

# **IOT장비를 활용한 서울시 통합 버스서비스 및 빅데이터 의사결정 시스템**

**2018년 06월 04일**

커넥티드 카를 위한 IoT 전문가 과정(NCS)

Aeris(팀명)

손애리

박준하

조용원

곽지훈

# 목 차

<b>1. 프로젝트 개요</b>	<b>1</b>
1.1 프로젝트 기획 배경 및 목표	1
1.2 구성원 및 역할	2
1.3 프로젝트 추진 일정	3
<b>2. 프로젝트 현황</b>	<b>4</b>
2.1 시장 분석	4
2.2 경쟁 제품 장단점 분석	6
2.3 차별화 핵심 전략 기술	9
<b>3. 프로젝트 개발 결과</b>	<b>9</b>
3.1 주요 기능 및 상세 동작	9
3.2 사용 디바이스 및 목표성능 or 주요 다이어그램 및 분석서	13
3.3 핵심 알고리즘	16
3.4 주요 동작	19
<b>4. 기대 효과</b>	<b>21</b>
4.1 향후 개선 사항	21
4.2 기대 효과	23
<b>5. 개발 후기</b>	<b>24</b>
<b>6. 강사 의견</b>	<b>25</b>

# 1. 프로젝트 개요 [Bus Information System]

## 1.1 프로젝트 기획 배경 및 목표

온라인 정보통신 기술이 오프라인 산업에 적용되면서, 4차 산업혁명이 일어나고 있다. 정보통신기술과 융합된 산업이 각광을 받고, 세계는 산업에서부터 일상전반까지 스마트 플랫폼으로 변모하고 있는데 그 중에서도 가장 가시적인 변화는 생활전반에 첨단정보통신기술(ICT)을 적용한 스마트 시티에 대한 세계의 관심이다.

국내외 기업들은 첨단정보통신기술을 활용하여, 빅데이터를 구축할 수 있는 기반을 만들기에 집중하고 있고, 각국 정부 역시 정책적, 예산적 지원을 꾸준히 늘려가고 있다. 국내에서도 이러한 노력의 일환으로 특정지역을 지정하여 스마트 시티화하고, 도로의 상황을 반영하는 시스템을 구축하고자 하는 등의 노력을 기울이고 있다.

이러한 국내외 시장의 상황을 반영하여, 우리는 스마트시티로의 가장 기반이 되는 통신, '공공와이파이의 확대' 라는 화두에 초점을 맞추고, 아직 이를 활용할 만한 플랫폼이 부재한다고 판단하여, IOT장비를 활용한 서울시 통합버스서비스 및 빅데이터 의사결정 시스템(BIS)이라는 주제로 서울시 버스 내부의 설치 된 공공와이파이를 활용한 플랫폼 서비스를 기획하게 되었다.

우리는 IOT장비를 활용한 유기적 분석시스템을 개발하여, 대중교통, 그 중에서도 버스의 승객 및 관리자, 운전자의 의사결정과정 전반에 도움을 주고자 한다.

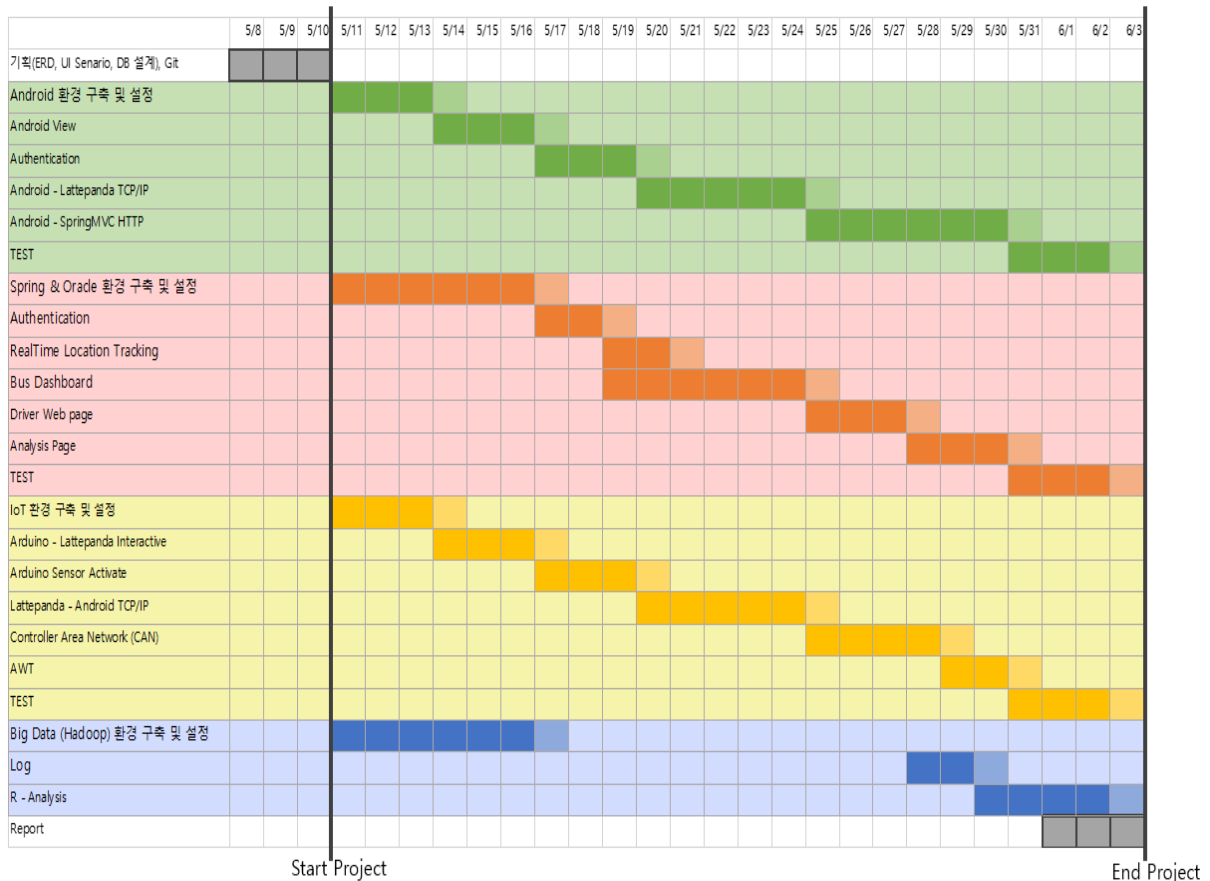
## 1.2 구성원 및 역할

이름	전공	역할	구현 부분
박준하	국제지역문화학과	팀장	프로젝트 관리 Android App 개발
곽지훈	경영학과	팀원	Web Server 개발 Web Front 개발 데이터 분석
조용원	정보통신공학과	팀원	IOT 장비 개발 CAN protocol

손애리	경영학과	팀원	UI 기획 및 개발 게시판 기능 구현
-----	------	----	-------------------------

### 1.3 프로젝트 추진 일정

구분	기간	활동	비고
사전 기획	5/8(화) ~ 5/10 (목)	프로젝트 기획 및 아키텍 처 설계	
	5/11(금) ~ 5/14 (월)	환경설정	IoT / Spring / Oracle / Hadoop, Hive
	5/14 (월)	팀별 착수 보고 (Kick-off)	
PJT 수행 / 완료	5/15(화) ~ 6/2(토)	프로젝트 수행	
	11/7(화)	팀별 중간 보고 (To-Be Workshop)	
	6/4(월)	팀별 최종 발표 (구축 완 료 보고)	최우수팀 선발



## 2. 프로젝트 현황

### 2.1 시장분석

서울시 통합 버스 서비스 및 빅데이터 의사결정 시스템 SBIS 는 ITS 산업, 실질적으로는 정보통신산업에 기반을 두고 있는데, ITS(Intelligent Transport System)시장은 전 세계적으로 매년, 성장에 성장을 거듭하고 있는 주목받고 있는 시장이다. 세계시장 2015 년 기준 약 22 조원에서 연평균 12.7% 성장해 2024 년 약 63 조원에 달할 것으로 전망되며, 각국의 정책적 지원과 스마트폰의 보급 확대, 개발도상국의 ITS 투자 활성화에 탄력을 받아 더욱 빠르게 성장하고 있다. 또한, ITS 서비스에 대한 사회적 인식 변화와 기술발전으로 교통안전과 환경보호, 주차관리, 자율주행(자동운전), 자동차 텔레매틱스 분야의 비중도 점차 확대되고 있으며 개발도상국들의 경우, 급속한 도시화 문제 개선을 위한 스마트시티 구축 열풍이 불고 있어 효과적인 대중교통 운영을 위한 ITS 도입이 확대되고 있다.

