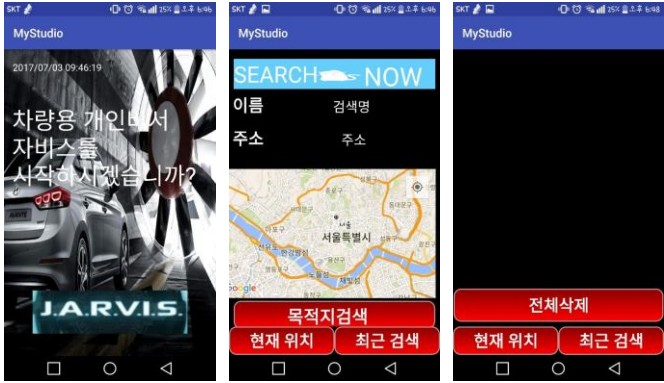
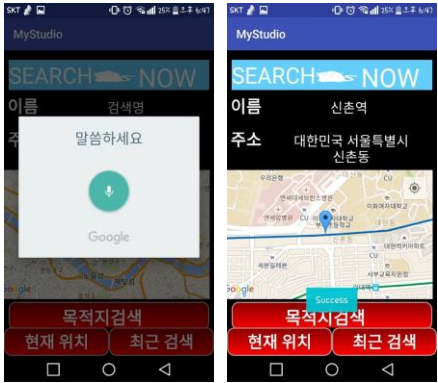


# 프로젝트 기술서

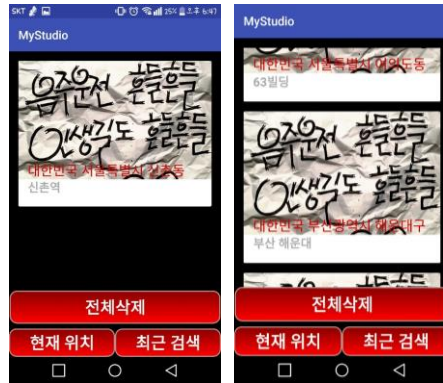
2017.3 - 2017.3

프로젝트명	자동주차시스템
소개	자동차의 주차보조시스템을 모티브로 한 초음파거리센서로 주차 공간을 확인 후 스텝모터로 핸들을 돌려 후진하여 주차하는 시스템
참여 인원	1인
본인이 맡은 부분	아두이노 우노로 센서 제어, 안드로이드와 아두이노 블루투스 통신, 라즈베리파이와 아두이노 시리얼 통신
개발 목적	주차에 미숙한 운전자를 위한 주차 시스템
사용 tool 및 언어	Raspberry pi, Arduino, Arduino IDE, Android, C
시연 영상 및 사진	자동 주차 시연 영상 : <a href="https://youtu.be/ffaUuokmC0s">https://youtu.be/ffaUuokmC0s</a> 주차 시 데이터 흐름 시연 영상 : <a href="https://youtu.be/rNSxs4kMd94">https://youtu.be/rNSxs4kMd94</a>
하드웨어 구성	아두이노 우노에 블루투스 통신 모듈 사용 안드로이드로 블루투스 App 사용 라즈베리파이에 초음파 거리 센서2개, 스텝모터, LED 사용
주요기능 및 진행 순서	<ol style="list-style-type: none"> <li>1. 안드로이드로 아두이노와 블루투스 통신을 활용해 자동주차 시스템의 시동을 켜다.</li> <li>2. 아두이노에서 안드로이드로부터 블루투스로 받은 Data를 시리얼통신으로 라즈베리파이에 송신(백색 LED 점등)</li> <li>3. 전진을 하며 초음파거리센서 2개로부터 주차공간을 찾는다.(녹색 LED 점등)</li> <li>4. 주차공간을 찾았을 경우 자동차의 크기만큼 전진을 한다.(녹색 LED 깜빡임)</li> <li>5. 주차하기 위해 핸들을 돌린다(노랑색 LED 점등 및 스텝모터 구동)</li> <li>6. 핸들을 전부 돌린 후 후진을 한다(빨간색 LED 점등)</li> <li>7. 인터럽트 핀이 다른 물체에 닿을 경우 시동을 끈다(백색 LED 소등)</li> </ol>

