 권리를 뜻하는 이동권, 누구나 누려야 하는 기본권이지만 교통약자인 장애인들에게는 쉽게 주어지지 않는다. 특히 시각장애인들에게 처음 가보는 길을 찾고, 수많은 장애물을 피해 대중교통을 타고 이동하는 일은 더더욱 어렵다. 보도에는 규격이 다른 볼라드가 설치되어 시각장애인 뿐만 아니라 비장애인 역시 길을 걷다 상해를 입기도 하고, 점자블록 위에는 차량이 불법 주차되어 있기도 한다. 코로나 바이러스가 확산되기 전인 2019년 10월에는 시각장애인이 길을 걸을 때 사용하는 도구이자 시각장애인의 자립과 성취를 상징하는 '흰지팡이의 날'을 기념하여 시각장애인과 봉사자 150명이 이동권 보장을 요구하며 행진하기도 했다.

단순히 보행만 문제가 아니다. 대중교통 이용 역시 큰 벽이다. 2019년, 시각장애인이 버스를 탈 수 있을까? 라는 내용으로 사회실험이 진행되었다. 서울 시내 중앙차로에 위치한 버스정류장에서 시각장애인이 홀로 버스를 타는 실험이었다. 정류장에 설치된 버스 번호 음성안내 시스템의 도움을 받아 승차하려 했으나 안내 음성은 잘 들리지 않았고, 잠시 후 도착한다는 안내를 듣고 버스를 타려 했을 때는 이미 버스가 출발하고 없었다. 이런 어려움을 겪는 시각장애인에게 조금이라도 도움을 주고자 시각장애인을 위한 길 안내 서비스 앱을 제작하고자 한다.

2. 시각장애인의 이동 및 버스 관련 실태

2.1 위험한 도로 보행

시각장애인의 도보 이동이 안전하지 못하다. 지난해 국민권익위원회가 발표한 자료에 따르면,

2018년~2020년까지 민원분석시스템으로 수집된 '점자블록' 관련 민원이 총 2847건이었다. 이는 이전 3년 동안 접수된 1672건의 약 1.7배에 달하는 수치로, 44.2%가 파손되거나 훼손된 점자블록에 대한 민원이었으며, 21.2%는 버스정류장과 전봇대 등 점자블록 위에 설치된 다른 시설물에 대한 민원이었다. 더불어 점자블록이 잘못 설치되거나 미설치된 구역에 관한 민원이 무려 32.3%에 달했다. 거리를 차지한 오토바이와 자전거, 법에 맞지 않는 볼라드뿐만 아니라 시각장애인의 눈이 되어주어야 할 점자블록마저도 오히려 안전을 위협한다는 지적이 나오는 현실이다.

2.2 교통 대안의 부재

버스 이외에 시각장애인이 편리하게 이용할 만한 교통 수단이 미흡하다. 지하철은 노선이 하나이기에 열차번호를 확인할 필요가 없고, 승차 구역이 정해져 있어 여러 면에서 시각장애인이 이용하기에 편리하지만 버스보다 그 수가 현저히 적으며 역이 집과 목적지에서 멀리 떨어져 있는 경우가 많다. 더불어 수도권과 인구수가 많은 도시를 제외하고는 도시 철도가 운행되지 않는다. 다양한 이동지원센터 역시 명확한 해결책이 되지 못한다. 서울시의 바우처 택시처럼 각 지자체에서 이동지원센터를 운영하고 있으나 대부분 비용이 청구되며, 이마저도 원활히 운영되지 않는 경우가 부지기수이다. 특히 시각장애인 복지콜 서비스 현황 자료에 따르면 최근 3년간 전체 운행 대수가 158대로 단 1대도 증차되지 못하고 유지되었고, 코로나19로 이동량 자체가 감소한 2021년에도 평균 37.9분을 기다려야 탑승할 수 있었다. 버스 이용이 조금만 더 개선된다면, 시각장애인들에게 더없이 좋은 대중교통 수단이 될 것이다.

2.3 현재 제공되는 대중교통 길 찾기 서비스

정류장까지의 경로와 탑승해야 할 버스 정보 등을 제공하는 다양한 길 찾기 서비스가 제공되고 있으나, 대부분 시각 장애가 없는 비장애인들을 대상으로 운영되고 있다. 국내의 주요 길 찾기 서비스 앱은 국제 거리 단위인 미터(m)로 경로를 설명하는데, 시각 장애를 가지고 있는 경우 미터(m)에 대한 거리 감각이 부족하여 어느 정도를 가야 하는지 알기 힘들다. 또한 작은 글씨와 다양한 색상 등 전체적인 디자인 역시 저시력, 색약과 색맹 등을 고려했다고 보기 어렵다.

2.4 정류장에서 제공되는 버스 정보

정류장의 버스 도착 정보 안내판은 버스가 현재 정류장에 도착하기까지 남은 시간을 시각적으로 제공한다. 잠시 후에 도착하는 버스 정보만 음성안내를 통해 제공하며, 이 음성 서비스조차도 안내판 바로 밑이 아니면 들리지 않을 정도의 크기이다. 실제로 국토교통부의 2020년 교통약자 이동편의 실태조사 연구에서 버스정류장의 '안내판 점자 및 음성안내'에 대해 기준적합 판정을 받은 비율은 11%에 불과했다. 심지어 41.7%는 미설치 판정을 받았다. 이렇듯 시각장애인은 타야 하는 버스가 언제 도착하는지 파악하기 어렵고, 해당 버스가 도착한 후에도 한 번에 여러 노선이 도착한다면 어느 위치에서 승차해야 할지 알 수 없다.