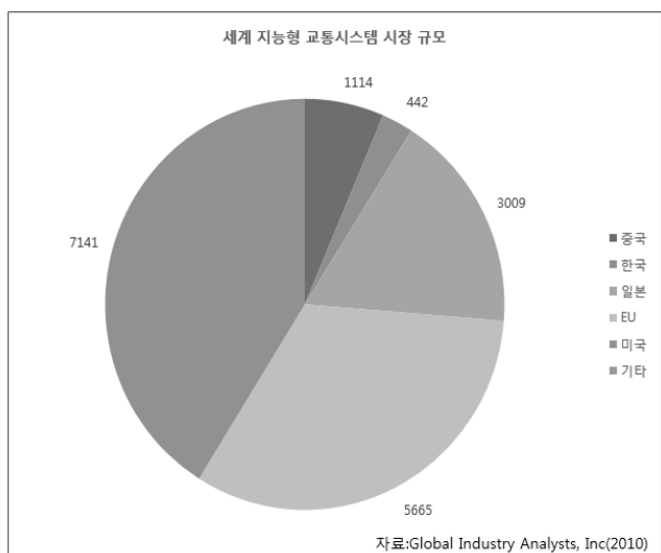
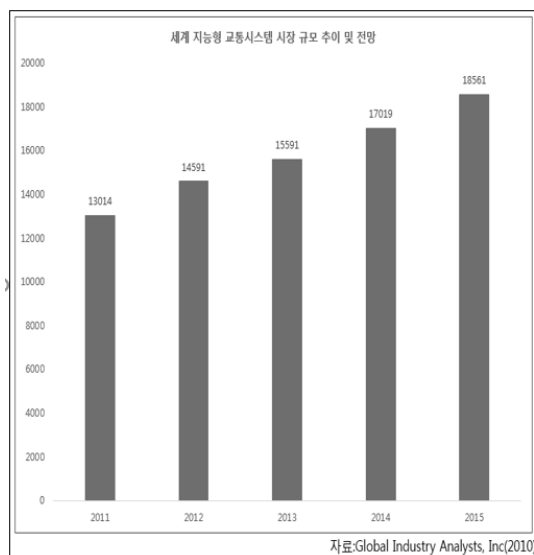
국내 ITS(Intelligent Transport System)시장을 살펴보자면, 규모는 2014년 기준 약 4억 달러 규모로 추정되고 지속적으로 확대되고 있으며, 2011년 2억 9,400만 달러에서 2015년 약 4억 4,200만 달러 규모로 연평균 10.7%의 성장률을 보이고 있다. 안전하고 편리한 이동 수단의 수요가 증대되고 첨단 기술과 융합이 가속화되면서 지능형 교통시스템의 시장규모가 꾸준히 확대되고 있다. 글로벌 ITS 시장은 미국, 일본, EU 등 선진국들이 주도할 것으로 전망되지만

중국이 시장점유율을 확대하면서, 새로운 리더로 발돋움할 전망이다(한국인터넷진흥원, 2012).

국내의 정책적 동향은 2020년까지 모든 4차로 이상의 도로를 대상(전국 도로의 약 30%)으로 실시간 도로 관리, 이용자 맞춤형 대중교통정보 등을 제공하는 지능형 교통시스템 구축이 본격화될 것으로 보인다. '생활형 스마트 도로교통 구현'의 비전 아래, 지속가능한 ITS 정책 추진과 관련 산업 성장을 통한 '안전한 도로, 편리한 도로, 고효율 녹색 도로'의 구축을 목표로 하고 있다.

주행환경, 도로환경 등을 자동 인식하여 운전자에게 제공하는 등 승용차 자동제어 및 안전운행을 지원하는 V2X 기술 기반 지능형 자동차 도로의 개발 및 구축을 확대하고 있다(국토교통부, 2012).

이렇듯, 시장자체가 아직 성장하는 단계이고, 시장을 점유하는 큰 축은 있지만, 사업자체가 확장성이 매우 크고 고정되어 있는 시장이 아니기 때문에, BIS의 미래는 밝다.



## 2.2 경쟁 제품 장단점 분석

기존에 존재하는 대중교통 어플리케이션이 있으나, BIS는 기존의 대중교통 어플리케이션이 가지고 있는 1차적인 기능(거리 정보, 버스도착정보, 버스정류장 정보, 승차권 결제 기능)에 집중하지 않고, ITS(Intelligent Transport System)적인 기능들(버스 내부에서 일어나는 상황들의 데이터, 승객 데이터)에 보다 중점을 두어 훗날 서비스가 ITS적인 확장성을 가질 수 있도록 기획하였다.

이처럼, 현재 대중교통 어플리케이션 시장에는 비슷한 기능의 서비스가 존재하지 않으며, 우리는 경쟁 서비스가 등장하기 이전에 얼마나 빠르게 유저들을 모으는 지가 사업의 성패를 결정할 것으로 보고 있다.

## 2.3 차별화 핵심 전략 기술

위에서도 말했듯이, BIS 는 IOT 장비를 통하여, 수집한 데이터를 분석, 버스의 승객 및 운전자, 관리자의 의사결정 과정 전반에 도움을 주는 것을 목표로 하고 있다.

우리는 IOT 장비(아두이노, CAN)를 통하여 받아온 정보(압력 데이터, 온도 데이터, 적외선 데이터)를 바탕으로, 안드로이드 화면을 통해서, 하차 벨 조작, 온도조절요청, 치한 신고 등의 편의를 탑승자들에게 제공하고 승객들의 요청과 반응, 버스의 상황(온도, 습도, 고장 빈도 등의 데이터)을 분석하여, 승객과 운전자, 관리자에게 가시적으로 유용한 정보를 제공할 것이다.

기존의 대중교통과 관련된 어플리케이션들이 일방적으로 버스의 위치를 제공하는 역할을 했다면, BIS 는 사용자들의 정보를 바탕으로, 분석하여 사용자들에게 도움이 될 정보를 창출해 내고, 일방적인 데이터 전달에서 벗어나서 고객이 직접 대중교통의 옵션을 선택할 수 있으며, 데이터를 바탕으로, 승객, 운전자, 관리자의 소통이 가능한 플랫폼을 만든다는 점에서 타 대중교통 어플리케이션과 차별화 되었다.

## 3. 프로젝트 개발 결과 [대제목 : 16pt / 맑은 고딕]

### 3.1 주요 기능 및 상세 동작 [소제목 : 12pt / 맑은 고딕]

#### 3.1.1 주요 기능

##### Passenger

- Smart Phone App 으로 고객 편의성 향상
- 고객 개인 디바이스를 통한 원격 하차 벨 제어
- 실시간 요청 기능 - 온도 제어, 안전 운전, 범죄 발생
- 실시간 버스 상태 정보 및 버스 기사 정보 확인
- 고객의 소리 게시판 운영
- 차량 내 WIFI 접속만으로 해당 차량 맞춤형 서비스 제공

##### Driver

- 드라이버의 차량 운용 편의성 향상
- 전 요청의 전산화를 통해 오직 차량 운행에 집중 가능
- 운행 일정 관리
- 대 고객, 대 관리자와의 의사소통 편의 증가
- 버스 제원 데이터 관리

- 실시간 차량 데이터 확인 (온 습도, CAN data 등)
- 승객 요청 정보 및 메시지

#### Administer

- 관리자의 차량 및 기사 관리
- 운용중인 모든 버스 및 기사 관리
- 운행 차량 현황 데이터(온, 습도, 위치, CAN data 등) 실시간 대시보드 제공
- 실시간 차량 위치 추적 시스템
- CAN data 실시간 분석(이상치 자동 필터링을 통한 차량 이상 예측)
- 승객 요청 실시간 처리
- 관리자에 의한 차량 제어
- 신규 버스 및 기사 등록

### 3.1.2 상세 동작

#### • IOT

##### 하차 벨 제어

Communication Module 이 서버로 받은 하차 벨 요청에 따라 CAN data 형식에 맞춰 메시지를 생성한다. 이후 CAN bus 를 통해 ECU 로 제어 요청을 한다. ECU 에서는 요청 받은 CAN data 를 분석해, 하차 벨 요청임을 확인하고, Serial Writer 를 통해 아두이노에 Piezo 동작을 요청한다. 이때 piezo 동작을 delay 로 구현하지 않고, 내장함수인 millis()를 이용해 구현했다. 이는 한개의 loop 인 Arduino 의 동작 특성에 따라 piezo 동작 시 타 기능이 동작하지 못하게 되는 현상을 피하기 위해서이다. 하차 벨 요청 시 요청 시각을 체크하고, loop 를 돌면서 현재 millis() 시각과의 차이가 특정 시간보다 작을 때만 piezo 를 동작하고, 그렇지 않으면 piezo 동작을 멈추는 구조이다.

##### 4\*4 keypad (좌석 정보)

좌석정보를 나타낼 수 있는 4\*4 keypad 는 패드 입력 시 Serial 통신을 통해 입력 받은 좌석정보의 String 값을 ECU 로 전달한다. ECU 에서는 Serial Reader 를 통해서 좌석 정보를 확인하고, 이를 AWT 를 통해 화면에 출력한다. 또한, 이 정보를 CAN data 로 정제하여 Communication Module 로 전달하고, Communication Module 은 TCP/IP 통신을 통해서 서버로 정보를 전달한다.

##### 조이스틱 (차량 속도)



조이스틱은 0~1023 까지의 범위를 갖는다. 기준 값이 512 였기 때문에 Arduino 에서 아날로그 값을 읽은 후 512 를 감소해 기준 값을 0 으로 변경시켜주었다. 또한 작은 움직임에도 값이 크게 변동하는 것을 방지하기 위해 아날로그 값에서 4 로 나눈 몫을 실제 속도로 지정해주었다.

#### 온습도 센서

DHT11 센서를 사용하기 위해서는 DHT11 라이브러리가 필요하다. 온습도 정보 요청을 받으면 Arduino 는 Serial 통신을 통해 ECU 로 온도와 습도를 전달하고, ECU 는 온습도를 CAN data 로 정제하고 CAN bus 를 통해 Communication Module 로 정보를 전달한다.

