

교통약자를 위한 대중교통 모바일 애플리케이션

안민¹, 강철수^{2*}

¹서원대학교 컴퓨터공학과 학생, ²케이앤엑스정보통신 기업부설연구소 연구소장

Public Transportation Mobile Application for Individuals with Mobility Challenge

Min An¹, Cheol-Soo Kang^{2*}

¹Student, Department of Computer Engineering, Seowon University

²Director, Company-affiliated Research Institute, KNX Information & Communication

요약 본 논문은 안드로이드 및 iOS 환경에서 교통약자들이 대중교통을 더 편리하게 이용할 수 있도록 하는 모바일 애플리케이션에 대한 연구를 다루고 있다. 교통약자를 시각, 청각, 또는 기타 제한을 겪는 개인으로 정의하며, 이들이 기존 대중교통 애플리케이션에서 마주치는 도전적인 상황을 구체적으로 분석한다. 기존 애플리케이션에서는 교통약자들이 정보에 접근하거나 이동 계획을 세우는 데 어려움을 겪는 경우가 있다. 이에 대응하여, 논문에서는 시각적으로 어려운 사용자를 위한 화면 낭독 지원 및 음성 안내 기능과 같은 특수 기능을 도입하여 교통약자들에게 최적화된 사용자 환경을 제공한다. 이 연구의 목표는 교통약자들이 독립적으로 이동할 수 있도록 지원하는 특화된 모바일 애플리케이션을 개발하는 것이다. 이를 통해 기존의 문제점을 극복하고 교통약자들에게 편의성과 독립성을 제공하여 일상적인 이동에 대한 어려움을 해소하는 방안을 제시하고자 한다.

주제어 : 모바일 애플리케이션, 대중교통, 교통약자, 사용자 환경 최적화, 편의성 향상

Abstract This paper discusses a study on a mobile application aimed at making public transportation more convenient for people with mobility challenges on both Android and iOS platforms. The research analyzes the limitations and weaknesses of existing mobile applications for public transportation from the perspective of individuals with mobility challenges. The goal is to overcome these limitations and provide an optimized user experience. The motivation behind this research stems from the recognition that people with mobility challenges face difficulties in their daily commute, and current public transportation applications do not adequately cater to their needs. Consequently, the study aims to develop a specialized mobile application for individuals with mobility challenges to support them in achieving greater independence in their daily travels.

Key Words : Mobile Application, Public Transportation, Transportation Accessibility, User Experience Optimization, Enhanced Usability

*Corresponding Author : Cheol-Soo Kang(ily01@hanmail.net)

Received February 15, 2024
Accepted March 21, 2024

Revised February 28, 2024
Published March 30, 2024

1. 서론

“찾길”은 교통약자를 대상으로 대중교통을 통해 원하는 장소로 가기 위한 길을 찾는다는 의미를 가진 기존 대중교통 모바일 애플리케이션과는 차별화된 새로운 모바일 애플리케이션이다.

기존의 대중교통 모바일 애플리케이션은 교통약자의 관점에서 바라본다면 너무 많은 기능을 가지고 있어 오히려 사용자에게 불편함을 줄 수 있어 간단하지만 복잡하지 않은 절차로 사용자가 쉽게 자신이 원하는 출발지, 도착지를 선정할 수 있고 자체적으로 구현된 경로 탐색 알고리즘을 통해 손쉽게 대중교통 경로를 확인하면서 안전하게 도착지를 갈 수 있는 모바일 애플리케이션을 제작하였다[1,2].

본 모바일 애플리케이션은 사용자가 출발하기 전 부터 당일 날씨 정보를 체크할 수 있도록 기상 정보를 제공하며 이를 통해 외출시 기상 상황을 고려한 준비를 할 수 있도록 지원한다[3]. 또한 사용자가 타려는 버스가 사용자가 원하는 정거장으로부터 몇 정거장 전에 위치해 있는지 인지 할 수 있고 이때 음성을 통해 버스 정보를 전달하여 사용자에게 도움을 줄 수 있다[4].