3. 프로젝트 목표

3.1 장애물 인식 서비스 - 안전한 보행

시각장애인의 안전한 보행을 위해 스마트폰에 내장된 카메라로 장애물을 인식할 수 있도록 개발한다. 구글 이미지를 크롤링하여 데이터를 수집하고, 이에 라벨링 작업을 거쳐 데이터셋을 마련한다. 이를 기반으로 TensorFlow Lite 모델을 학습시키고, 안드로이드 어플리케이션과 연결하여 전방의 장애물을 음성으로 안내한다.

3.2 버스 및 경로 안내 서비스 – 보폭 이용

미터(m) 단위가 익숙하지 않은 시각장애인들을 위해 보폭을 통해 길을 안내한다. 앱 실행 시 GPS를 이용하여 걸은 거리 정보를 얻고, 스마트폰의 내장 센서로 걸은 걸음 수 정보를 얻어 거리/걸음 수를 통하여 이용자의 보폭을 구한다. T-MAP API에서 제공하는 미터(m) 단위의 경로를 이용자의 보폭으로 나누어 앞으로 몇 걸음을 걸어야 하는지 음성으로 안내한다. 길 안내는 직진, 좌회전, 우회전 등 방향에 대한 설명과 횡단보도 정보를 포함한다.

3.3 버스 및 경로 안내 서비스 - 보다 적합한 버스 정보 제공

국토교통부에서 제공하는 버스위치정보 Open API를 이용해 정류장에서 버스를 기다리는 시각장애인에게 타야 하는 버스의 현재 위치와 도착 예정 시각 등을 음성으로 안내한다. 이때, 도로

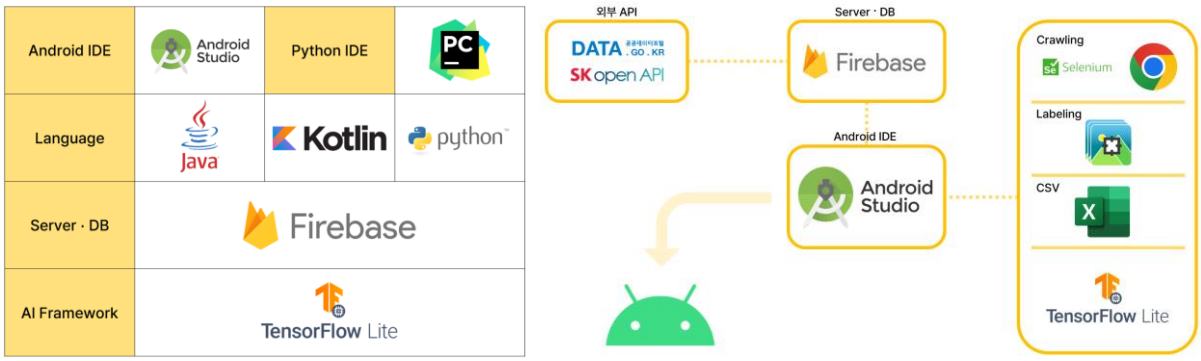
교통 상황이 변동될 것을 고려하여 주기적으로 새로운 데이터를 받아온다. 또한 버스 도착에 관하여 여러 버스가 한 번에 정류장으로 들어올 경우, 카메라로 노선 번호를 인식하여 승차할 버스를 찾는 방법은 버스가 서 있는 간격과 카메라 각도 등을 고려했을 때 현실적으로 무리라고 판단되었다. 따라서 버스 기사 회원가입을 도입하여 가입 시 운행하는 버스 노선 번호를 입력 받고, 앱 이용자가 버스 정류장에 도착하면 해당 버스를 운행하는 버스 기사에게 시각장애인이 정류장에서 대기하고 있음을 푸시 알림을 통해 알린다.

3.4 편리한 이용

많은 사람들이 쉽고 편리하게 이용할 수 있도록 앱을 구성한다. 노란색은 파장이 길고 주목도가 높아 저시력자와 시야각이 좁은 사람들도 구분할 수 있다. 이를 고려하여 전체적인 앱 디자인에 검은색 배경에 노란색 버튼을 사용하고, 크고 단순하게 디자인한다.

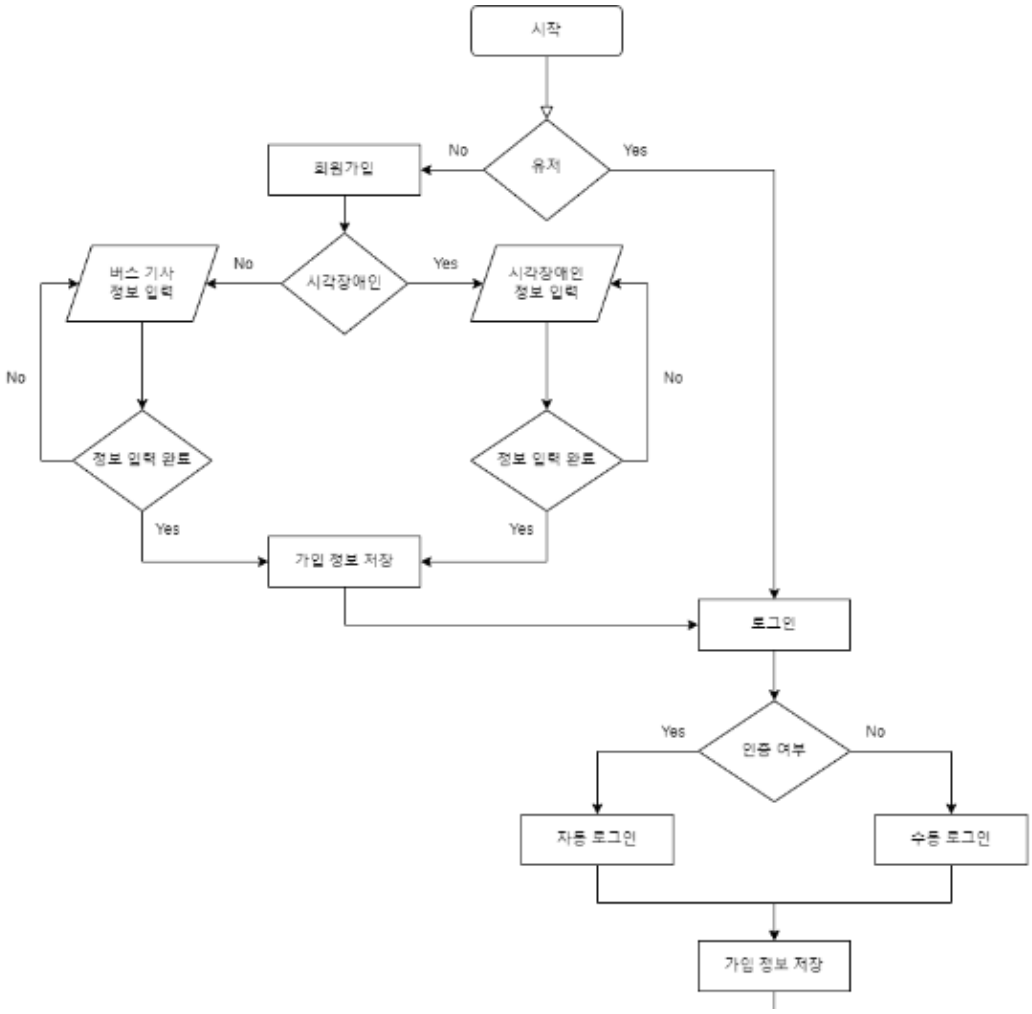
또한 목적지까지의 경로, 이용자의 정류장 도착 여부, 버스 정보를 AI 음성을 통해 적절히 제공하고 음성 인식을 통한 빠른 검색을 지원한다.