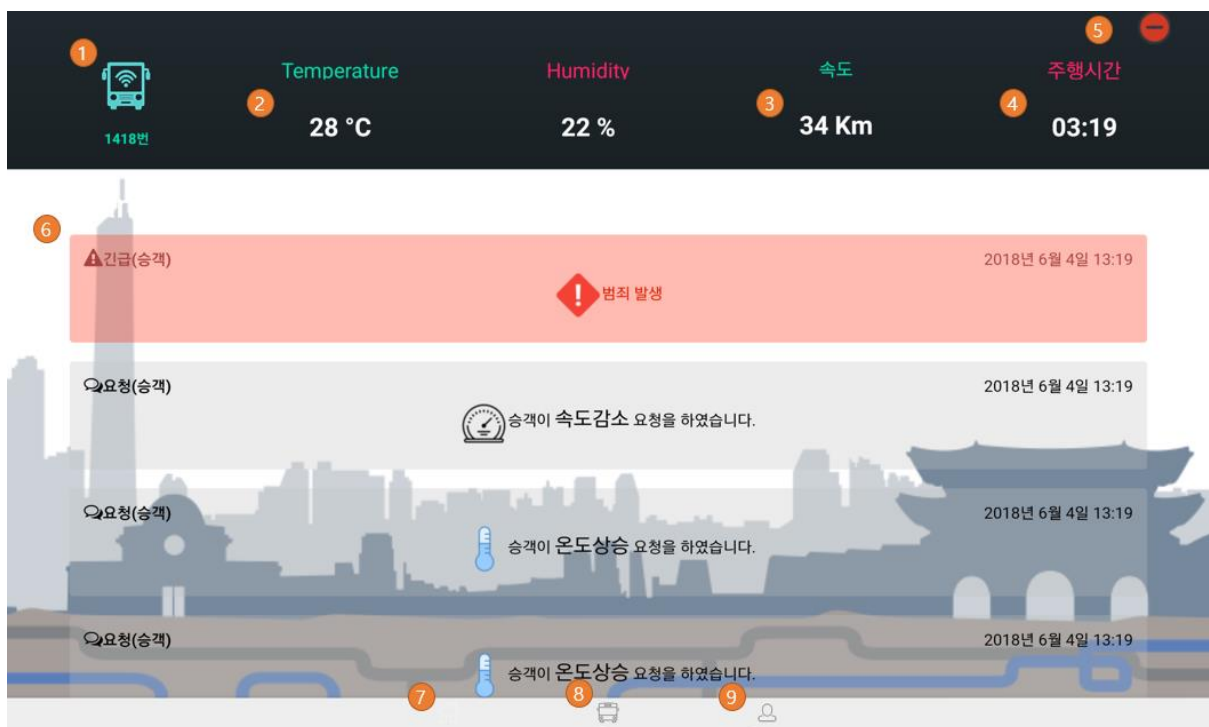
### • Driver App

#### 1) Log in

미리 발급 받은 Driver ID 와 Password 를 입력하면 AsyncTask 를 통해 Web server 로 로그인 인증을 수행한다. 로그인이 성공하면 기사 개인 정보와 버스 정보를 Web server 에서 응답 받아 Shared Preferences 를 통해 기기에 저장한다.

로그아웃(운행 종료)을 하기 전까지 Shared Preferences 를 활용해 자동 로그인이 가능하게 했다.

#### 2) Main



1. Log in 을 통해 Web server 로부터 응답 받은 버스 번호를 Textview 로 표시한다.
2. IoT 장비로부터 TCP/IP 통신으로 받은 실시간 온도 습도 data 를 TimeTask 를 활용해 주기적으로 업데이트하여 Textview 로 보여준다.

3. IoT 장비로부터 TCP/IP 통신으로 받은 실시간 속도 data 를 TimeTask 를 활용해 주기적으로 업데이트하여 TimeTask 로 보여준다.
4. Chronometer 를 활용하여 주행시간을 카운트한다.
5. Log out 기능을 하는 Image Button 으로 Shared Preferences 에 있는 정보를 초기화하고 기사의 운행 정보를 AsyncTask 를 활용해 Http 통신으로 웹서버에 전송한다.
6. 실시간으로 승객 개인용 스마트 디바이스를 통해 TCP/IP 로 보낸 요청 정보들(온도 상승, 온도 하강, 안전 운전, 범죄 발생)을 서버로 전송하고 Web view 를 통해 보여준다.
7. 메인 화면인 Home Fragment 로 이동하는 버튼이다.
8. 버스의 실시간 CAN data 와 차량 이상 유무를 파악할 수 있는 Bus Fragment 로 이동하는 버튼이다.
9. 기사 개인 정보와 고객에 대한 피드백을 확인할 수 있는 Driver Fragment 로 이동하는 버튼이다.

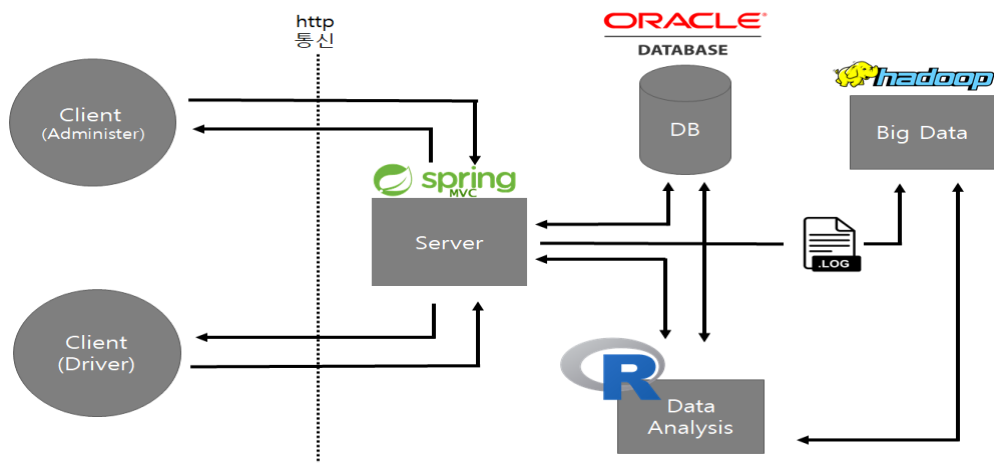
#### • Passenger App



1. App 이 실행되고 Driver App TCP/IP Socket server 에 접속하게 되면 Driver App 에 저장된 driver data 와 bus data 를 받는다. 메인 화면에는 driver 이름과 버스 번호, 차량 등록 번호를 전송 받는다.
2. Driver App 에서 TCP/IP 통신을 통해 주기적으로 속도 data 와 온도 습도 data 를 전송 받아 Textview 에 업데이트 한다.
3. 지금 내 위치 버튼을 누르면 Map Fragment 로 이동하고 실시간 GPS 정보에 따른 현재 위치를 Google Map 에 Tracking 하여 보여준다.
4. 하차 벨 버튼을 누르면 TCP/IP 통신을 통해 Driver App 으로 하차벨 요청 데이터를 보내고 Driver App 은 다시 TCP/IP 통신으로 ECU 에 하차 벨 제어 요청을 전달한다.

5. 추워요, 더워요, 빨라요, 신고하기 버튼을 누르면 TCP/IP 통신을 통해 Driver App 으로 요청 데이터를 보내고 Driver App 은 Http 통신으로 Web Server 에 고객 요청을 전송한다.
6. 버스와 기사님에 대한 피드백을 남길 수 있는 고객의 소리 페이지로 이동한다.
7. 개인 정보를 수정할 수 있는 마이페이지 화면으로 이동한다.

#### • Web Page



#### 웹 서버 주요 개발목표

- 대한민국 공공기관 웹 서비스에서 사용이 권장되는 스프링 환경의 MVC 프레임워크로 웹 서버 개발
- 관리자 중심 맞춤형 차량 정보 제공 전략 및 시스템 개발
- 상시 대응이 가능하도록 실시간 데이터 수신 및 발신
- 다수의 차량에서 발생된 상태 정보의 실시간 수집 및 분석
- 차량 상태 이상 예측 시나리오 개발
- 웹(http) 통신을 통한 관리자의 차량 역 제어
- 유동적인 운행 계획에 대응 가능하도록 개발
- 사용자 중심의 웹 UI 개발
- 자동화된 데이터 수집 및 저장 분석 시스템 개발
- 타 서비스로 확장 가능한 관리자용 웹 서비스 개발

#### 상세 기능 및 동작

##### 1. 관리자 인증

관리자 계정 로그인

ID

ID 입력

Password

비밀번호 입력

☐ Remember Password

Login

☞ ID,PWD를 잊으셨나요?

- 1) 관리자 계정은 보안상 DB 에서 직접 입력되어 추가 및 수정이 가능하며 HandlerInterceptor 를 상속받아 구현한 loginInterceptor 를 통해 정상적인 인증 없이 관리자 권한에 접근할 수 없도록 구현
- 2)

## 2. 운용 중인 전체 차량 및 기사 관리

1 관리자 관점의 메뉴 구성

2 운용 중인 전체 버스 현황(실시간)

3 버스 & 기사 등록

차량 고유번호(IDX)	버스 번호	차량 번호	운전자 명	현 위치(lat)	현 위치(lon)
2	15218	1AG1242	1	0	0
3	12118	12DG1242	1	0	0
5	1418	G12G1242	1	127.123	37.123
6	1418	D12G1242	1	127.123	37.123

Showing 1 to 4 of 4 entries

업데이트 시간: 2018년 6월 4일 1:28

차량 등록

버스 번호  
버스 번호 입력

차량 번호  
차량 번호 입력

차량 연식  
YYYY/MM

차량 연료 타입  
연료 타입 입력

버스 타입  
버스 타입 입력

관리자 ID  
ID 입력

관리자 PWD  
관리자 PWD

등록

### 1) 관리자 관점의 메뉴 구성

### 2) 운용 중인 전체 버스 현황

- 버스내 기 설치된 전용 디바이스를 통해 해당 기사가 로그인을 함과 동시에 운행시작으로 기준을 잡아 DB 에 업데이트, 이후 관리자는 실시간으로 업데이트된 정보를 확인 가능하도록 구현
- 이 기록은 로그로 저장돼 다수 기사들의 운행 일정 관리 가능

