

## A Design and Implementation of Bus Operation Information Application Based on Android Platform

Seo Young Lee\*, Won Joo Lee\*\*

\*Student, Dept. of Computer Science & Engineering, Inha Technical College, Incheon Korea

\*\*Professor, Dept. of Computer Science & Engineering, Inha Technical College, Incheon Korea

### [Abstract]

In this paper, we propose an Android-based application that provides bus operation information. This application utilizes the Open API provided by the Public Data Portal to deliver real-time bus operation information for buses running in Incheon Metropolitan City. The bus operation information includes bus station, the list of buses stopping at each bus station, bus locations, bus arrival time, and the number of passengers on board. Users of this application can set their commuting times and register frequently used bus stations and bus routes as favorites to receive personalized bus operation information. Additionally, the application displays customized bus operation information on the main screen, allowing users to check the information intuitively. Furthermore, the scalability of this application allows it to be adapted to other regions by modifying the Open API provided by the Public Data Portal accordingly, making it possible to implement a real-time bus operation information application for other cities and regions.

▶ **Key words:** Android platform, Public Data Portal, Bus Station, Open API, SQLite, Room

### [요 약]

본 논문에서는 버스 운행 정보를 제공하는 안드로이드 플랫폼 기반의 애플리케이션을 설계하고 구현한다. 이 애플리케이션은 공공데이터포털에서 제공하는 Open API를 이용하여 인천광역시를 운행하는 버스의 운행 정보를 실시간으로 제공한다. 버스 운행 정보는 정류소 이름, 정류소에 정차하는 버스 목록, 버스 위치, 버스 도착 시간, 버스 탑승 인원 등을 포함한다. 이 애플리케이션의 사용자는 자신의 출·퇴근 시간을 설정하고, 해당 시간에 이용할 수 있는 정류소와 노선 버스 정보를 즐겨찾기에 등록함으로써 맞춤형 버스 운행 정보를 제공받을 수 있다. 또한, 맞춤형 버스 운행 정보를 앱의 메인화면에 출력함으로써 사용자가 직관적으로 확인할 수 있다는 장점이 있다. 그리고 이 애플리케이션의 확장성은 공공데이터포털에서 제공하는 Open API를 각 지역에 맞게 수정한다면 해당 지역의 버스 운행 정보를 실시간으로 제공하는 애플리케이션을 구현할 수 있다는 것이다.

▶ **주제어:** 안드로이드 플랫폼, 공공데이터포털, 버스 정류소, Open API, SQLite, Room

- 
- First Author: Seo Young Lee, Corresponding Author: Won Joo Lee
  - \*Seo Young Lee (hojoon00905@gmail.com), Dept. of Computer Science & Engineering, Inha Technical College
  - \*\*Won Joo Lee (wonjoo2@inhac.ac.kr), Dept. of Computer Science & Engineering, Inha Technical College
  - Received: 2025. 02. 04, Revised: 2025. 03. 10, Accepted: 2025. 03. 10.

## I. Introduction

대중교통은 자가용에 비해 교통난과 환경에 긍정적이면서 탑승 비용을 내면 이용할 수 있으므로 많은 사용자가 이용하고 있다. 국토교통부의 2023 국토교통 통계 연보 대중교통 현황조사에 따르면 주 이용 교통수단은 지하철이 31.5%, 시내버스가 68.5%로 시내버스 이용률이 가장 높음을 그림 1에서 볼 수 있다. 그리고 통행 목적으로는 출·퇴근, 등하교가 30.4%로 가장 높은 비율을 차지하고 있다[1].

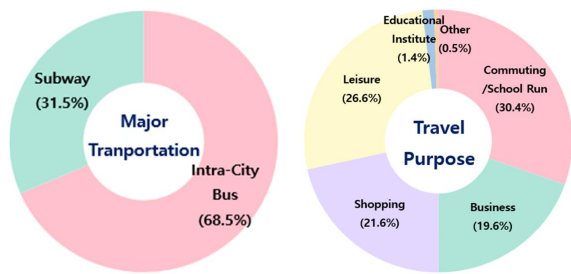


Fig. 1. Main transportation and transportation purposes in Korea [1]

또한 2023년 인천광역시 사회조사에 따르면 통근·통학 시 가장 많이 이용하는 교통수단은 대중교통이다. 대중교통 중에서는 시내버스가 29.3%, 도시철도가 17.5%로 시내버스 이용률이 가장 높다[2]. 그러나 인천광역시 시내버스 정류소 6,992개 중 버스 도착 안내 단말기가 설치된 정류소는 3,068개로 설치 비율이 약 43.88%이다[3][4]. 즉, 절반 이상의 정류소에는 버스 도착 안내 단말기가 설치되지 않아 통근·통학 시 대중교통 이용자들이 불편함을 느끼고 있다. 이러한 불편함을 줄일 수 있는 방법은 버스 운행 정보를 제공하는 애플리케이션을 제공하는 것이다.