4. 개발 환경

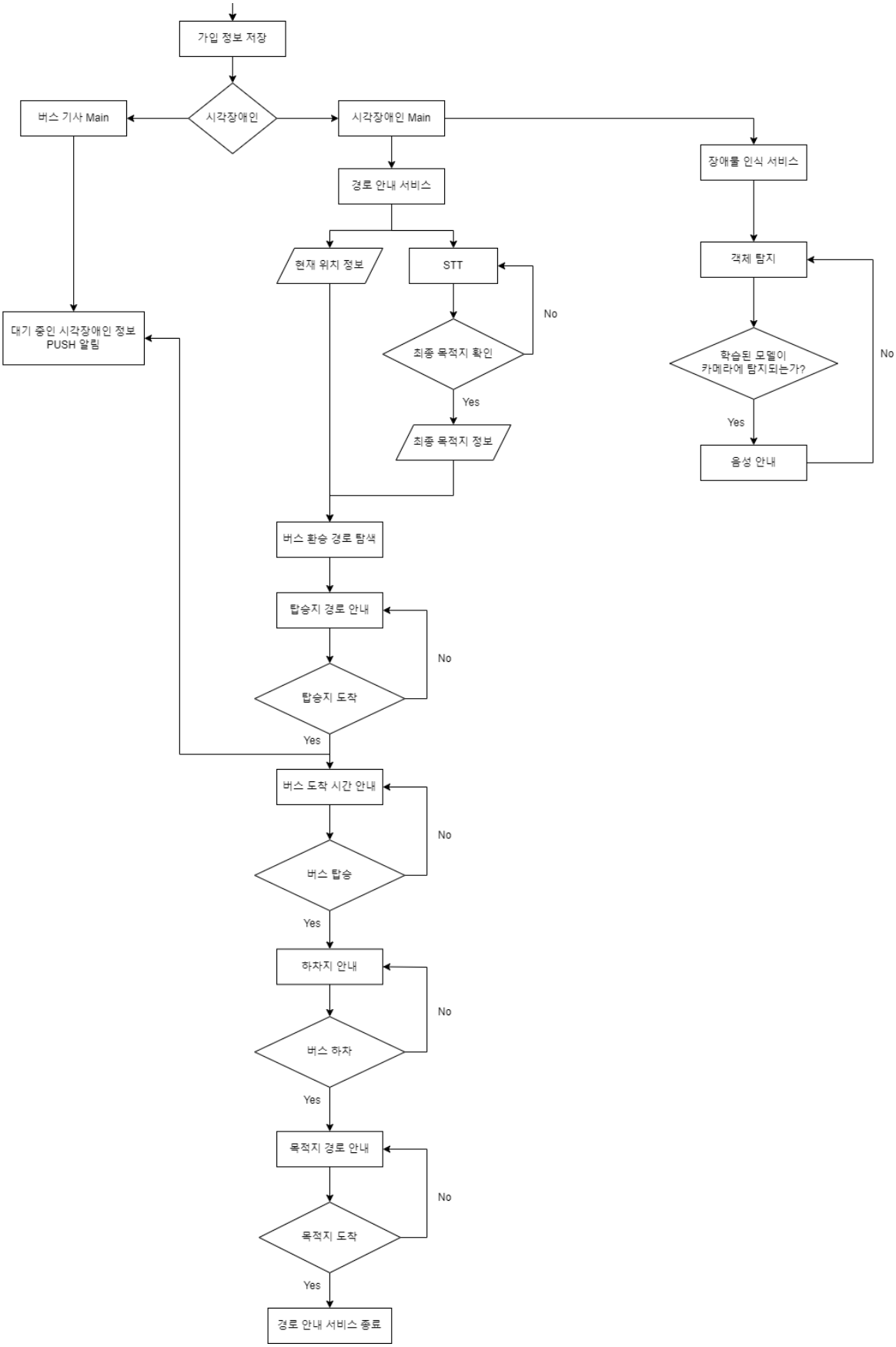


카메라를 통한 장애물 인식 및 인공지능 학습 모델과 어플리케이션 간의 원활한 연결을 위해 Android Studio를 사용하여 제작한다. 서버 및 버스 기사와 사용자의 데이터베이스 관리를 위해 Google의 Firebase를 사용한다. 또한 TensorFlow Lite를 통해 TensorFlow 기반 객체 탐지 모델을 만들어 장애물 안내 서비스를 구현한다. 모든 작업은 git, GitHub 등 원격 저장소를 통해 협업한다.