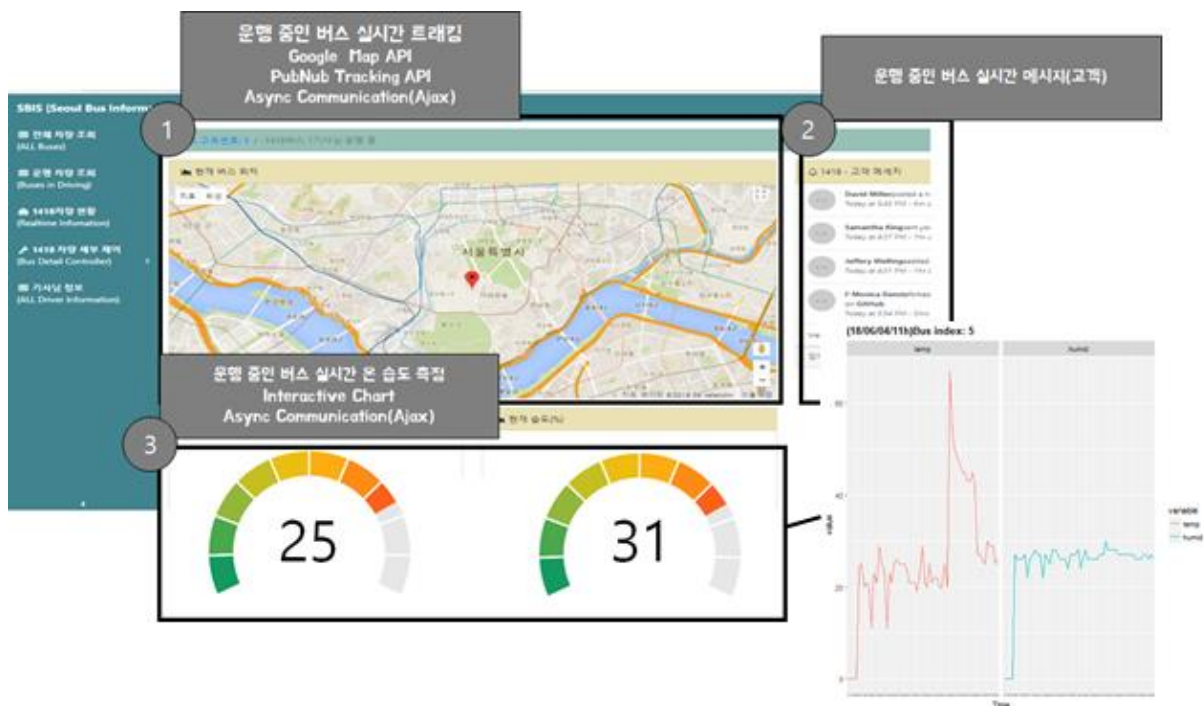
### 3) 신규 버스 및 기사 등록

- 신규 등록은 관리자 권한에서 진행되며 관리자 등록 시 특정 관리자의 재 인증이 필요

### 4) 서버 메모리에서 Instance 변수로 Data 유지

- AdminController Class 에 static 으로 선언된 HashMap 객체에 드라이버 운행 시작과 함께 동적으로 차량 상태 정보를 유지할 공간 할당 및 운행 종료와 함께 제거
- ClientController Class 에도 유사한 알고리즘을 적용하여 고객 개인 디바이스에서 발생된 instance 메시지 업데이트

### 3. 현재 운행 중인 차량 관리 -1



#### 1) 실시간 버스 Tracking System

- 차량내 디바이스로부터 3 초 마다 위치 정보를 수신 받아 운행 시작과 함께 동적으로 임시 유지공간으로 할당된 HashMap(key = 버스 고유번호, member variable) 의 value(List)에 단위 초마다 해당 차량 위치 정보 수정 및 유지
- 비동기 통신(ajax)을 단위 초마다 재귀호출하여 동적으로 업데이트된 데이터 지속 수신
- 관리자에 의해 요청된 차량 위치 시각화, Google Map API 이용
- PubNub Location Tracking API 를 통해 위치 변화에 따라 동적으로 맵 시각화

#### 2) 차량 내 고객 요청 메시지 수신

- 실시간 고객 요청 수신을 통한 관리자 의사결정 범위 확대
- 긴급 요청 발생 시 경고 창 발생

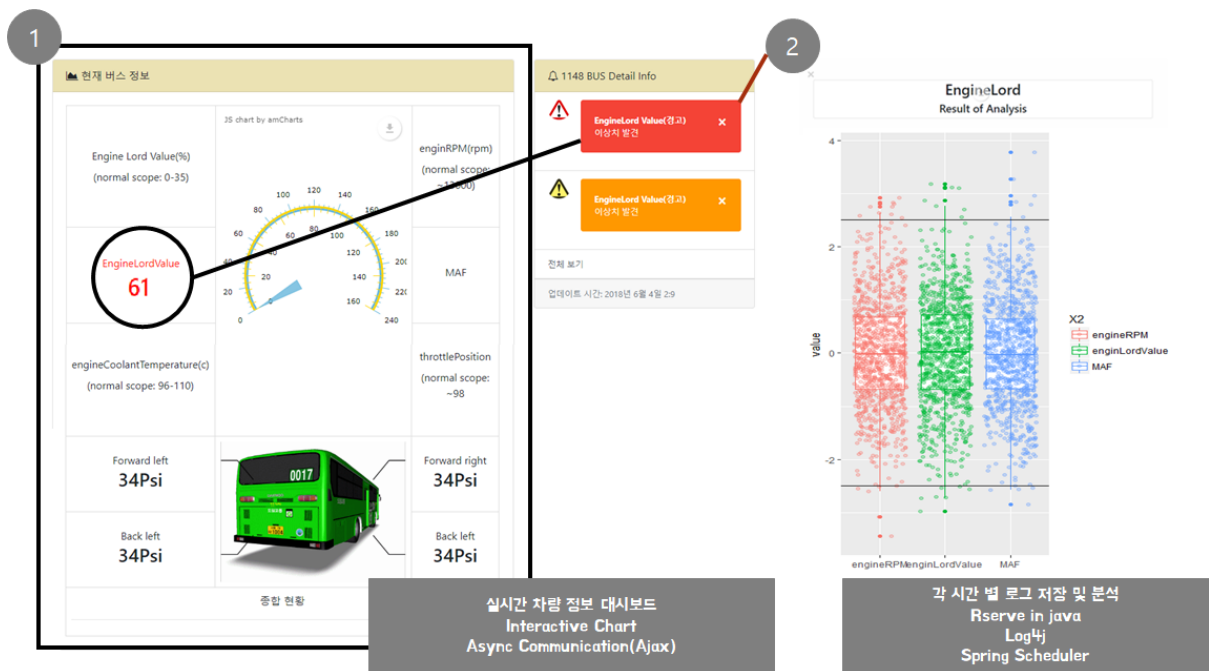
#### 3) 차량 내 온 습도 실시간 시각화

- 차량내 디바이스로부터 3 초 마다 온 습도 정보를 수신 받아 운행 시작과 함께 동적으로 임시 유지공간으로 할당된 HashMap(key = 버스 고유번호)전역변수 value(List)에 단위 초마다 해당 차량 온 습도 정보 수정 및 유지
- 비동기 통신(ajax) 재귀 호출로 지속적으로 서버로부터 수신 받은 데이터 Java script Interactive Chart (high chart)를 이용하여 시각화

#### 4) 관리자에 의한 차량 제어

- TCP/IP 로 연결된 차량 디바이스에 HTTP 통신을 통한 관리자 제어 요청

#### 4. 현재 운행 중인 차량 관리 -2



#### 1) 실시간 차량 상태 정보 업데이트 및 시각화 (Can Data 대시보드)

- 차량내 디바이스로부터 10 초 마다 차량 내부 상태 정보(can data)를 수신 받아 운행 시작과 함께 동적으로 임시 유지공간으로 할당된 HashMap(key = 버스 고유번호, member variable) value(List)에 단위 초마다 해당 차량 내부 상태 정보 수정 및 유지
- 비동기 통신(ajax) 재귀 호출로 지속적으로 서버로부터 수신 받은 데이터 Java script Interactive Chart (high chart)를 이용하여 시각화

#### 2) 서버에 기 입력된 데이터 별 정상치 범주를 기준, 이상치 필터링 및 경고 알림

- 이상치 경고 수준을 경고, 심각 수준으로 나누어 실시간으로 차량 상태 경고 수신
- 경고창 클릭 시, 최근 시간(1 시간 단위)의 로그 데이터 분석 시각화 결과물 제공

### 3.2 사용 디바이스 및 목표성능 or 주요 다이어그램 및 분석서

#### HardWare

LattePanda - ECU

Arduino Leonardo - ECU

센서 - piezo, led, 4\*4keypad, 조이스틱, 온습도센서

Galaxy Tab 10.1 – Bus driver Application

Galaxy S8 – Passenger Application

Galaxy Note5 – Passenger Application

#### SoftWare

	IoT	Android	Web	Big Data
OS	Windows 10	Windows 10	Windows 10	CentOS 7.0
IDE	Eclipse Oxygen Arduino Sketch	Android studio 3.0	Eclipse Oxygen	R Studio Eclipse Oxygen
Lang	JDK 1.8.0_171 Arduino	JDK 1.8.0_171	JDK 1.8.0_171	R 3.4.3 JDK 1.8.0_171
Server			Apache Tomcat v9.0 Spring 5.0.5 MyBatis	
Database		Oracle 11g SQL Developer	Oracle 11g SQL Developer	Hadoop HIVE MariaDB
View	AWT	Android XML	Bootstrap3	Interactive Chart