따라서 본 논문에서는 안드로이드 플랫폼 기반의 버스 운행 정보를 제공하는 애플리케이션을 설계하고 구현한다. 이 애플리케이션은 인천광역시에서 통근·통학하는 대중교통 이용자가 탑승하는 버스의 도착 정보를 출근과 퇴근으로 나누어 저장하고, 이를 저장한 시간대에 메인화면이 나타나도록 함으로써 버스 도착 정보를 직관적으로 확인할 수 있도록 구현한다.

본 논문은 다음과 같이 구성한다. 2장에서 국내에서 서비스되고 있는 대중교통 앱 및 서비스들의 특징을 분석한다. 3장과 4장에서는 출퇴근 버스 운행 정보를 제공하는 안드로이드 플랫폼 기반의 애플리케이션을 설계하고 구현한다. 그리고 마지막으로 5장에서 결론을 맺는다.

## II. Preliminaries

### 1. Bench-marking

현재 국내에서 서비스 중인 지역 버스 도착 정보 제공 애플리케이션을 벤치마킹한 결과는 표 1과 같다.

Table 1. Bench-marking result

	Gyeonggi Bus Informaion	Ulsan Bus Information
Favorite buses and bus stops	O	O
Providing past bus arrival times	O	X
Commute settings	X	X

표 1에서 경기 버스 정보는 경기도청에서 개발한 안드로이드 플랫폼 기반의 애플리케이션이다. 버스 정류소 및 버스 즐겨찾기, 버스 도착 정보를 확인할 수 있는 기능을 제공한다. 또한 버스 도착 시간을 예측할 수 있는 과거 버스 도착 시간을 제공한다[5]. 울산 버스 정보는 울산교통관리센터에서 개발한 안드로이드 플랫폼 기반의 애플리케이션이다. 버스 정류소 및 버스 즐겨찾기, 버스 도착 정보를 확인할 수 있는 기능을 제공한다[6].

이러한 기존 애플리케이션의 경우, 즐겨찾기 한 정보와 도착 정보를 모두 하나의 화면에 나열하여 보여준다. 이는 출퇴근이나 등·하교와 같이 반복적으로 버스에 탑승할 때는 탑승하려는 버스 정보가 하단에 위치하여 늘 찾아야 한다는 단점이 있다. 따라서 출근과 퇴근의 시간대를 설정하고, 즐겨찾기를 출근과 퇴근으로 각각 설정하여 각 시간대에 탑승할 버스의 도착 정보를 확인할 수 있도록 가독성을 높이는 방법으로 구현하는 것이 필요하다.

### 2. Open API

출·퇴근 버스 운행 정보 제공 애플리케이션은 공공데이터 포털에서 제공하는 Open API를 이용하여 실시간으로 업데이트되는 인천광역시 버스 정보를 제공받는다[7]. 인천광역시\_정류소 조회, 인천광역시\_도착 정보 조회 기능을 제공하는 Open API는 표 2와 같다. 표 2에서 Incheon\_City\_Bus\_Stop\_Query API는 정류소 이름 목록 조회 기능으로 정류소 이름이 포함된 정류소 ID, 정류소 이름 등을 조회할 수 있다. 그리고 정류소 경유 노선 목록조회 기능으로 해당 정류소를 경유하는 노선의 ID, 노선명 등을 조회할 수 있다[8]. 그리고 Incheon\_City\_Arrival\_information\_Query API는 버스 도착 정보 목록조회 기능으로 정류소를 경유하

는 노선의 도착 예정 버스의 차량번호, 도착 예정 시간 등의 정보를 조회할 수 있다[9].