5. 프로그램 개요



[회원가입 및 로그인]


가입 정보 저장
시각장애인
버스 기사 Main
시각장애인 Main
장애물 인식 서비스
경로 안내 서비스
현재 위치 정보
STT
객체 탐지
대기 중인 시각장애인 정보
PUSH 알림
No
최종 목적지 확인
학습된 모델이
카메라에 탐지되는가?
No
Yes
Yes
최종 목적지 정보
음성 안내
버스 환승 경로 탐색
탑승지 경로 안내
No
탑승지 도착
Yes
버스 도착 시간 안내
No
버스 탑승
Yes
하차지 안내
No
버스 하차
Yes
목적지 경로 안내
No
목적지 도착
Yes
경로 안내 서비스 종료

[주요 서비스]

6. 프로그램 기능

6.1 회원가입·로그인 및 버스 도착 알림을 위한 Firebase 활용

Google에서 제공해주는 클라우드 서버인 Firebase를 활용하여 간단한 회원가입과 로그인 기능을 구현하고 푸시 알림 시 필요한 정보를 수집하기 위해 사용자의 정보를 입력 받아 데이터 베이스에 저장하였다. 운전자에게서 받아야 할 정보와 시각 장애인 분께 받아야 할 정보가 다르므로 각각 다른 화면에서 정보를 입력하도록 구성하였다. 또한, 운전자의 정보를 입력 받을 때 Firebase Cloud Messaging(FCM)을 위해 필요한 FCM 토큰을 생성하여 함께 저장한다.

[회원가입 및 로그인]

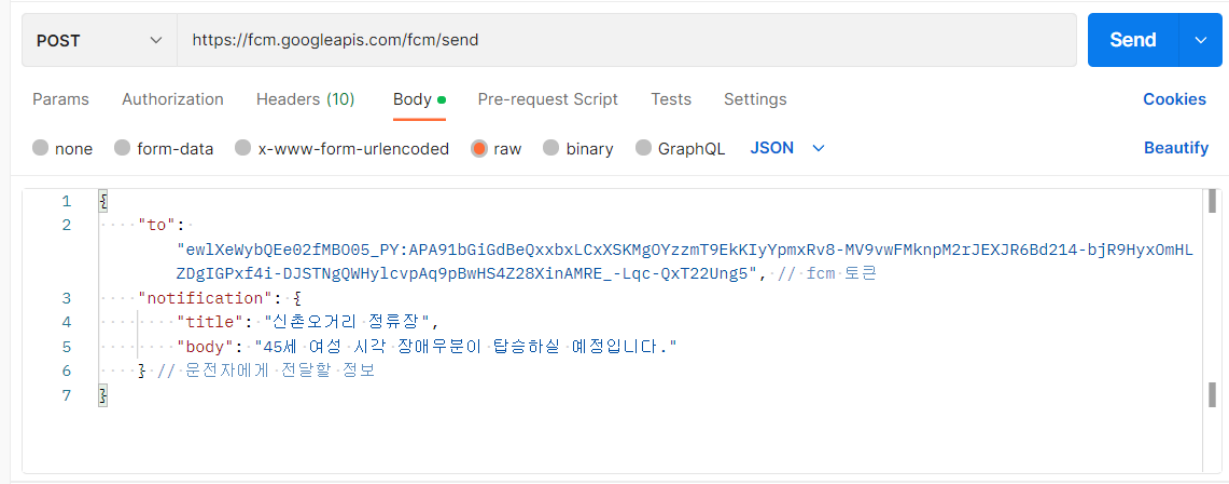
6.2 버스 기사에게 푸시 알림

Firebase Cloud Messaging(FCM)을 이용하여 버스 기사에게 '어느 정류장에 OO대 남자/여자 시각장애인이 버스를 타기 위해서 대기 중이다'라는 알림을 보낸다.

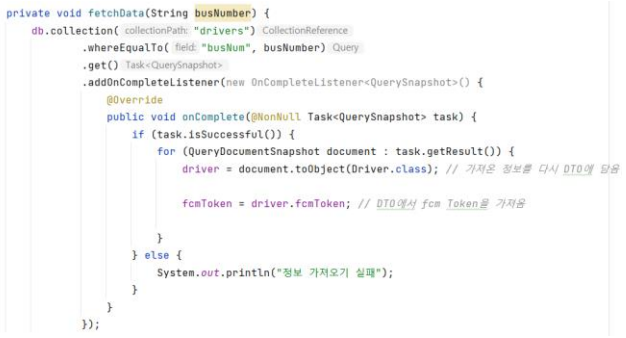
이것을 위해 데이터 베이스에서 버스 번호를 통해 FCM 토큰을 가져온 후 이를 이용해

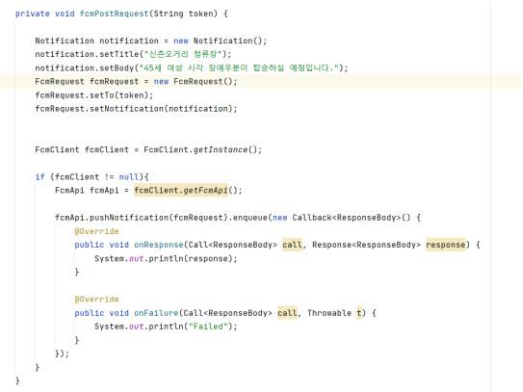
Https://fcm.googleapis.com/fcm/send 엔드 포인트에 HTTP POST 요청을 한다. 요청을 할 때 Json 포맷으로 요청을 하게 되는데, to에는 FCM 토큰을 넣고 Notifications의 title과 body에 보내고자 하는 메시지의 정보를 입력한다. title에는 시각장애인이 기다리고 있는 정류장의 정보를 입력하고 body에는 사용자의 성별과 나이 정보를 입력한다. 시각장애인의 정보는 현재 사용자의 인증 객체의 uid를 외래키로 활용해 DB에 있는 회원정보를 가져온다.