사용자가 입력한 정보는 모바일 애플리케이션과 네트워크로 연결되어 있는 서버의 데이터베이스에 저장되며 사용자별로 최근 기록으로부터 50건에 해당하는 정보를 기록한다[5]. 또한 경로 탐색에 있어 최적의 경로를 선정할 수 있다. 교통약자를 대상으로 하였기 때문에 기본적으로 사용자 인터페이스는 한 화면에 모든 정보를 보여주기 보다는 직관적으로 확인할 수 있도록 버튼 하나 하나가 잘 보이는 디자인을 사용함으로써 프로그램을 이해하지 못하거나 어려워 하지 않도록 하는 방향성을 중점으로 개발되었다. 최종적으로 시스템 구성도 및 논리적인 동작 알고리즘 등은 아래 기술된 2장의 연구내용을 통해 보다 세부적으로 기술한다.

2. 설계

2.1 시스템 구성도 설계

2.2.1 시스템 설계

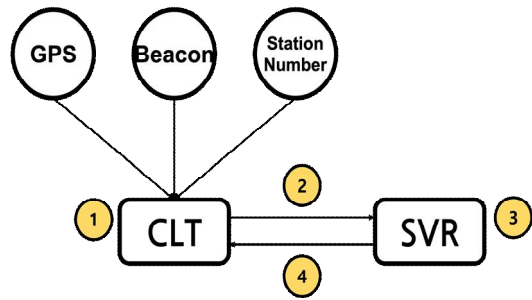


Fig. 1. System Architecture Configuration Diagram

본 모바일 애플리케이션의 시스템 아키텍처는 Fig. 1과 Fig. 2와 같이 구성되어 있다. 우선 Fig. 1은 GPS, Beacon, 정거장 번호 등을 통해 사용자가 수집한 정보를 서버로 전달하고 서버는 이를 통해 사용자에게 정거장의 기본적인 정보를 제공하는 일련의 과정을 표현한 구성도이다. 이를 통해 사용자는 움직이지 않고 정거장의 기본정보와 정거장 및 인근 지역의 기상 정보를 제공받을 수 있다[6]. 이를 통해 사용자가 기상 환경에 따른 적절한 외출 준비를 할 수 있도록 도와준다[7,8].

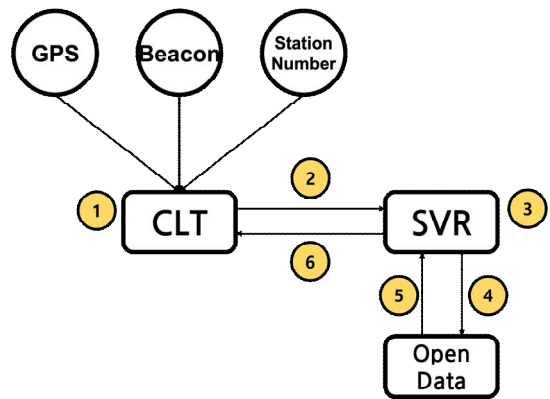


Fig. 2. System Architecture Configuration Diagram

또한 Fig. 2와 같이 공공데이터포털의 OpenAPI(Open Application Programming Interface)서비스와 서버 내 데이터베이스를 활용하여 사용자에게 최적화 대중교통 경로를 전달할 수 있는 서버-클라이언트 방식으로 설계되었다[9]. 모든 데이터는 서버를 통해 클라이언트로 전송되며 이를 통해 대중교통 환승경로 등 대량의 데이터를 복잡한 연산을 통해 정제함에 있어 사용자별로 성능 편차가 심한 모바일 환경에서 연산을 수행하는 것 보

다 성능적인 이점을 얻을 수 있었으며 실제 사용자가 설치를 해야 하는 모바일 애플리케이션의 용량을 최소화할 수 있다.

2.2.2 데이터베이스 설계

2.2.2.1 개발환경

본 모바일 애플리케이션의 데이터베이스는 로컬 서버 테스트를 위한 JAVA 기반의 관계형 데이터베이스 시스템(RDBMS) H2와 실제 환경에서 사용될 오픈소스 RDBMS MySQL로 구성된다.