#### Software Architecture

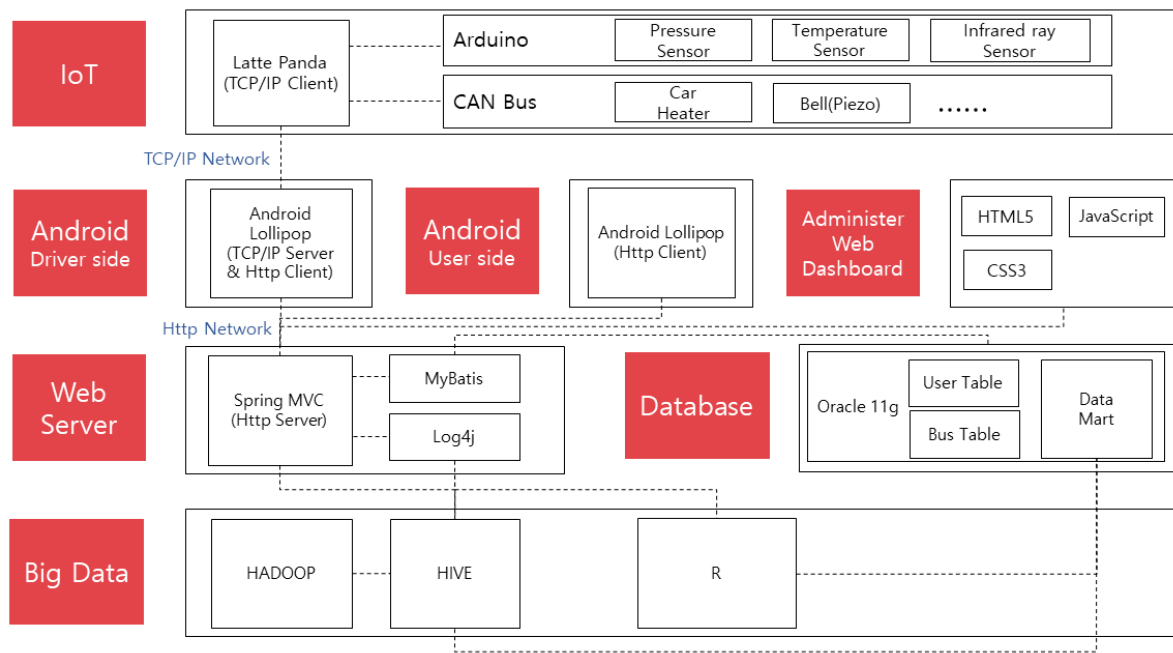


Figure 1

라떼판다(차량 ECU, 차량 통신 모듈)

FEATURES	LATTEPANDA
Processor	Intel Cherry Trail Quad Core 1.8Ghz
Operation System	Pre-installed Full Edition of Windows10
RAM	2GB
Graphics Capabilities	Intel Gen8
Video Supported	1080p 60 HEVC decode, H.265, VP8
Video Out	HDMI Disply Connector
Storage Capability	32GB
Power Supply	5V

차량 ECU 라떼판다는 Serial통신을 통해 각종 센서로부터 받은 값을 CAN data 형식으로 변환하여 CAN bus를 통해 차량 통신 모듈용 라떼판다로 정보를 전달한다. 이때 CAN data 형식은 OBD-II PID 프로토콜을 따른다. 차량 통신 모듈은 전달받은 CAN data를 정제하여 TCP/IP 서버로 소켓을 통해 전달한다. 마찬가지로 서버의 요청을 차량 통신 모듈이 CAN data로 변환하여 차량 ECU로 전달하고, 이를 통해 센서를 제어한다. Figure2은 IOT System Architecture 를 나타낸다.



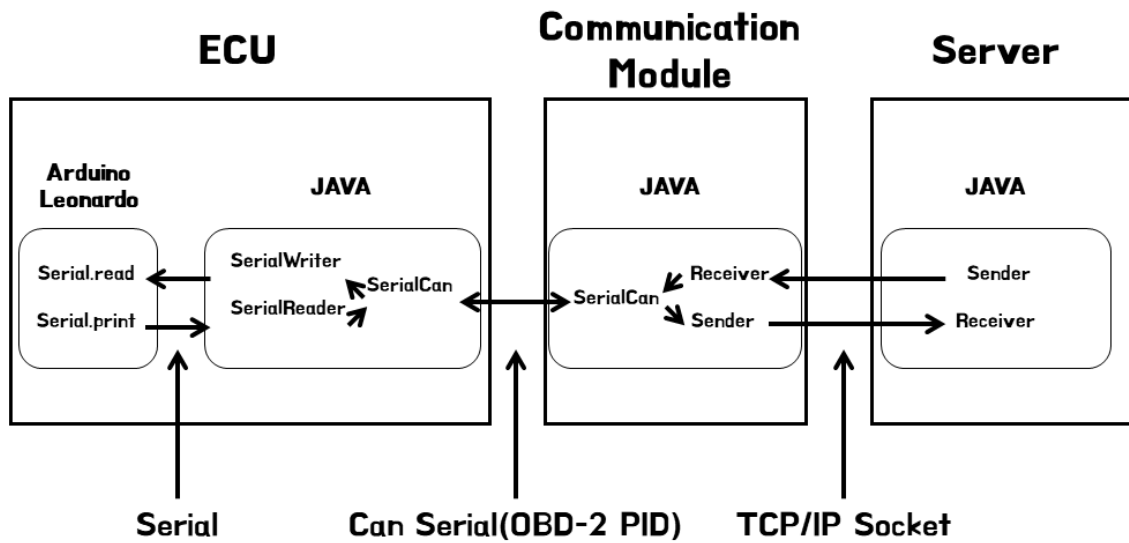


Figure 2

### 3.3 핵심 알고리즘

#### 1. OBD- II PID

- On Board Diagnostics Parameter ID 의 약자로 프로토콜 또는 코드를 의미함
- CAN 통신을 지원하는 차량의 대부분은 OBD- II PID 를 지원
- SAEJ1979 표준인 한 부분으로 1996 년부터 북미 모든 차량에 구현됨
- 고장 진단 장치가 차량 ECU 와의 데이터 교환을 위한 일종의 규약
- CAN data 는 ID 4byte, Data 8byte 로 구성

아래 표 1 은 주요 PID 종류와 관련 응답데이터 형식을 나타낸다. 표 1 의 공식에서 사용되는 알파벳 A,B,C,D 등은 수신된 데이터의 바이트 순서와 일치한다. 첫 번째 바이트는 A, 두 번째는 B, 세 번째는 C, 마지막으로 네 번째는 D 이다.

Mode	PID	수신크기	설명	Min	Max	단위	공식
00	00	4	PID Supported[01-20]	-	-	-	Bit encoded [A7..D0] == [PID 0x01..PID 0x20]
01	04	1	Calculated engine load value	0	100	%	$A \times 100 / 255$
01	05	1	Engine Coolant Temperature	-40	215	°C	$A - 40$
01	0C	2	Engine RPM	0	16383.75	Rpm	$((A \times 256) + B) / 4$
01	0D	1	Vehicle speed	0	255	Km/h	A

01	0E	1	Timing Advance	-64	63.5	Relative to #1 cylinder	A/2-64
01	0F	1	Intake air Temperature	-40	215	°C	A-40
01	10	2	MAF air flow rate	0	655.35	g/s	((A*256)+B)/100
01	11	1	Throttle Position	0	100	%	A*100/255

표 1

표 1의 Mode의 각 숫자 별로 의미하는 내용은 아래 표 2와 같다. ECU의 각종 정보 데이터를 얻는 경우는 대부분 현재 데이터 표시용 Mode 0x01만 이용되며, 차량의 고장진단 코드를 얻고자 하는 경우 Mode 0x03을 사용한다.

Mode	설명
0x01	Show current data
0x02	Show freeze frame data
0x03	Show stored Diagnostic Trouble Codes
0x04	Clear Diagnostic Trouble Codes and stored values
0x05	Test results, oxygen sensor monitoring (non CAN only)
0x06	Test results, other component/system monitoring (Test results, oxygen sensor monitoring for CAN only)
0x07	Show pending Diagnostic Trouble Codes (detected during current or last driving cycle)
0x08	Control operation of on-board component/system
0x09	Request vehicle information
0x0A	Permanent DTC's (Cleared DTC's)

표 2

차량으로부터 데이터를 얻기 위해 ECU로 PID를 전달하기 위한 질의 CAN 메시지 형태는 각 메시지 바이트 별로 아래 표 3과 같은 형식을 갖는다. Data 전체 8byte 중 첫 번째 바이트(Byte 0)는 몇 바이트의 정보가 오는지를 의미하는 것으로 그 뒤로 Mode와 PID Code가 각각 1byte씩 두바이트가 오기 때문에 0x02가 되며, 그 이상은 사용되지 않으므로 0x55로 채워진다.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
추가데이터 바이트수 0x02	Mode 0x01=show current data	PID Code	0x55	0x55	0x55	0x55	0x55

표 3

표 3과 같은 CAN 메시지를 ECU로 전송하면 ECU는 표 4와 같은 형태의 CAN 메시지로 응답한다. 첫 번째 바이트는 질의 메시지와 동일하게 이후에 올 추가 데이터의 크기이다. 두 번째는 질의 메시지에서 사용된 Mode 값에 0x40을 더한 값과 같다. 네 번째부터는 PID에 대한 응답데이터가 된다.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
추가데이터 바이트수 0x03 ~0x06	Custom Mode 0x40+Mode	PID Code	Value 0 (A)	Value 1 (B)	Value 2 (C)	Value 3 (D)	0x55