Table 2. Open API List

Open API Name	Function Name	Description
Incheon_City_Bus_Stop_Query	getBusStationNmList	Query information of the bus stops that include the bus stop name
	getBusStationViaRouteList	Query information of the routes that pass through the specified bus stop
Incheon_City_Arrival_information_Query	getAllRouteBusArrivalList	Query information of all routes passing through the bus stop

Open API에서 사용하는 요청 쿼리 파라미터와 사용할 응답 데이터는 표 3과 같다.

Table 3. Open API Specification

Function Name	Request Query Parameters	Response Data
GetBusStationNmList	bstopNm	BSTOPNM, BSTOPID
getBusStationViaRouteList	bstopId	ROUTEID, ROUTENO
getAllRouteBusArrivalList	bstopId	ROUTEID, REST_STOP_COUNT, ARRIVALESTIMATE TIME, CONGESTION

GetBusStationNmList는 입력한 정류소 이름과 비슷한 정류소 이름을 가진 정류소의 목록을 가져온 후, 그중 원하는 정류소의 이름과 ID를 가져오기 위해 사용한다. getBusStationViaRouteList는 정류소의 ID를 가지고 해당 정류소에 정차하는 노선의 ID와 번호를 가져오기 위해 사용한다. getAllRouteBusArrivalList는 즐겨찾기 한 정류소 ID로 요청한 후 즐겨찾기 한 노선 ID에 해당하는 몇 정거장 전, 도착예정시간, 혼잡도 정보를 가져오기 위해 사용한다.

### 3. Room

즐거찾기 한 정보는 SQLite 데이터베이스에 저장하기 위해 그림 2의 Room을 사용한다. Room은 데이터베이스를 보유하고 앱의 영구 데이터와 3가지의 기본 구성요소로

구성된다. 3가지의 기본 구성요소는 데이터베이스 클래스, 데이터 항목, 데이터 액세스 객체(DAO)이다. 데이터베이스 클래스는 기본 연결을 위한 기본 액세스 포인트의 기능을 제공한다. 데이터 항목은 앱 데이터베이스에서 테이블을 나타낸다. 데이터 액세스 객체는 데이터베이스의 데이터를 쿼리, 업데이트, 삽입, 삭제하는 데 사용할 수 있는 메소드를 제공한다. 또한, 데이터베이스 클래스는 데이터베이스와 연결된 DAO 인스턴스를 사용하여 데이터를 연결된 데이터 항목 객체의 인스턴스로 검색할 수 있게 된다. 앱은 정의된 데이터 항목을 사용하여 상응하는 테이블의 행을 업데이트하거나 삽입할 새 행을 만들 수 있다. SQLite를 직접 사용하는 것보다 오류를 줄이고 간소화할 수 있어 Room을 사용하여 로컬 데이터베이스에 데이터를 저장한다[10].

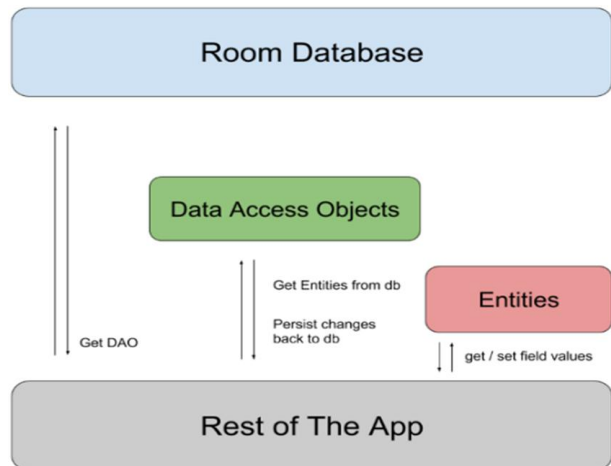


Fig. 2. Room Library Architecture Diagram

### 4. Shared Preferences

출·퇴근 시간대에 해당하는 화면을 출력하기 위해 Shared Preferences를 사용하여 출·퇴근 시간대를 저장한다. Shared Preferences는 저장하려는 키-값 컬렉션이 비교적 작은 경우에 사용하기 좋은 API이다. 키-값 쌍을 읽고 쓸 수 있는 간단한 메서드를 제공하며, 애플리케이션에 파일 형태로 데이터를 저장하고, 애플리케이션이 삭제되기 전까지 내부에 보관된다[11]. 출·퇴근 시간대 정보는 간단한 정보이기 때문에 데이터베이스가 아닌 Shared Preferences를 사용하여 간편하게 저장함으로써 빠르게 불러온다.