서울특별시_버스도착정보조회_서비스에서 도착 예정 버스의 번호판을 받아올 수 있고 이를 통해 데이터 베이스에 저장된 버스 번호판 번호와 일치하는 버스기사의 기기 토큰에 알림을 보낼 수 있게 된다.



[Firebase Cloud Messaging(FCM) POST 요청]

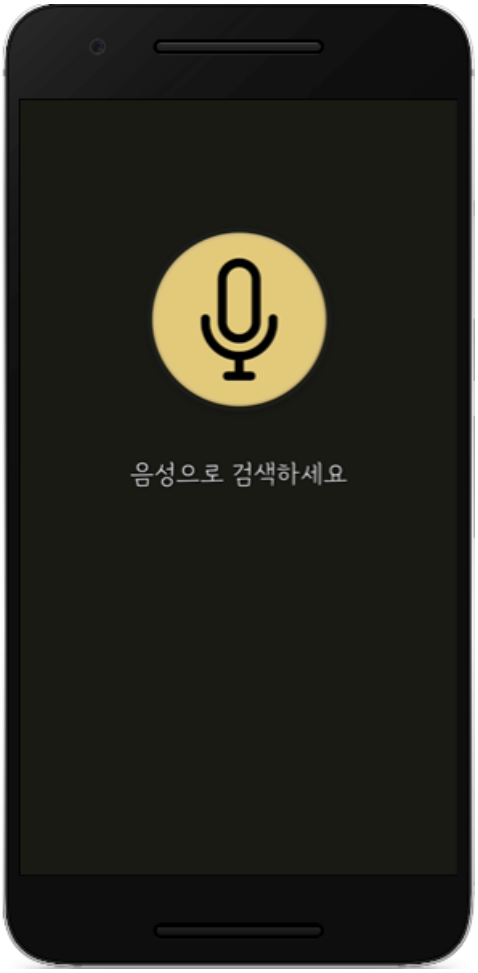




[DB에서 버스번호를 통해 FCM 토큰을 가져옴] **[HTTP 라이브러리인 Retrofit으로 POST 요청]**

6.3 목적지 검색

Android Studio에서 제공해주는 Speech-to-Text 라이브러리를 사용하여 음성인식을 구현하였다. Text-to-Speech를 사용하여 STT 기능을 이용할 수 있도록 화면 상단을 클릭하라는 안내를 하였다. 안내에 따라 화면 상단의 마이크 버튼을 클릭하면 버튼을 눌렀을 때의 소리가 나온다. 소리가 나온 후에 목적지를 말하면 음성을 텍스트로 변환한다. 음성인식의 부정확함을 고려하여 검색시에 검색하는 텍스트를 TTS로 말해줘 어떤 검색어로 검색이 되는지 사용자가 알 수 있게 하였다. 예를 들어 '연세대학교'로 검색하면 TTS로 '연세대학교로 검색합니다'를 말하게 되어 어떻게 인식하였는지 사용자에게 전하며 다시금 확인할 수 있도록 구현하였다.



[STT 검색 전]

변환된 텍스트는 국토교통부에서 제공하는 검색 API를 호출하는데 사용되며 사용자가 정확한 장소의 명칭을 모를 경우를 고려하여 텍스트와 관련된 장소들이 리스트로 뜰 수 있도록 구현하였

다. 예를 들면 개봉초등학교를 검색하였을 때 개봉초등학교와 관련된 장소들, 서울개봉초등학교 , 서울 개봉초등학교 병설유치원 등이 리스트에 뜨게 된다. 리스트를 한 번 클릭하면 목적지의 이름을 TTS를 이용하여 말하여 주고 원하는 목적지일 경우 더블 클릭하면 목적지가 설정되며 해당하는 목적지의 GPS 값을 가져온다.

[STT를 이용하여 검색 후]

6.4 버스 경로 선택

목적지까지 가기 위해 탑승해야 하는 버스 정보를 얻기 위해서 공공데이터 포털에서 제공하는 서울특별시_대중교통환승경로 조회 API를 사용하였다.

앞서 6.3에서 가져온 목적지의 GPS 값과 검색할 때의 현재 위치 정보를 사용하여 API를 호출하게 된다.

받아온 response를 transferlist라는 객체에 담아 사용하게 된다. transferlist 안에는 각 객체마다 Pathlist를 갖게 되고 Pathlist 안에는 탑승지 좌표(Fx, Fy), 탑승지 이름 (fname), 탑승 정류장의 고유 아이디(fid), 하차지 좌표 (Tx, Ty), 하차지 이름(tname), 타야 하는 버스(BusNm), 버스를 타고 이동하는 시간(time)이 담겨 있다. 이 transferlist를 인덱스 별로 View에 목적지까지 가기 위해 현재 타야 하는 버스, 현재 탑승해야 할 버스정류장, 소요시간, 앞으로 남은 버스 탑승 횟수를 띄워 목적지까지 가기 위한 방법을 선택할 수 있게 하였다. 너무 많은 버스 이용을 하는 경로는 어려움을 느낄것이라 생각되어 탑승 횟수가 3회이상 즉 환승을 2번 이상하게 되는 경로들의 경우에는 리스트뷰에 띄우지 않게 설정을 하였다. 만약 6.3의 좌표로 가는 경로가 모두 3회 이상인 경우에는 경로가 없습니다 다시 검색해주세요란 안내 음성을 한 후에 검색을 할 수 있는 엑티비티로 다시 이동하게 해두었다.