표 4

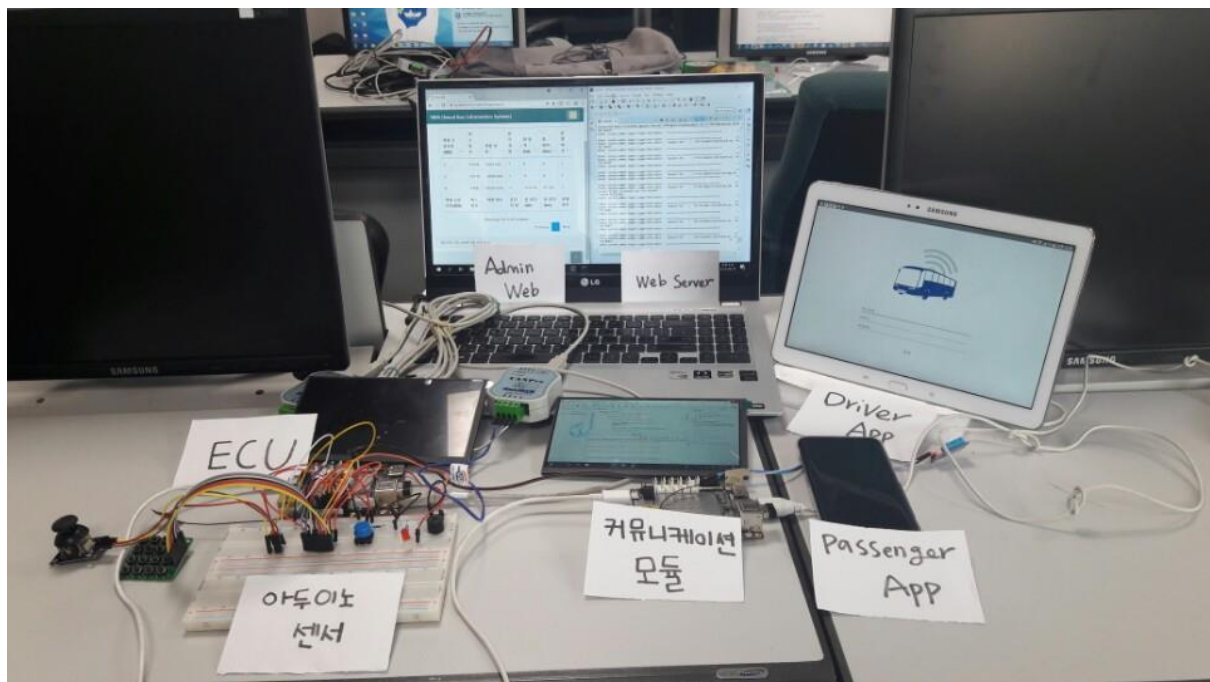
## 2. IoT - Android – Web Server] Network

大

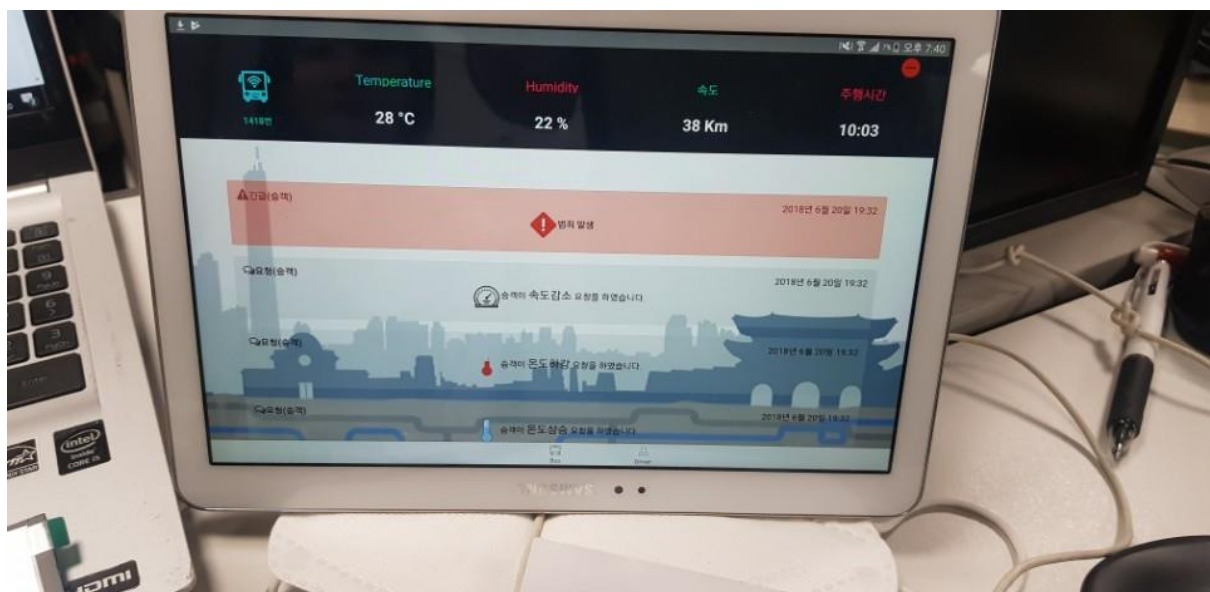
Driver App 은 IoT 장비와 Passenger App, Web Server 들 간에 유기적인 data 전송을 도와주는 연결고리 역할이다.

- ① TCP/IP 통신을 활용하여 IoT 장비가 전송한 data 를 수집한다.
- ② Driver App 의 TCP/IP Sender 는 Android Timer 를 활용하여 주기적으로 IoT 장비에게 온 · 습도 data, CAN data 요청 시그널을 보낸다. 또한, ③에서 승객들이 누른 하차 벨 동작 명령 데이터가 전송된다.
- ③ 승객들이 스마트폰으로 동작시킨 하차 벨 데이터를 Driver App 으로 전송한다.
- ④ Passenger App 이 Driver App 과 Socket 연결이 되면 기사와 차량 제원에 대한 데이터를 전송 받는다. 또한, 주기적으로 IoT 장비를 통해 수집된(①) 온도, 습도, 속도 data 를 전송 받는다.
- ⑤ 1) 버스의 위도, 경도 값을 Android Timer 를 활용하여 주기적으로 서버에 전송한다. 또한 주기적으로 IoT 장비를 통해 수집된(①) 온도, 습도, 속도 data 를 전송한다.  
2) 승객이 실시간으로 요청한 온도 조절 정보, 안전운전 요청 정보, 범죄 발생 정보들을 서버에서 주기적으로 수집한다.  
3) CAN data 를 통해 분석된 차량 이상 정보를 기사에게 보여준다.  
4) 기사 개인 페이지에 대한 정보를 통해 승객들의 피드백 보여준다
- ⑥ 승객은 온도 조절 정보, 안전운전 요청 정보, 범죄 발생 정보들을 서버에 전송한다. 또한, 고객의 소리 게시판을 활용해 버스와 기사에 대한 피드백을 가능하게 한다.