2.2.2.2 모델정의

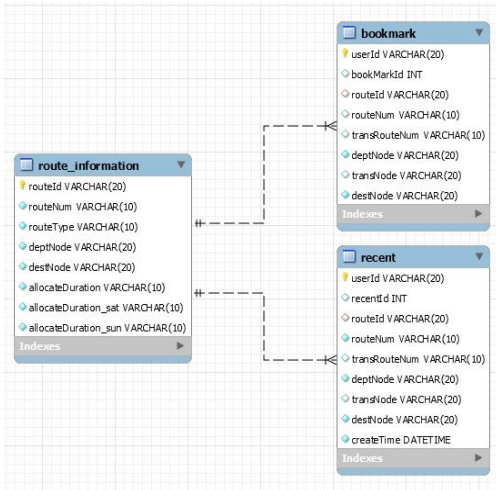


Fig. 3. E-R Diagram

Fig. 3의 다이어그램과 같이 데이터베이스 스키마는 3개의 테이블을 정의하고 있다.

route_infomation은 버스 노선과 관련된 데이터를 가지고 있다. 이를 통해 공공데이터포털에서 데이터를 받지 못하는 경우가 발생하더라도 서버는 데이터베이스에 보유한 데이터만으로 대중교통 경로 제공이 가능하도록 고려할 수 있다. bookmark는 사용자가 즐겨찾기로 추가한 대중교통 경로를 저장하는 테이블이다. 사용자 별 모바일 기기의 식별을 위해 IMEI(International Mobile Equipment Identity)번호를 사용하며 이를 통해 사용자별 뷰(View)를 구성할 수 있다[10]. recent는 사용자의 최근 기록을 저장하는 테이블로 사용자별 최대 50개까지 저장할 수 있다. 또한 bookmark와 마찬가지로 IMEI번호를 통해 사용자 식별이 가능하다.

2.2.2.3 테이블 정의

Table 1. Route Information Table

FIELD NAME	TYPE	NULL	KEY
routeId	VARCHAR(20)	NN	PK
routeNum	VARCHAR(10)	NN	
routeType	VARCHAR(10)	NN	
deptNode	VARCHAR(20)	NN	
destNode	VARCHAR(20)	NN	
allocate	VARCHAR(10)	NN	
allocate_sat	VARCHAR(10)	NN	
allocate_sun	VARCHAR(10)	NN	

Table 1의 테이블은 버스 노선과 관련된 테이블의 정보를 가지고 있다. 버스 노선 데이터의 특성상 숫자뿐 아니라 “-”부호 등이 사용될 소지가 있으므로 모든 데이터는 문자열로 저장되어 관리된다.

Table 2. Route Information Table

FIELD NAME	TYPE	NULL	KEY
userId	VARCHAR(20)	NN	PK
bookMarkId	INT	NN	UQ
routeId	VARCHAR(20)	NN	FK
routeNum	VARCHAR(10)	NN	
trRouteNum	VARCHAR(10)		
deptNode	VARCHAR(20)	NN	
transNode	VARCHAR(20)		
destNode	VARCHAR(20)	NN	

Table 2의 즐겨찾기 테이블은 본 논문의 모바일 애플리케이션을 이용하는 사용자들이 저장한 모든 즐겨찾기 경로 정보가 저장된다. 사용자 식별을 위해 userId 컬럼의 값을 사용하며 해당 컬럼에는 사용자별 단말기기의 IMEI번호가 저장된다.

Table 3. Recent Records Table

FIELD NAME	TYPE	NULL	KEY
userId	VARCHAR(20)	NN	PK
recentId	INT	NN	UQ
routeId	VARCHAR(20)	NN	FK
routeNum	VARCHAR(10)	NN	
trRouteNum	VARCHAR(10)		
deptNode	VARCHAR(20)	NN	
transNode	VARCHAR(20)		
destNode	VARCHAR(20)	NN	
createTime	DATETIME	NN	

Table 3의 최근 기록 테이블은 사용자들이 조회한

모든 경로 정보가 저장된다. 2.4.1에 기술되어 있는 Figure 5의 최근 기록에 따라, 사용자별로 최대 50개의 기록을 저장할 수 있다. 보다 자세한 데이터 저장에 관한 내용은 2.3을 참고한다.

2.2 서버 설계

2.2.1 개발환경

본 모바일 애플리케이션의 서버 프로그램은 자바(Java)언어와자바에서 제공하는 웹 프레임워크 스프링부트(Springboot)를 활용한 Rest(Representational State Transfer)방식의 웹서버로 개발되었으며 리눅스 기반의 서버컴퓨터에서 동작하도록 구성되었다.