경로가 존재하는 경우에는 사용자가 한 번 클릭하면 뷰에 담긴 정보 즉, 버스 번호 , 타야하는 정류소 , 예상 소요 시간, 버스 탑승회수를 TTS로 안내를 하게 되며 두 번 클릭하면 선택된 경로 정보를 담은 transferlist를 다음 액티비티에 넘겨 보행 경로 안내 액티비티에서 탑승지까지 도보 안내를 시작할 수 있게 된다.

[버스 경로 선택]

6.5 버스 도착 정보 및 하차지 안내

버스 도착 정보를 안내하기 위해서 서울특별시_정류소정보조회 서비스내에 정류소 명칭 검색 API와 정류소 버스 도착 정보목록을 엮어서 사용하였다.

우선 정류소 명칭 검색 API는 6.2에서 선택된 TransferIlst에서 fname을 사용하여 호출하게 된다. 예를 들어 홍대정문이라는 버스정류장 이름을 가지고 호출하게 되면 같은 이름을 가진 여러 개의 정류장 정보를 얻게 된다. 그 중에서 하나를 특정하기 위해 TransferIlst에서 fid(정류장 고유아이디)를 사용한다.

fid와 Response에서 일치하는 Stid(정류장 고유번호)를 가진 정류소의 arsId(정류장 번호)를 갖고 온다.

이를 통해 버스 도착 정보목록 API를 호출한다.


```
▼<ServiceResult>
   <comMsgHeader/>
  ▼<msgHeader>
     <headerCd>0</headerCd>
     <headerMsg>정상적으로 처리되었습니다.</headerMsg>
     <itemCount>0</itemCount>
   </msgHeader>
  ▼<msgBody>
    ▼<itemList>
       <arsId>14871</arsId>
       <posX>193275.35797421218</posX>
       <posY>450392.4809009908</posY>
       <stId>113900144</stId>
       <stNm>홍대정문</stNm>
       <tmX>126.9238877068</tmX>
       <tmY>37.5529817959</tmY>
     </itemList>
    ▼<itemList>
       <arsId>14890</arsId>
       <posX>193254.35846302955</posX>
       <posY>450283.9808850507</posY>
       <stId>113900158</stId>
       <stNm>홍대정문</stNm>
       <tmX>126.9236510232</tmX>
       <tmY>37.5520039556</tmY>
     </itemList>
    ▼<itemList>
       <arsId>14899</arsId>
       <posX>193358.7560158751</posX>
       <posY>450411.180901045</posY>
       <stId>113900162</stId>
       <stNm>홍대정문</stNm>
       <tmX>126.9248314705</tmX>
       <tmY>37.5531509051</tmY>
     </itemList>
   </msgBody>
 </ServiceResult>
```

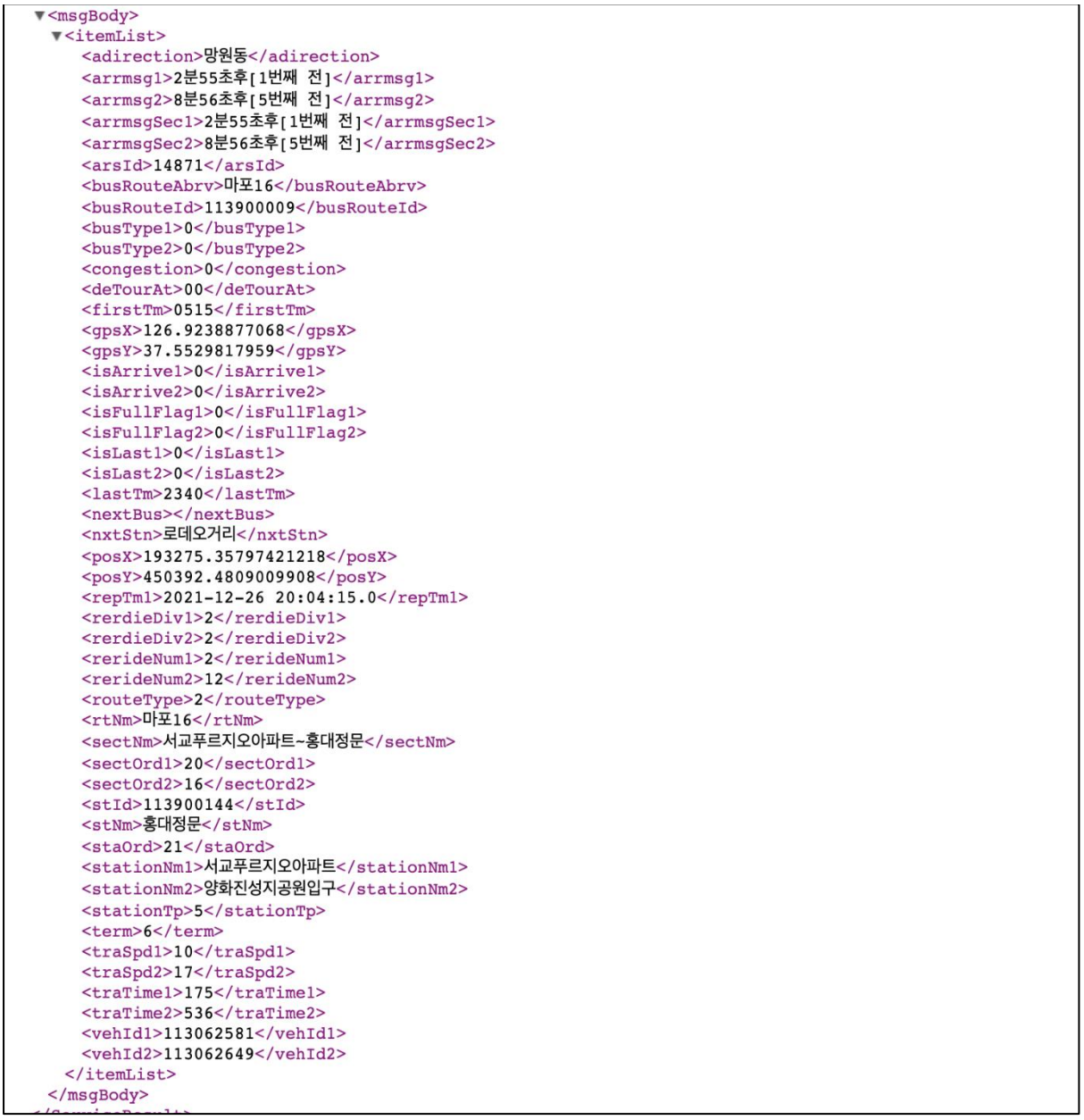

[정류소 명칭 검색 Response]