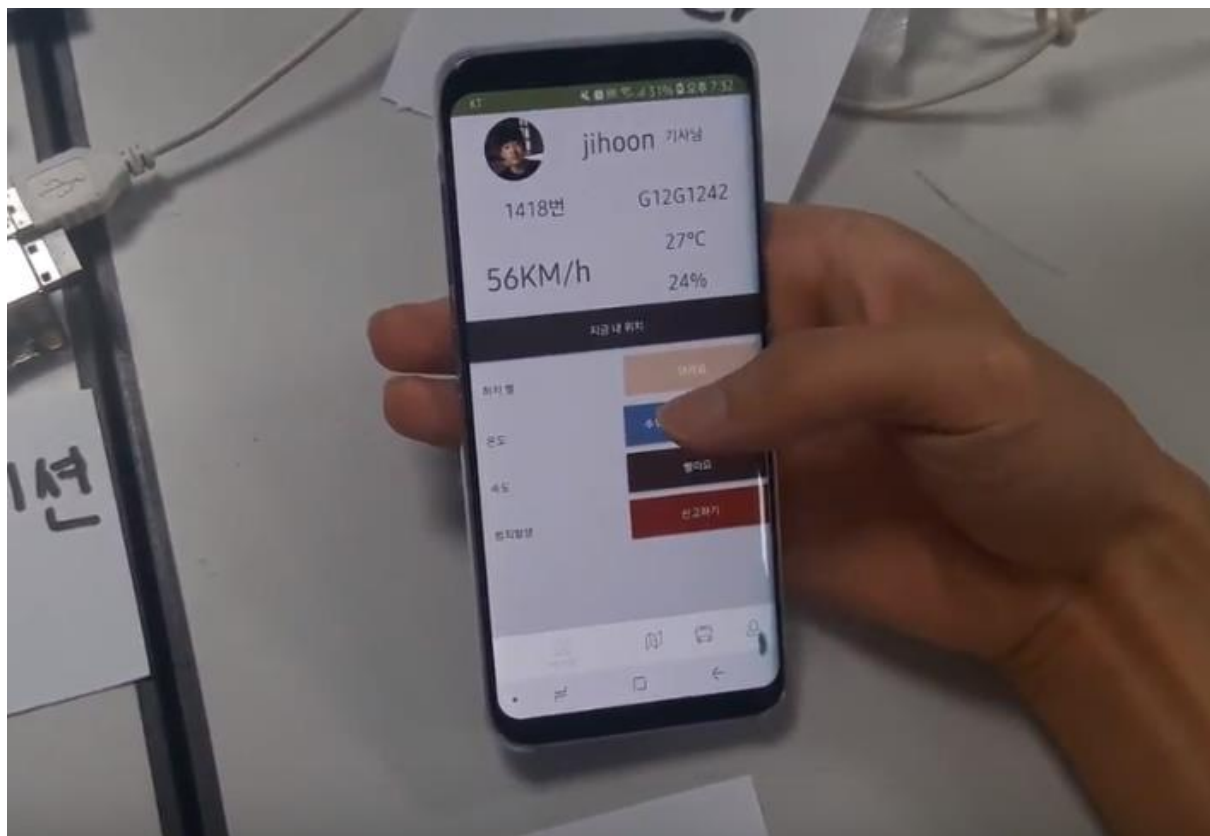
## 3.4 주요 동작



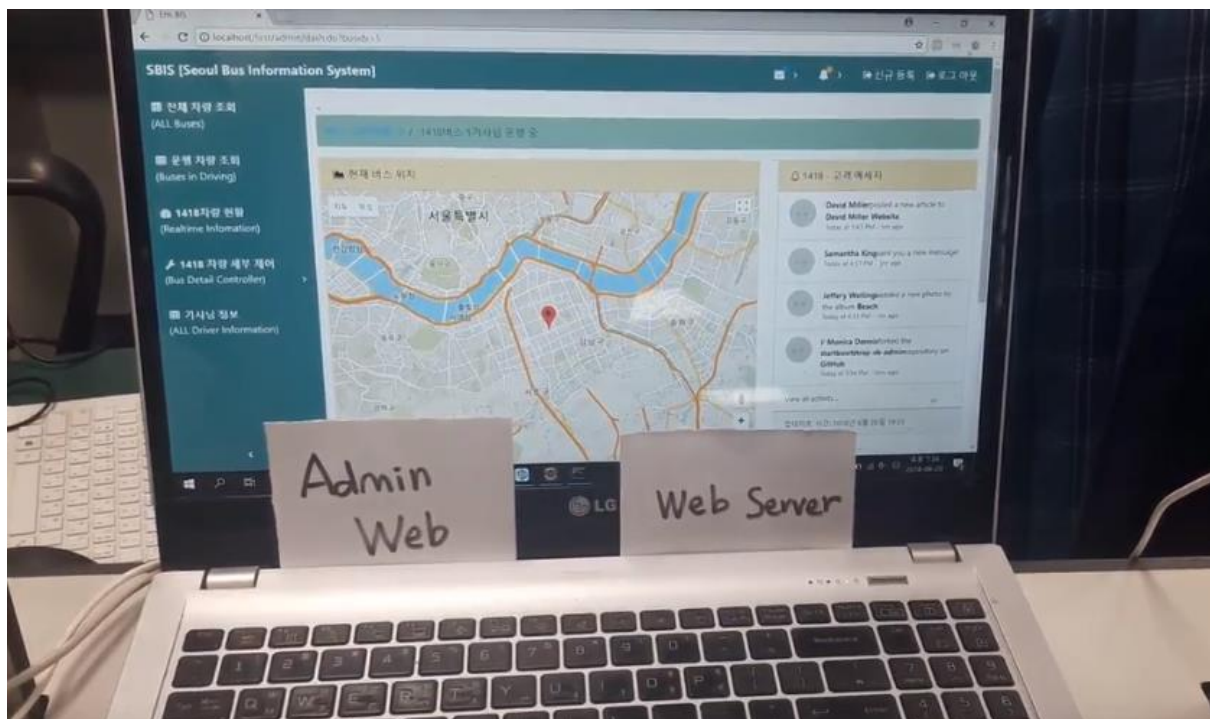
전체 기기 사진



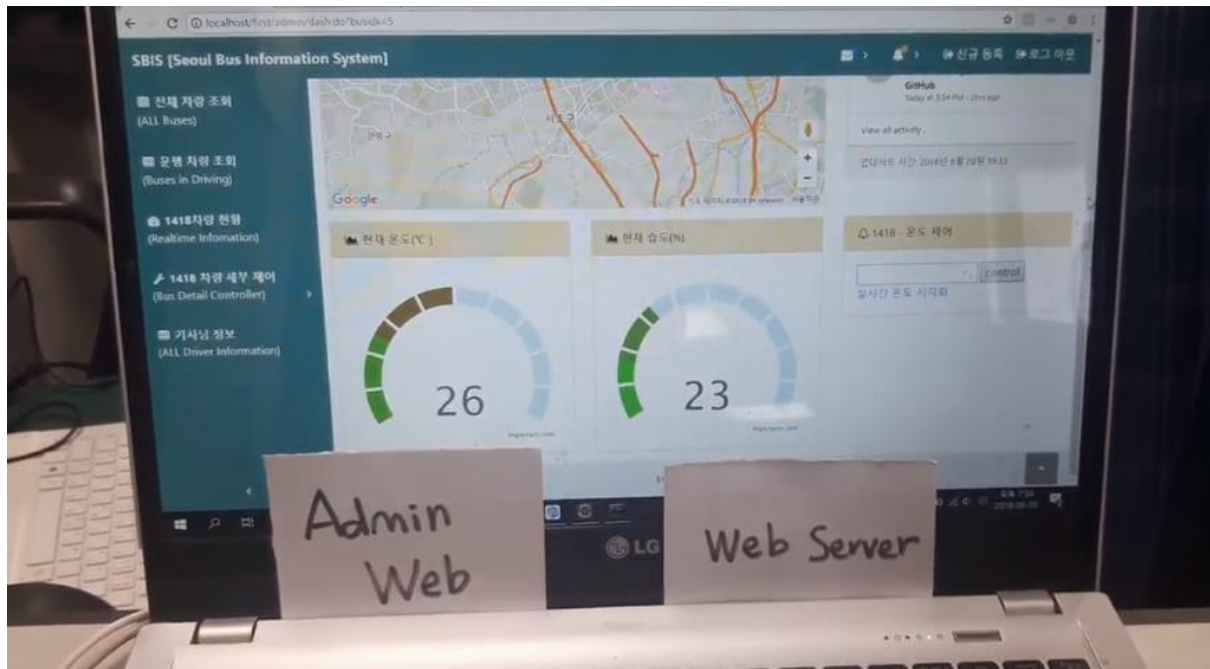
Driver App Main page



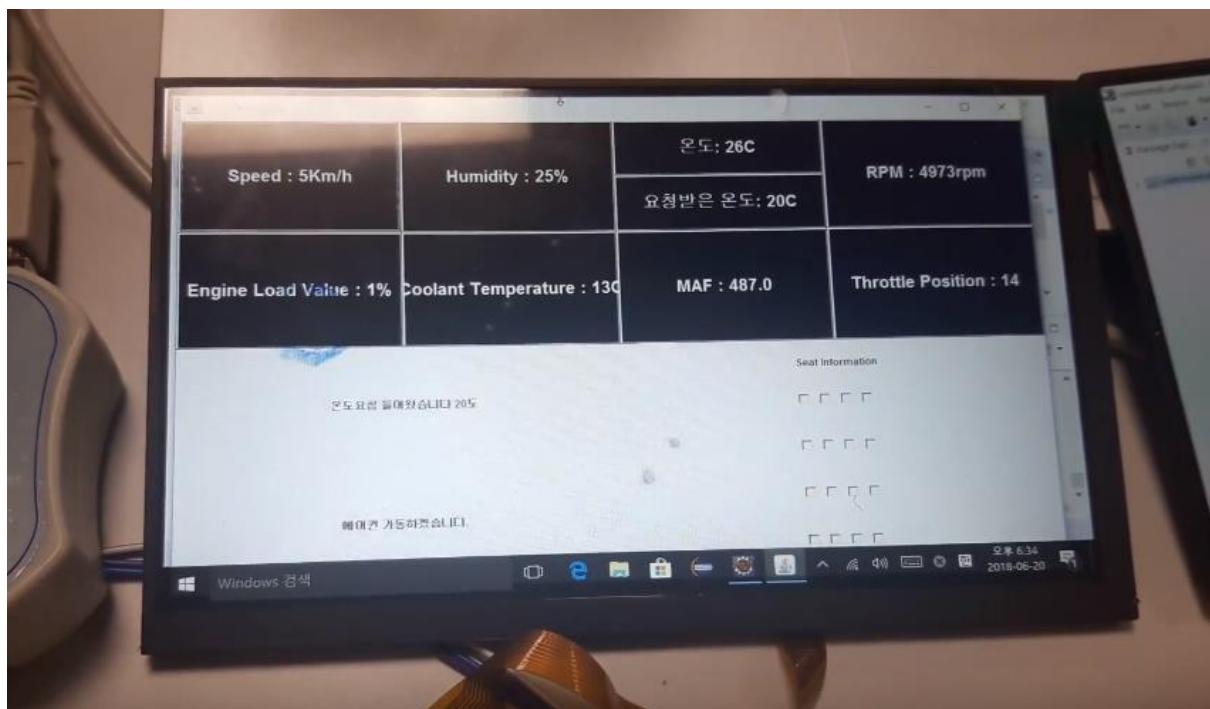
Passenger App Main page



특정 버스 Dashboard

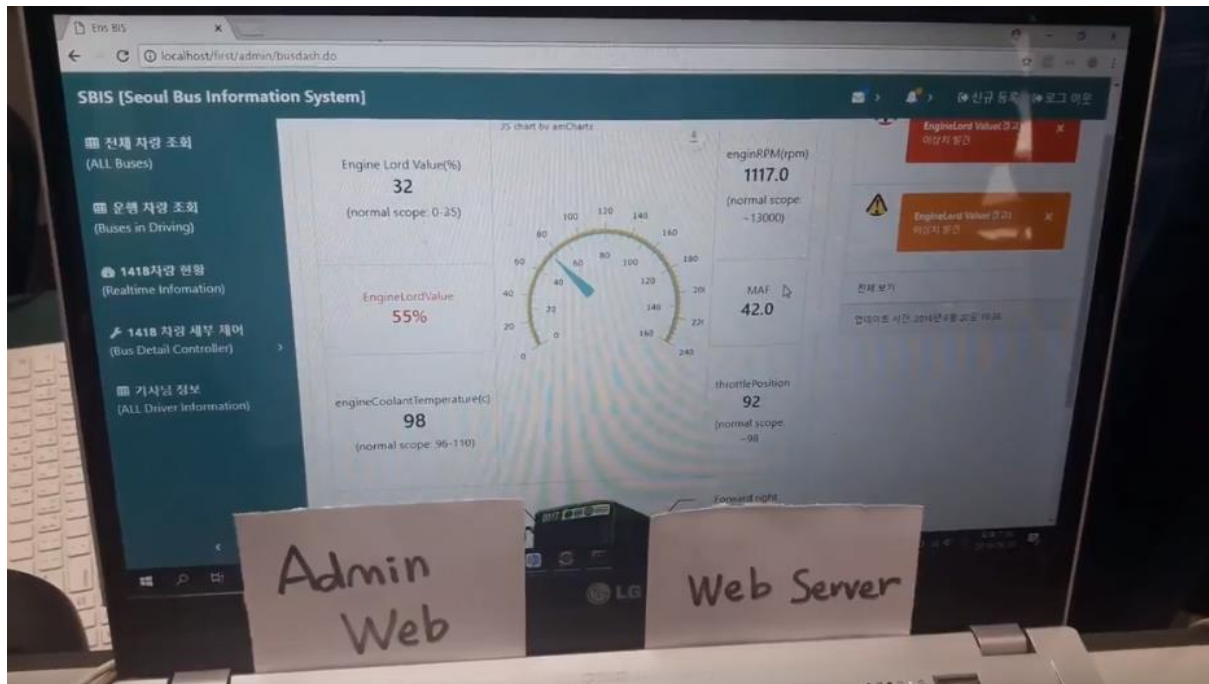


실시간 온 습도 정보

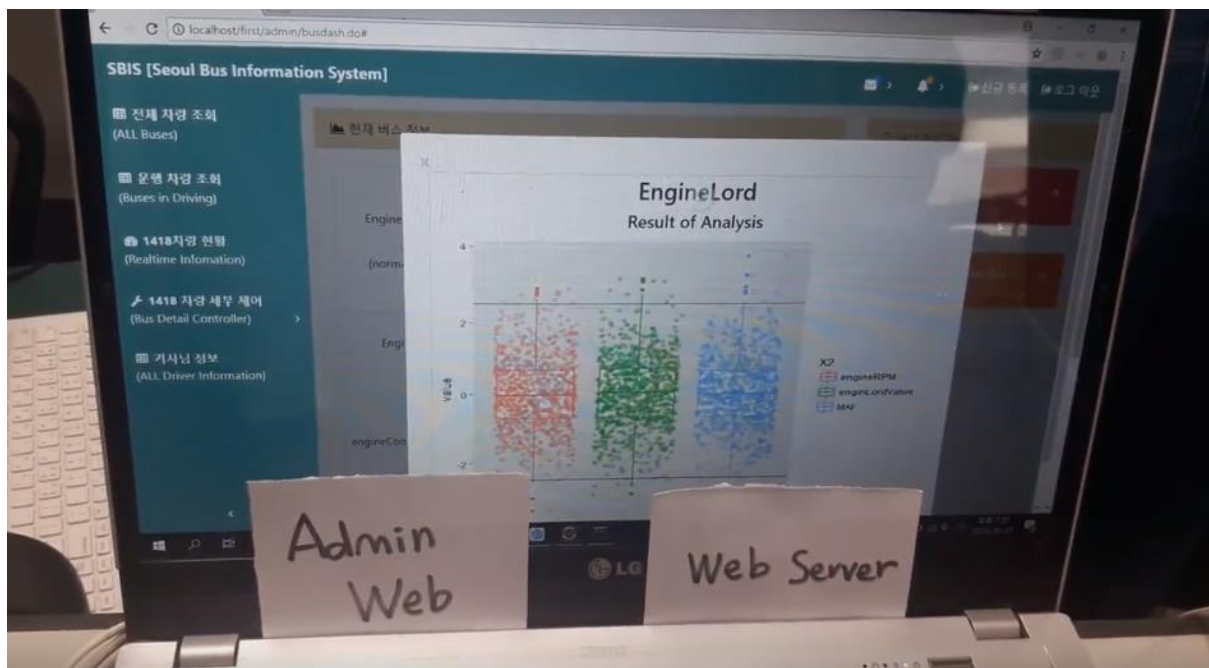


ECU 실시간 data





실시간 차량 정보 데이터



이상치 분석 차트

주요 동작 시연 동영상 : <https://youtu.be/022D2zJLRfQ>

## 4. 기대 효과

#### 4.1 향후 개선 사항

- 아두이노와 ECU 라떼판다간 통신 속도 차이로 인한 제한적인 정보 송수신 개선
- 아두이노의 HW 스펙으로 인한 제한적인 센서 사용 해결
- 다양한 ECU 간 CAN bus 통신 구현
- 안드로이드 통신 Thread 통합
- 안드로이드 Fragment 안정화
- 사용자 친화적인 안드로이드 APP UI

#### 4.2 기대 효과

BIS(Bus Information System) 는 크게 경제적, 지역적, 사회적 효과를 얻을 수 있다. 경제적 기대효과로는 대중교통 활성화에 따른 교통혼잡비용을 감소할 수 있고, 지속적인 차량 상태 확인으로 인해 차량운행과 정비비용을 감소시킬 수 있다. 또한 대중교통기반의 도로 인프라와 공공 WiFi 를 기반으로한 플랫폼 사업으로 확장 가능성이 높다. 플랫폼 사업은 사용자 이동경로에 따른 빅데이터를 구축해 관광 수익 모델로 확장 가능하다.

지역적 기대효과로는 버스 이용자의 정류소 대기 시간을 감소할 수 있으며, 대중교통 혼잡도 정보 제공을 통해 고객 편의성을 확대할 수 있다. 신뢰성 있는 고품질의 대중교통 서비스를 제공함으로써 고객의 서비스 만족도가 향상되며, 고객의 소리를 활용해 이용자의 불편사항과 민원이 감소될 것으로 예상된다. 결과적으로 대중교통을 활성화하여 보다 나은 서비스를 제공할 수 있다.

사회적 기대효과로는 표준 시스템 적용으로 타 지자체 연계 확장 구성이 가능하다. 또한 광역버스 정보 등 생활권 중심의 요구정보를 수용 기반을 조성해서 향후 광역 대중교통 연계의 기반을 마련할 수 있다.

### 5. 개발 후기 [대제목 : 16pt / 맑은 고딕]

--





--

성명	후기