2.2.2 공공데이터 정보 파싱

서버 프로그램은 공공데이터 포털의 OpenAPI서비스를 통해 경로 탐색에 필요한 대중교통 정보를 전달받는다. 본 논문에서 사용된 공공데이터는 “국토교통부_버스정류소정보”, “국토교통부_버스노선정보”, “국토교통부_버스도착정보” 총 3가지 이며 JSON형태의 데이터를 RestAPI방식으로 전달받은 뒤 클라이언트의 요청에 따라 별도의 JSON 형태로 가공하여 클라이언트로 전달한다[11].

2.2.3 데이터베이스연동

서버 프로그램의 연산 소요시간이 오래 걸리는 일부 오퍼레이션 에서 클라이언트와 서버의 빠른 응답을 위해 서버는 클라이언트의 요청 시에 미리 데이터베이스에 캐싱 된 데이터를 클라이언트로 전달한다.

2.2.4 버스 경로 알고리즘

본 모바일 애플리케이션 사용자는 클라이언트에 출발할 정류장과 목적지 정류장을 입력한다. 입력 받은 출발지 정류장, 목적지 정류장의 문자열 데이터를 파라미터로 서버에 전송한다. 서버에서 위 데이터를 가공하여 공공데이터 포털의 API를 통해 각 “정류장의 도착 예정 버스 리스트”를 받아 오도록 요청한다. 얻어온 리스트를 JSON으로 파싱해 오퍼레이션을 수행한다[12].

직통 노선에 대한 오퍼레이션은 출발지 정류장과 목적지 정류장을 무조건 경유하는 노선이 된다. 먼저 본 논문의 “버스 경로 알고리즘” 첫 번째 문단의 과정을 수

행 후 두 리스트를 파라미터로 오퍼레이션을 수행 한다. 출발지 정류장의 도착예정 버스 정보가 담긴 리스트를 1. 목적지 정류장의 도착예정 정보가 담긴 리스트를 2로 취급 한 후 리스트1과 리스트2의 크기만큼 중첩 반복문을 사용해 노선 번호가 일치하는 것이 있으면 해당 노선에 대한 정보(출발지 정류장이름, 예상 도착시간, 목적지 정류장이름, 노선번호 등)를 JSONObject로 묶어 결과 JSONArray에 추가한다. 모든 반복이 종료되면 결과 리스트를 반환하고 클라이언트에선 정제된 리스트에서 원하는 데이터를 뽑아 출력한다.

1회 환승에 대한 오퍼레이션의 기준은 환승 지점이다. 먼저 출발지 정류장에서 경유하는 버스들이 사용자가 원하는 목적지 정류장으로 도달하지 않을 때 어떠한 지점에서 환승이 필요하게 된다. 이에 출발지 정류장을 가는 버스들의 모든 경우 정류장과 목적지 정류장을 가는 버스들의 모든 경우 정류장을 탐색해 일치하는 지점이 있으면 그 곳이 바로 환승 지점이 될 것이다.

따라서 출발지 정류장의 도착예정 버스정보가 담긴 리스트1, 목적지 정류장의 도착예정 버스정보가 담긴 리스트2에서 각 리스트의 요소, 즉 버스 노선에 대한 모든 경우 정거장 정보를 받아와 JSONArray로 묶는다. 이 리스트를 본 논문에서 리스트 1-1, 2-1로 정의. 리스트 1-1과 2-1의 크기만큼의 중첩 반복문을 사용 리스트 1-1을 순차탐색하고 리스트2-1을 역순으로 탐색 하면서 일치하는 정류장이 있으면 그시점에서 출발지 정류장부터 환승지점 정류장에 대한 리스트를 결과 리스트1. 환승지점 정류장에서 도착지 정류장까지에 대한 리스트를 결과 리스트2. 두 결과 리스트를 JSONObject로 묶어 최종 결과 리스트가 될 JSONArray에 추가한다. 모든 반복이 종료되면 결과 리스트를 반환하고 클라이언트에서 정제된 리스트에서 원하는 데이터를 뽑아 출력한다.

2.3 클라이언트 설계

2.3.1 개발환경

본 모바일 애플리케이션의 클라이언트 프로그램은 사용자 모바일 환경이 다양할 수 있는 점을 고려하여 Flutter를 통해 안드로이드와 iOS환경에서 동시에 동작할 수 있도록 개발되었다. 이와 같은 운영체제로부터 독립적인 언어를 통해 개발 및 관리에 있어 하나의 프로젝트를 통해 관리할 수 있었으며 향후 웹 서비스 등으로 확장 할 수 있는 가능성을 고려할 수 있다.