### III. The Design of Bus Operation Information Application

#### 1. Service Flow

버스 운행 정보 제공 애플리케이션은 사용자의 출·퇴근 시간에 맞추어 즐겨찾기 한 버스의 도착 정보를 제공한다. 출·퇴근 버스 운행 정보 제공 애플리케이션의 서비스 흐름도는 그림 3과 같다.

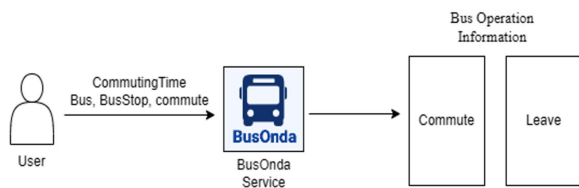


Fig. 3. Service Flow

출·퇴근 버스 운행 정보 제공 애플리케이션은 사용자로부터 출·퇴근 시간 및 즐겨찾기 할 버스와 정류소를 입력받아 지정한다. 그리고 출·퇴근 시간에 맞추어 해당 버스 도착 정보를 제공할 수 있다.

#### 2. Development Environment

출·퇴근 버스 운행 정보 제공 애플리케이션의 개발 환경은 그림 4와 같다.

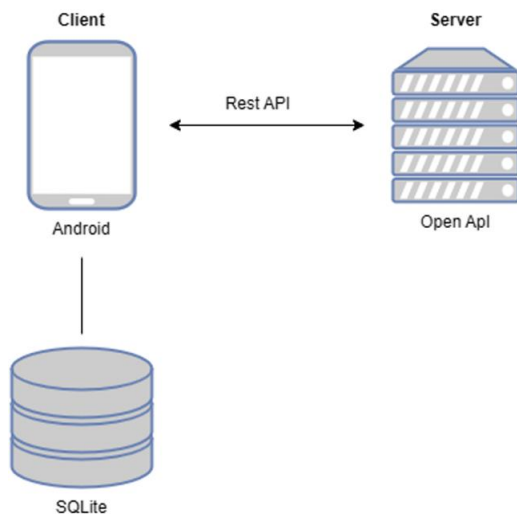


Fig. 4. Development Environment

그림 4의 클라이언트는 Android 앱으로 구현하며 인천광역시 Open API를 사용하여 인천 버스 정류소 및 버스 도착 정보에 대한 정보를 가져와 출력하는 기능을 제공한다. 또한 SQLite 데이터베이스를 사용하여 즐겨찾기 한 데이터를 저장하고 관리하는 기능을 한다.

### 3. Database Table Design

본 논문에서 설계한 Favorite DB 테이블의 column name은 표 4와 같다.

Table 4. Favorite DB Table Design

Column Name	Description
bstop_id	Bus Stop ID
bstop_nm	Bus Stop Name
route_id	Route ID
route_no	Route Number
commute	The State Of Commuting

Open API를 통해 버스 도착 정보를 추출하기 위해 즐겨찾기 한 정보를 요소로 하는 column으로 구성한다. Favorite 테이블에 총 5개의 column으로 구성되며, commute column에는 즐겨찾기 한 버스가 출근, 퇴근 중 어느 것에 해당하는지 저장한다.

### IV. The Implementation of Bus Operation Information Application

#### 1. Initial Commute Time Settings

출·퇴근 버스 운행 정보 제공 애플리케이션의 출·퇴근 시간 초기 설정 화면은 그림 5와 같다[12].



Fig. 5. Initial Commute Time Settings

초기 설정 화면에서는 출·퇴근 시간을 각각 설정할 수 있다. 입력받은 출·퇴근 시간은 SharedPreferences를 사용하여 CommuteTime이라는 파일에 키-값 쌍으로 저장되며, 입력한 시간대에 맞추어 출근화면과 퇴근화면 중 하나가 메인화면으로 설정된다. 완료 버튼을 클릭하면 설정