호출하여 받아온 버스 정보 중 Transferlist 내 BusNm을 사용하여 일치하는 rtNm의 버스 도착 메세지 arrmsg, tratime, VehId 변수를 저장하여 사용한다.

vehicleid는 도착 예정 버스의 고유 번호이며 이를 통해 도착정보 API를 호출해 push 메세지에서 사용하는 고유한 버스번호판을 받아 오게 된다.

arrmsg에서는 [n번째 전] 만을 추출하여 사용하고 남은 시간은 tratime을 분 초로 변환하였으며 count down timer를 통해 API의 재호출 없이 매초 시간이 줄도록 구현하여 view에 메세지를 띄웠다. 또한 도로 상황에 따라 도착 시간이 달라질 수 있음을 고려하여 15초마다 API를 재호출하게 구현하였다. API 호출 시간을 줄인 이유는 0초가 됐을 때 다음 버스가 바로 넘어가지지 않아 0초~1초사이의 매우 짧은 순간 무수히 많은 수의 호출을 하게 되어 텀을 정하였다.

또한 view를 한 번 클릭하면 버스 도착시간에 대한 음성 안내를 두 번 연속으로 클릭하면 탑승했다고 판단하여 내려야 할 버스 정류장의 이름을 뷰에 띄워주었다. 마찬가지로 한 번 클릭하면 하차지를 음성 안내하고 두 번 클릭 시 하차지부터 목적지 또는 하차지부터 환승지까지 도보 안내를 하도록 데이터를 수정하여 보행 경로 안내 액티비티로 전달한다.



```
▼<msgBody>
  ▼<itemList>
      <adirection>망원동</adirection>
      <arrmsg1>2분55초후[1번째 전]</arrmsg1>
      <arrmsg2>8분56초후[5번째 전]</arrmsg2>
      <arrmsgSec1>2분55초후[1번째 전]</arrmsgSec1>
      <arrmsgSec2>8분56초후[5번째 전]</arrmsgSec2>
      <arsId>14871</arsId>
      <busRouteAbrv>마포16</busRouteAbrv>
      <busRouteId>113900009</busRouteId>
      <busType1>0</busType1>
      <busType2>0</busType2>
      <congestion>0</congestion>
      <deTourAt>00</deTourAt>
      <firstTm>0515</firstTm>
      <gpsX>126.9238877068</gpsX>
      <gpsY>37.5529817959</gpsY>
      <isArrive1>0</isArrive1>
      <isArrive2>0</isArrive2>
      <isFullFlag1>0</isFullFlag1>
      <isFullFlag2>0</isFullFlag2>
      <isLast1>0</isLast1>
      <isLast2>0</isLast2>
      <lastTm>2340</lastTm>
      <nextBus></nextBus>
      <nxtStn>로데오거리</nxtStn>
      <posX>193275.35797421218</posX>
      <posY>450392.4809009908</posY>
      <repTm1>2021-12-26 20:04:15.0</repTm1>
      <rerdieDiv1>2</rerdieDiv1>
      <rerdieDiv2>2</rerdieDiv2>
      <rerideNum1>2</rerideNum1>
      <rerideNum2>12</rerideNum2>
      <routeType>2</routeType>
      <rtNm>마포16</rtNm>
      <sectNm>서교푸르지오아파트~홍대정문</sectNm>
      <sectOrd1>20</sectOrd1>
      <sectOrd2>16</sectOrd2>
      <stId>113900144</stId>
      <stNm>홍대정문</stNm>
      <staOrd>21</staOrd>
      <stationNm1>서교푸르지오아파트</stationNm1>
      <stationNm2>양화진성지공원입구</stationNm2>
      <stationTp>5</stationTp>
      <term>6</term>
      <traSpd1>10</traSpd1>
      <traSpd2>17</traSpd2>
      <traTime1>175</traTime1>
      <traTime2>536</traTime2>
      <vehId1>113062581</vehId1>
      <vehId2>113062649</vehId2>
    </itemList>
  </msgBody>
```

[버스 도착 정보목록 API response]

[버스 도착정보] **[하차지 안내]**

6.6 경로 안내 서비스

사용자가 어떤 버스 노선을 이용할 것인지 선택한다면, 그와 동시에 보행 경로 및 장애물 안내 액티비티로 전환된다. 이때 액티비티가 전환되며 목적지 좌표와 정류장 리스트의 좌표가 함께 전달된다. 이후 사용자의 실시간 위치를 파악하여 TMAP API에 request를 보내고 응답 데이터를 바탕으로 보행 경로 정보를 안내한다. 모든 정보는 각 노드에서 음성으로 안내되며, 안내하는 정보로는 걸음 수로 환산한 다음 노드까지의 거리, 우회전과 좌회전 등의 방향 정보, 도로 타입, 횡단보도나 계단과 같은 시설물 유무 등이 있다.