2.3.2 화면 구성

화면은 크게 경로탐색, 최근기록, 즐겨찾기로 구성되어 있다.

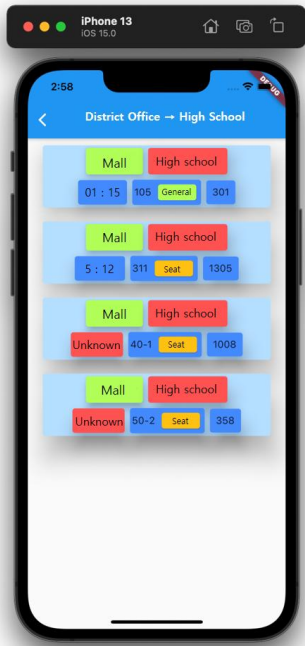


Fig. 4. Route Search Screen

Fig. 4과 같이 경로탐색 화면의 경우 출발지와 도착지 정보를 입력할 수 있는 버튼이 있으며 터치입력을 통해 각각 정류장 정보를 입력할 수 있다. 출발지와 도착지에 정류장 정보가 입력되면 자동으로 Figure 5의 경로조회결과 화면으로 전환되며 동시에 서버 프로그램으로 RestAPI의 GET방식을 통해 경로 탐색을 요청한다.

서버로부터 경로 탐색 결과를 수신 받으면 Figure 6의 화면에 출발지부터 도착지까지 대중교통의 이동경로가 리스트 형태로 출력되며 원하는 경로를 선택하면 해당 경로의 정보를 사용자에게 제공한다[13].

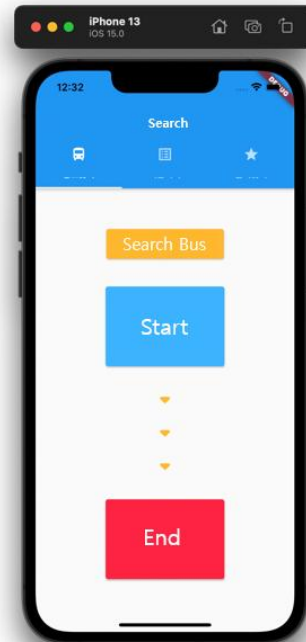


Fig. 5. Route Search Results

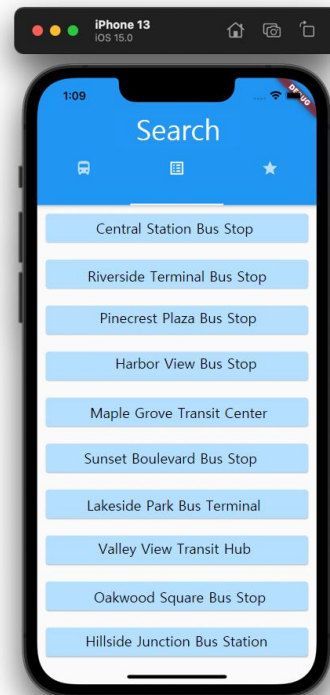


Fig. 6. Recent Records Screen

Fig. 4의 화면에서 선택한 경로 정보는 선택함과 동시에 RestAPI의 PUT방식을 통해 서버로 전달되어 데이터베이스에 저장된다[14]. 이렇게 저장된 기록은 Figure 4의 최근 기록 화면을 통해 확인할 수 있으며 최대 50개까지의 기록을 저장할 수 있다. 50개를 넘는 기록이 발생하면 가장 오래된 기록은 RestAPI의 DELETE 방식으로 삭제된 후 현재 경로에 대한 기록이 저장된다.