한 시간대에 맞는 메인화면으로 이동하도록 구현한 소스 코드는 그림 6과 같다.

```
@Override
public void onClick(View v) {

    // getSharedPreferences("파일이름", '모드')
    SharedPreferences sharedPreferences = getSharedPreferences("CommuteTime", Context.MODE_PRIVATE);

    // Editor를 preferences에 쓰기위해 연결
    SharedPreferences.Editor editor = sharedPreferences.edit();

    // putInt(KEY(String), VALUE(Int))
    editor.putInt("startHour", Integer.parseInt(edtStartHour.getText().toString()));
    editor.putInt("startMin", Integer.parseInt(edtStartMin.getText().toString()));
    editor.putInt("endHour", Integer.parseInt(edtEndHour.getText().toString()));
    editor.putInt("endMin", Integer.parseInt(edtEndMin.getText().toString()));

    // 항상 commit & apply 를 해주어야 저장이 된다.
    editor.apply();
}
```

Fig. 6. Source code of Set Initial Commute Time

## 2. Set Commute Time

출·퇴근 버스 운행 정보 제공 애플리케이션의 출·퇴근시간 설정 화면은 그림 7과 같다.



Fig. 7. Set Commute Time

초기화면에서 설정 텍스트 뷰를 클릭하면 그림 7의 시간 설정 화면으로 이동할 수 있다. 그림 7의 화면에서는 초기에 설정했던 출·퇴근 시간을 확인할 수 있고, SharedPreferences를 사용하여 출·퇴근 시간을 각각 다시 설정할 수 있다. 이러한 기능을 구현한 소스 코드는 그림 8과 같다.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_commute_setup);
    /*...*/
    // getSharedPreferences("파일이름", '모드')
    SharedPreferences sharedPreferences = getSharedPreferences("CommuteTime", Context.MODE_PRIVATE);

    // 저장한 값 가져오기
    int startHour = sharedPreferences.getInt("startHour", -1);
    int startMinute = sharedPreferences.getInt("startMin", -1);
    int endHour = sharedPreferences.getInt("endHour", -1);
    int endMinute = sharedPreferences.getInt("endMin", -1);

    // 저장한 값 세팅하기
    edtStartHour.setText(Integer.toString(startHour));
    edtStartMin.setText(Integer.toString(startMinute));
    edtEndHour.setText(Integer.toString(endHour));
    edtEndMin.setText(Integer.toString(endMinute));
}
```

Fig. 8. Source code of Get Commute Time

## 3. Search Bus Stop

출·퇴근 버스 운행 정보 제공 애플리케이션의 버스 정류소 검색 화면은 그림 9와 같다.

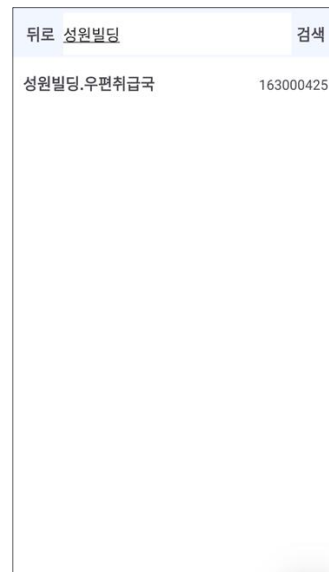


Fig. 9. Search Bus Stop

그림 9에서 검색하고 싶은 버스 정류소 이름을 입력하고 검색을 클릭하면 GetBusStationNmList 메소드를 사용하여 해당하는 버스 정류소 이름과 ID를 추출하여 리사이클러 뷰에 출력한다. 이 기능을 구현한 소스 코드는 그림 10과 같다.

```
protected void onPostExecute(String result) {
    /*...*/
    while (eventType != XmlPullParser.END_DOCUMENT){
        if(eventType == XmlPullParser.START_DOCUMENT){
            ;
        } else if (eventType == XmlPullParser.START_TAG) {
            String tag_name = xpp.getName();
            if(tag_name.equals("resultCode")) bSet_ResultCode = true;
            if(tag_name.equals("BSTOPNM")) bSet_BstopNm = true;
            if(tag_name.equals("BSTOPID")) bSet_BstopId = true;
        } else if (eventType == XmlPullParser.TEXT) {
            if(bSet_ResultCode){
                strResultCode = xpp.getText();
                bSet_ResultCode = false;
            }
            if(strResultCode.equals("0")){
                if(bSet_BstopNm) {
                    strBstopNm = xpp.getText();
                    bSet_BstopNm = false;
                    itemList.add(new Search(strBstopNm, strBstopId));
                }
                if(bSet_BstopId) {
                    strBstopId = xpp.getText();
                    bSet_BstopId = false;
                }
            }
        }
    }
}
```

Fig. 10. Source code of Search Bus Stop

그림 9의 버스 정류소 검색 결과에서 해당 버스 정류소를 클릭하면 추출한 정보를 가지고 버스와 버스 정류소 즐겨찾기 화면으로 이동한다.

#### 4. Favorite Bus Stop and Bus

출·퇴근 버스 운행 정보 제공 애플리케이션의 버스와 버스 정류소 즐겨찾기 화면은 그림 11과 같다.



Fig. 11. Favorite Bus Stop and Bus

그림 11의 즐겨찾기 화면에서는 검색한 정류소에 정차하는 버스 번호를 getBusStationViaRouteList 메소드를 사용하여 노선의 ID와 번호를 추출한다. 이 기능을 구현한 소스 코드는 그림 12과 같다.