박준하	과정 중에 배운 HTML5, CSS3, JavaScript, Spring MVC, Oracle DB, Android, Hadoop 모든 기술들을 활용할 수 있고 실질적으로 사용될 수 있는 커넥티드 카 프로젝트를 기획하고 개발할 수 있어 만족스러운 프로젝트였습니다. PM 으로서 System Architecture 를 설계하고 전체적인 프로젝트 일정을 관리하면서 전체적인 커넥티드 카 프로젝트의 데이터 흐름 과정을 알 수 있었습니다. 또한, 이더넷 통신에 기본이 되는 HTTP, TCP/IP 통신을 통해 Server 부터 IoT 장비까지 데이터를 주고 받고 기능을 제어할 수 있었던 프로젝트였습니다. IoT 프로젝트에서 중요한 이슈는 통신과 효율적인 Thread 관리라는 것을 알 수 있었습니다.
곽지훈	긴 시간 동안의 프로젝트를 마침내 마무리 지었습니다. 개발 초기에 목표했던 제품을 높은 완성도가 있게 구현을 할 수 있어 매우 인상깊고 오랜 기억에 남을 것 같은 프로젝트입니다. 다소 긴 시간의 프로젝트 기간 동안 지칠 법도 했지만 단 한번도 흥미를 잃지 않고 정말 열심히 개발에 몰두했던 것 같습니다. 이 덕에 기술 적인 부분뿐만 아니라 개발자로서 갖추어야할 기획 설계 등의 자질 부분에서도 깊게 배웠던 프로젝트가 되었습니다.
조용원	우선 긴 프로젝트를 무사히 마칠수있어서 감사합니다. 또한 학부시절부터 GPIO 를 이용한 센서 제어등에 관심이 많았는데 개인적으로 IOT 파트를 맡아서 프로젝트를 진행해서 기뻐했습니다. 아두이노와 자바를 연동하기 위해 시리얼 통신에 대한 이해도를 높였고 캔 버스 통신 방식을 공부할수있어서 뜻깊은 프로젝트였습니다. 초기에 TCP/IP 통신과 캔 시리얼 그리고 아두이노 시리얼 통신을 연동하는데 있어서 어려움이 있었지만 그만큼 공부도 많이 할 수 있었고 무엇보다 좋은 팀원들을 만나서 즐겁게 진행했던것 같습니다.
손애리	이 프로젝트를 하면서 지난 6 개월 간 배웠던 것을 모두 활용했습니다. IOT 장비(아두이노, CAN, 센서) 를 통하여, 외부로부터 정보를 받아오고, 데이터들을 저장, 분석하여, 서버로 명령을 동작시키고, 안드로이드 화면을 통해서 사용자와 소통이 가능하도록 했습니다. IOT 와 안드로이드, 서버, 데이터를 서로 연동시키는 과정에서 오류도 많았고, 디자인을 입히는 과정에서도 오류가 많아서 팀원 모두가 고생을 했지만, 하나씩 시도하면서 많은 것을 배울 수 있는 프로젝트였습니다.

## 6. 강사 의견 [대제목 : 16pt / 맑은 고딕]

평 가 요 소	배점	평
아이디어 : 유사한 서비스 존재 유무 및 체계성	/20	
2. 개발 : 실제 구현 정도 및 배포 유무, 코드의 무결성 및 난이도, 현업적용도, 실무기술 반영정도	/30	
3. PJT 수행력 : 일정관리 및 역할분담, 목표 일정 달성도, 팀내 참여도 등	/30	
4. 준비도 : 프리젠테이션 및 프로젝트 준비 정도	/20	
계	/100	