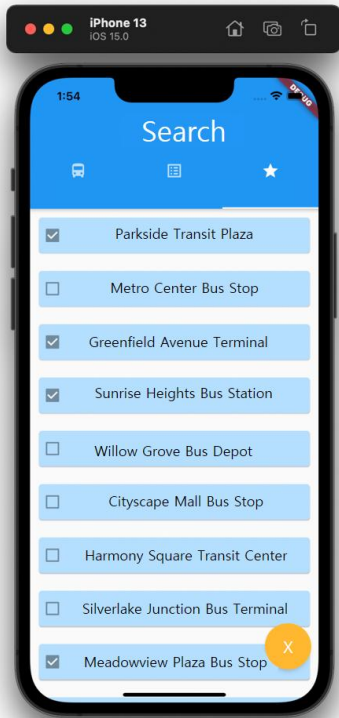


Fig. 7. Favorites Screen

Fig. 6에서 선택된 경로는 사용자가 임의로 즐겨찾기 기능을 통해 저장할 수 있다. 즐겨찾기 기능을 통해 저장한 경로 정보는 Figure 6에서 정보를 저장하는 것과 동일하게 PUT방식으로 서버를 통해 데이터베이스에 저장된다. 이후 Figure 7의 즐겨찾기 화면을 통해 저장한 경로를 조회할 수 있으며 즐겨찾기에 저장된 정보는 사용자가 삭제를 하기 전까지 삭제되지 않는다[15].

2.3.3 내부 동작

클라이언트 프로그램은 내부적으로 Flutter 디자인 패턴(Design Pattern)의 방식 중 하나인 프로바이더

(Provider)를 통해 작성되었다. 아래에 작성된 코드는 본 모바일 애플리케이션의 내부 동작을 간단한 형태로 표현한 것으로 모든 동작은 작성된 코드 이후에 기술된 내용을 바탕으로 연산을 수행한다[16].

```
class Provider with ChangeNotifier{
  var _data;
  var get data => _data;

  function(){
    //do something
    notifyListeners();
  }
}
```

Fig. 8. Business Logic

```
class UI extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MultiProvider(
      providers: [ChangeNotifierProvider(create: (_) => Provider()),],
      child: Body()
    );
  }
}

class Body extends StatefulWidget {
  @override
  _BodyState createState() => _BodyState();
}

class _BodyState extends State<Body> {
  @override
  Widget build(BuildContext context) {
    final provider = Provider.of<Provider>(context);

    return Column(
      children: [
        ElevatedButton(
          onPressed: () {
            provider.function();
          },
          child: Text("${provider.data}")
        ),
      ],
    );
  }
}
```

Fig. 9. User Interface

Fig. 8은 클라이언트 프로그램 내부에서 실질적인 연산을 수행하는 비즈니스 로직이다. 별도의 비즈니스 로직 클래스를 구현함으로써 Flutter가 가진 단점인 UI와 비즈니스 로직의 구별이 어려운 점을 극복하였으며 보다 체계적인 코드를 작성할 수 있다.

비즈니스 로직인 Provider클래스는 function함수의 연산을 통해 data변수를 다룰 수 있으며 data의 변동사항을 사용자 인터페이스에 나타내고자 할 경우 반드시 notifyListeners함수를 function함수 마지막에 작성해야 한다.

Fig. 9은 사용자 인터페이스를 구현한 코드이다. 클라이언트 프로그램은 ChangeNotifierProvider를 통해 비즈니스 로직과 사용자 인터페이스를 연동하며 Provider.of<ProviderName>을 통해 사용자 인터페이스에서 비즈니스 로직의 함수와 변수를 호출할 수 있다.

3. 결론

이 연구에서는 안드로이드 및 iOS 환경에서 교통약자들을 대상으로 하는 특화된 모바일 애플리케이션의 개발과 최적화에 관한 연구를 수행했다. 기존 대중교통 모바일 애플리케이션의 한계와 약점을 분석하여 교통약자 관점에서의 사용자 경험을 향상시키기 위해 노력했다. 연구 결과, 새로 개발된 모바일 애플리케이션은 교통약자들에게 높은 편의성을 제공하는 것으로 나타났다. 이 애플리케이션은 사용자가 일상적인 이동에서 경험하는 어려움을 극복하고 독립적으로 이동할 수 있도록 지원하는 기능을 포함하고 있다. 또한 사용자들로부터의 피드백을 토대로 애플리케이션의 기능을 지속적으로 개선하는 방향으로 노력하였다.