```
protected void onPostExecute(String result) {
    /*...*/
    while (eventType != XmlPullParser.END_DOCUMENT){
        if(eventType == XmlPullParser.START_DOCUMENT){
            ;
        } else if (eventType == XmlPullParser.START_TAG) {
            String tag_name = xpp.getName();
            if(tag_name.equals("resultCode")) bSet_ResultCode = true;
            if(tag_name.equals("ROUTEID")) bSet_RouteId = true;
            if(tag_name.equals("ROUTENO")) bSet_RouteNo = true;
        } else if (eventType == XmlPullParser.TEXT) {
            if(bSet_ResultCode){
                strResultCode = xpp.getText();
                bSet_ResultCode = false;
            }
            if(strResultCode.equals("0")){
                if(bSet_RouteNo) {
                    strRouteNo = xpp.getText();
                    bSet_RouteNo = false;
                    itemList.add(new Bookmark(strRouteId, strRouteNo, "출근", "퇴근"));
                }
                if(bSet_RouteId) {
                    strRouteId = xpp.getText();
                    bSet_RouteId = false;
                }
            }
        }
    }
}
```

Fig. 12. Get Route ID and Number

추출한 노선의 번호와 출근과 퇴근 텍스트를 출력한다. 즐겨찾기 하고자 하는 버스의 출근과 퇴근 텍스트 중 하나를 클릭하여 출·퇴근을 설정한다. 출·퇴근 설정 후, 정류소 ID, 정류소 이름, 노선 ID, 노선번호, 출퇴근 상태를 SQLite 데이터베이스에 저장하는 소스 코드는 그림 13과 같다.

```
adapter.setOnItemClickListener((item, clickedTextView) -> {
    /*...*/

    if (!commuteType.isEmpty()) {
        Favorite favorite = new Favorite();
        favorite.setRoute_id(routeId);
        favorite.setBstop_nm(strBstopNm);
        favorite.setRoute_no(routeNo);
        favorite.setBstop_id(bstopId);
        favorite.setCommute(commuteType);

        // 데이터베이스에 저장 (별도의 AsyncTask 사용)
        new InsertFavoriteTask(favoriteDao).execute(favorite);
    }
    /*...*/
});
```

Fig. 13. Save Favorite Contents to Database

#### 5. Provide Bus Arrival Information

출·퇴근 버스 운행 정보 제공 애플리케이션의 출퇴근 버스 도착 정보 제공 화면은 그림 14와 같다.

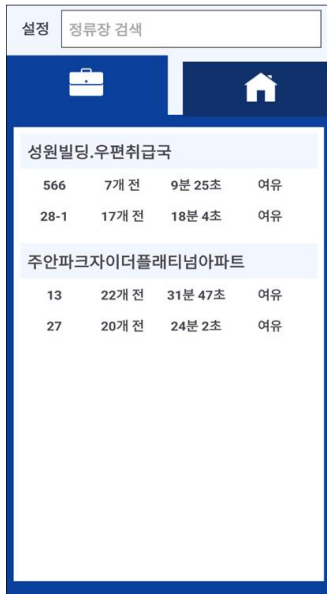


Fig. 14. Provide Bus Arrival Information

getAllRouteBusArrivalList 메소드를 사용하면 정류소 이름과 해당 정류소에 정차하는 버스 번호, 버스 위치, 도착예정시간, 혼잡도 등의 도착 정보를 확인할 수 있다.