[경로 안내 서비스 (구버전)]

위 사진은 지난 데모 테스트 당시 실행 화면으로, 여러 정보가 적절하게 안내됨을 소개하고자 첨부하였다. 구버전의 경우 새로고침 버튼을 누르면 다음 노드까지 남은 거리를 계산하여 안내하였으나, 최종 버전의 경우 UI 대부분이 실시간 카메라 영상이므로 별도의 버튼을 두지 않고 화면을 한 번 터치했을 때 이를 안내하도록 구현하였다.

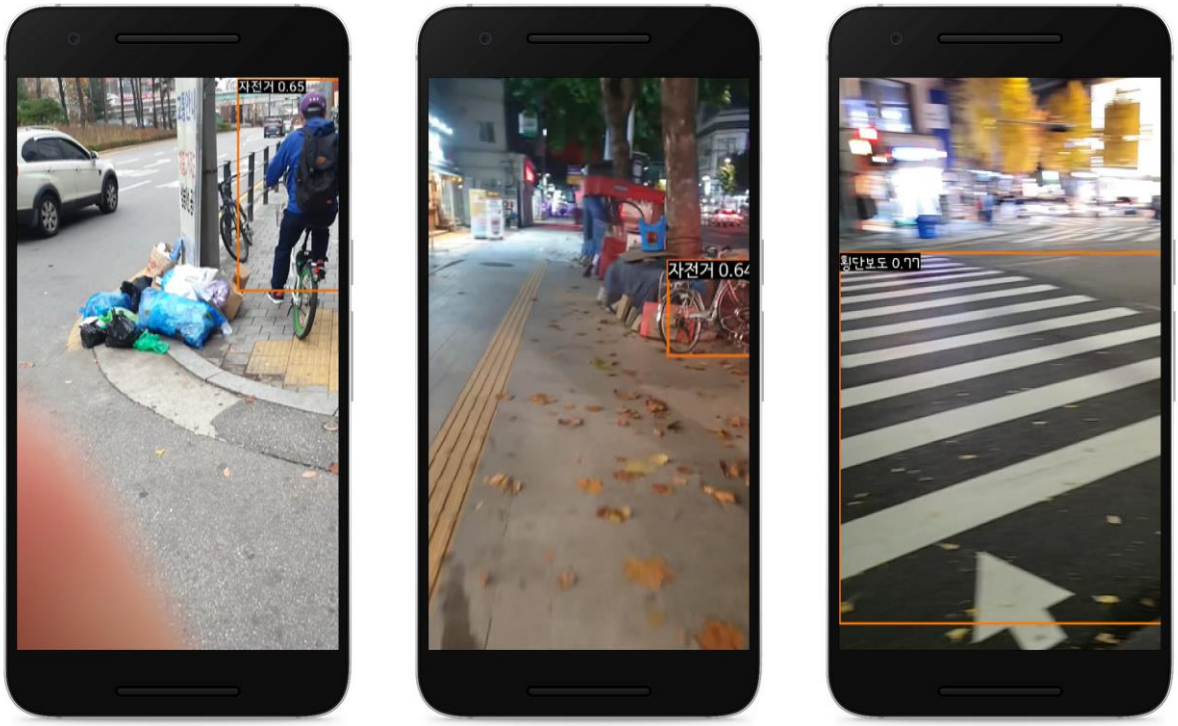
이 기능을 통해 출발지부터 정류장까지의 경로, 환승 횟수에 따라 하차지에서 다음 정류장까지의 경로, 그리고 최종적으로 마지막 정류장에서 목적지까지의 보행 경로를 안내하게 된다.

6.7 장애물 인식 서비스

6.6 경로 안내 서비스와 동시에, 스마트폰에 내장된 카메라를 활용하여 보행 중 사용자 전방에 인식되는 객체를 실시간으로 안내한다. 이 과정에서 고성능 장치가 아닌 모바일 기기와 같은 임베디드 시스템에도 적용하여 사용할 수 있는 딥러닝 프레임 워크인 TensorFlow Lite를 사용하였다.

먼저 웹 어플리케이션 자동화를 위한 프레임워크인 Selenium과 ChromeDriver를 통해 Google의 이미지를 크롤링하였고, 이후 labelImg를 사용한 라벨링 작업을 거쳐 직접 데이터셋을 확보하였다. 데이터셋을 Train data, Validation data, Test data로 분할하여 csv 파일을 생성한 후 TensorFlow

Lite Model Maker 라이브러리를 활용하여 분류 모델을 제작하였다. 그리고 완성된 모델을 안드로이드 프로젝트에 적용하여 사용하였다.



[장애물 인식 서비스]

학습된 객체로는 사용자의 보행에 영향을 미칠 수 있는 자전거, 킥보드, 볼라드, 횡단보도 등이 있다. 보행 중 카메라에 이 객체들이 탐지된다면 이를 안내하게 된다. 특정 객체가 무한히 안내되는 것을 방지하고자 약 4초 주기로 인식된 객체를 안내하며, 나란히 설치된 볼라드와 같이 동일한 범주의 객체 여러 개가 동시에 인식되는 경우 중복 안내를 막고 이를 한 번만 안내하도록 구현하였다.

7. 결론

우리는 보행자의 안전성을 높이고, 적절한 정보를 제공하여 버스 이용에 대한 시각장애인의 접근성을 향상하고자 하였다. 졸업 전시회 당시 많은 관람객으로부터 '네비게이션 같다'는 총평을 듣게 되었는데, 이것이 바로 본 프로젝트의 목표라 할 수 있다. 보다 정확한 도로 정보와 데이터셋이 있고, 스마트폰 기기의 GPS 및 각종 센서가 더욱 개선된다면 실생활에서 시각장애인의 길잡

이가 되어줄 어플리케이션이 충분히 만들어질 수 있을 것이다. '시각장애인을 위한 길 안내 서비스'가 그 가능성을 보여주었다고 생각하며, 나아가 교통약자와 디지털 소외계층을 위한 소프트웨어 개발이 더 활발히 이루어져 이들의 삶의 질 향상에 기여할 수 있기를 기대해 본다.