이러한 연구 결과는 교통약자들의 삶의 질을 향상시켜 그들의 불편함 점을 해소할수 있는데에 큰 기여할 수 있을 것으로 기대된다. 더불어, 이 연구에서 개발된 모바일 애플리케이션은 향후 대중교통 시스템의 발전과 교통약자들의 사회 참여 증진에 긍정적인 영향을 미칠 것으로 전망된다. 종합적으로, 교통약자를 위한 모바일 애플리케이션의 개발 및 최적화에 대한 본 연구는 새로운 가능성을 제시하며, 향후 더 많은 연구와 현장 적용을 유도할 것이다.

REFERENCES

- [1] C. H. Hwang. (2018). *Current Status of Transportation Utilization by Transportation Disadvantaged Individuals*. Tata Daewoo Commercial Vehicle Co., Ltd.
DOI : 10.23000/TRKO201800002440.48
- [2] H. S. Lee. (2017) *Variable public transportation demand-based alternative 1 Figure 40 Variable public transportation demand-based alternative*. Korea Institute of Science and Technology Information.
DOI : 10.23000/TRKO201700008923.40.
- [3] M. O. Kang. (2015). *Frequency analysis results for public transportation use: public transportation*. Korea Environmental Policy and Evaluation Institute.
DOI : 10.23000/TRKO201800014507.29.
- [4] J. C. Kim. (2013). *Simulation of airport facilities for the transportation vulnerable*. Korea Transport Institute.
DOI : 10.23000/TRKO201400002980.35.
- [5] J. K. Kim. (2011). *Target for installation of transportation convenience facilities for the transportation vulnerable*. Korea Research Institute for Human Settlements.
DOI : 10.23000/TRKO201300013724.88.
- [6] K. H. Ma. (2011). *Major plans related to transportation convenience policies for the transportation vulnerable*. Korea Women's Policy Institute.
DOI : 10.23000/TRKO201300015106.224.
- [7] C. H. Hwang. (2018). *Status of the transportation vulnerable*. Tata Daewoo Commercial Vehicles Co., Ltd.
DOI : 10.23000/TRKO201800002440.46.
- [8] Y. S. Park. (2018). *Proportion of the transportation vulnerable*. Tata Daewoo Commercial Vehicle Co., Ltd.
DOI : 10.23000/TRKO201800002440.236.
- [9] K. A. Kim (2011). *Urban Mobility Simulation*. *Journal of The Korea Computer Graphics Society*, 17(4), 23-30.
DOI : 10.15701/kcgs.2011.17.4.23.
- [10] K. H. Kim. (2014). *Android platform and Android package creation process*. Soongsil University Industry-Academic Cooperation Foundation.
DOI : 10.23000/TRKO201400012342.12.
- [11] W. C. Kang. (2017). *Android platform structure (SQLite, an embedded database, is provided as a core service built into the Android platform*. Incheon National University.
DOI : 10.23000/TRKO201800004343.3.
- [12] Y. G. Lee. (2016). *JSON Format*. Gwangju Institute of Science and Technology.
DOI : 10.23000/TRKO201600015504.143.
- [13] S. M. Lee. (2012). *Changes in express bus traffic volume*. Korea Transport Institute.
DOI : 10.23000/TRKO201300032687.6.
- [14] S. M. Oh. (2013). *Display Screen of Predicted Routes for Users According to Changes in Public Transportation Operation Patterns*. Korea Railroad Research Institute.
DOI : 10.23000/TRKO201800008795.30.
- [15] J. H. Oh. (2010). *A Model for Evaluating Connectivity in Transit Networks*. Korea Transport Institute
DOI : 10.34143/JTR.2010.17.1.1.
- [16] S. U. Choi (2016). *Micro-scale Public Transport Accessibility by Stations-KTX Seoul Station Case Study*. *Journal of Korean Society for Geospatial Information Science*, 24(1), 9-16.

DOI : 10.7319/kogsis.2016.24.1.009.

안 민(Min An)

[준회원]



- 2019년 3월 ~ 현재 : 서원대학교 컴
퓨터공학과 재학
- 관심분야 : 앱/웹, 데이터분석
- E-Mail : iededg@naver.com

강 철 수(Cheol-Soo Kang)

[정회원]



- 2015년 2월 : 국가평생교육원 정보
통신공학과(공학사)
- 2021년 10월 ~ 현재 : ㈜케이엔엑스
정보통신 기업부설연구소 연구소장
- 관심분야 : IoT 서비스, 관제 시스템
- E-Mail : ily01@hanmail.net