```
protected void onPostExecute(String result) {
    //...
    while (eventType != XmlPullParser.END_DOCUMENT){
        if(eventType == XmlPullParser.START_DOCUMENT){
            ;
        } else if (eventType == XmlPullParser.START_TAG) {
            String tag_name = xpp.getName();
            if(tag_name.equals("resultCode")) bSet_ResultCode = true;
            if(tag_name.equals("ROUTEID")) bSet_RouteId = true;
            if(tag_name.equals("REST_STOP_COUNT")) bSet_RStopCnt = true;
            if(tag_name.equals("ARRIVALESTIMATIME")) bSet_ArrTime = true;
            if(tag_name.equals("CONGESTION")) bSet_Congestion = true;
        } else if (eventType == XmlPullParser.TEXT) {
            if(bSet_ResultCode){
                strResultCode = xpp.getText();
                bSet_ResultCode = false;
            }
            if(strResultCode.equals("0")){
                if(bSet_RouteId) {
                    strRouteId = xpp.getText();
                    bSet_RouteId = false;
                }

                // DB에서 가져온 리스트에 존재하는 노선인 경우에만 리스트에 저장
                for(Favorite fav : favList){
                    if(fav.getRoute_id().equals(strRouteId)){
                        etaList.add(new ETA(fav.getRoute_no(), strRStopCnt, strArrTime, strCongestion));
                    }
                }
            }
        }
    }
}
```

Fig. 15. Get Bus Arrival Information

## V. Conclusions

본 논문에서는 안드로이드 플랫폼 기반의 버스 운행 정보를 제공하는 애플리케이션을 설계하고 구현한다. 이 애플리케이션은 공공데이터포털에서 제공하는 Open API를 이용하여 인천광역시를 운행하는 버스의 운행 정보를 실시간으로 제공한다. 버스 운행 정보는 정류소 이름, 정류소에 정차하는 버스 목록, 버스 위치, 버스 도착 시간 등을 포함한다. 이 애플리케이션을 사용하는 사용자는 자신의 출·퇴근 시간을 설정하고, 해당 시간에 이용할 수 있는 정류소와 노선 버스 정보를 즐겨찾기에 등록함으로써 맞춤형으로 제공받을 수 있다. 또한, 맞춤형 버스 운행 정보는 앱의 메인화면에 출력함으로써 사용자가 직관적으로 확인할 수 있다는 것이 장점이다.

## REFERENCES

- [1] Ministry of Land, Infrastructure and Transport, 2023 National Land and Transport Statistical Yearbook, pp. 681, 2023
- [2] Incheon Metropolitan City, Incheon Metropolitan City Social Survey, 2023
- [3] Incheon Metropolitan City, Status of City Bus Stops, 2024
- [4] Incheon Metropolitan City, Status of Bus Arrival Information Systems, 2024
- [5] <https://play.google.com/store/apps/details?id=kr.or.gbushybrid&hl=ko>
- [6] <https://play.google.com/store/apps/details?id=com.ulsanbus.city2023&hl=ko>
- [7] <https://www.data.go.kr/ugs/selectPublicDataUseGuideView.do>
- [8] <https://www.data.go.kr/tcs/dss/selectApiDataDetailView.do?publicDataPk=15058487>
- [9] <https://www.data.go.kr/tcs/dss/selectApiDataDetailView.do?publicDataPk=15059084>
- [10] <https://developer.android.com/training/data-storage/room?hl=ko#groovy>
- [11] <https://developer.android.com/training/data-storage/shared-preferences?hl=ko>
- [12] W. J. Lee, S. Y. Lee, "A Design and Implementation of a Bus Operation Information Application," Proceedings of the 32nd Summer Conference on Korea Society of Computer and Information(ISSN 2005-1344), Vol. 32, No. 2, pp. 239-240, July 2024

## Authors



Seo Young Lee received the Associate Degree in Computer Science and Engineering from Inha Technical College, Korea, in 2024. Her research interests include Mobile Programming.



Won Joo Lee received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Hanyang University, Korea, in 1989, 1991 and 2004, respectively. Dr. Lee joined the faculty of the Department of

Computer Science and Engineering at Inha Technical College, Incheon, Korea, in 2008, where he has served as the Director of the Department of Computer Science and Engineering. He is currently a Professor in the Department of Computer Science and Engineering, Inha Technical College. He has also served as the president of The Korean Society of Computer Information. He is interested in parallel computing, internet and mobile computing, and cloud computing, data science, artificial intelligence.