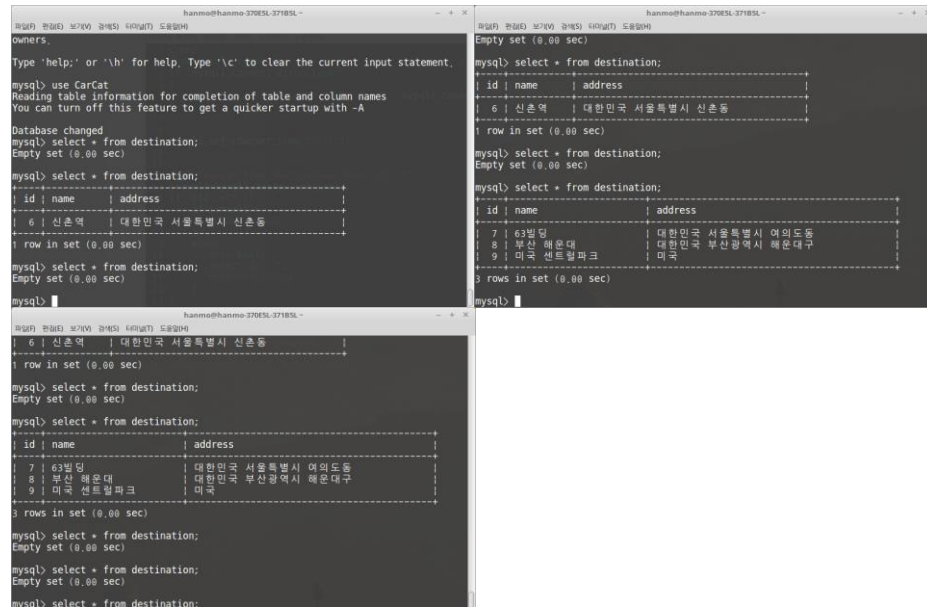
2017.6 - 2017.6

프로젝트명	자동차용 개인 비서	
소개	운전 중 안전을 위해 음성인식으로 목적지를 검색, 검색하였던 목적지를 조회 및 삭제 가능	
참여 인원	3인	
본인이 맡은 부분	프로젝트 아이디어 제공(팀장) 서버, DB 구축 및 Android 코딩(STT, Google Map API 활용 및 Recycler view) Raspberry Pi와 Android 통신, GitHub로 프로젝트 버전 관리	
개발 목적	안전운전을 위해 최소한의 터치로 목적지를 검색 최근 목적지 조회 및 삭제	
개발 도구	개발 하드웨어	Raspberry Pi, Android
	개발 SW 및 언어	Android Studio, Apache, Php, Mysql, Github, JAVA
	개발 환경	Windows, Linux
시연 영상 및 사진	<p>기본 화면 :</p>  <p>목적지 검색 화면 :</p> 	

최근 목적지 조회 화면 :



DB 화면 :



시연 영상 : <https://youtu.be/fNF68P08h2k>

하드웨어 구성

Android로 Client 구현

Raspberry Pi로 Server, DB 구현

주요기능  
및 진행 순서

1. Google Maps 화면의 현재 위치 표시
2. STT를 통한 사용자의 음성 입력
3. 입력 받은 값을 GeoCoder를 활용하여 좌표(위도, 경도) 변환
4. Google Map의 화면에 해당 위치 마커 추가, MySQL DB Insert 질의문 실행
5. php를 사용하여 DB에 select 질의문을 실행하여 최근 검색지 결과 출력
6. swipe를 통한 DB Delete 질의문 실행(recycler view의 item 삭제)
7. 전체 삭제 버튼을 통한 DB insert된 데이터 삭제



## Table 설계 : 차간거리

PK	AI	FK	Null	Name	Type	
✓				checkTime	DATETIME	-
				frontDistance	INT	-
				rightDistance	INT	-
				leftDistance	INT	-

PK	AI	FK	Null	Name	Type	
✓				time	DATETIME	-
				winker	NVARCHAR(1)	-
				handle	NVARCHAR(1)	-

## 차선변경

PK	AI	FK	Null	Name	Type	
✓				checkTime	DATETIME	-
				frontDistance	INT	-
				rightDistance	INT	-
				leftDistance	INT	-

PK	AI	FK	Null	Name	Type	
✓				time	DATETIME	-
				winker	NVARCHAR(1)	-
				handle	NVARCHAR(1)	-

## 저장 프로시저 사용 :

### Insert

```

mysql> create table inputensor (
  -> input_val int not null auto_increment primary key,
  -> input_time datetime not null,
  -> )
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'datetime' at line 4
mysql> create table inputensor (
  -> input_val int not null auto_increment primary key,
  -> input_time datetime
  -> )
Query OK, 0 rows affected (0.22 sec)

mysql> delimiter //
mysql> create procedure insertdata
  -> inputVal varchar(10),
  -> inputTime datetime
  -> begin
  -> insert into inputensor (input_val, input_time) values (inputVal, inputTime);
  -> end //
Query OK, 0 rows affected (0.08 sec)

mysql> delimiter ;
mysql>

```

### Select

```

mysql> call insertdata(20, '2017-07-07 19:14:02');
Query OK, 0 rows affected (0.00 sec)

mysql> select * from inputensor;
+-----+-----+
| input_val | input_val | input_time |
+-----+-----+
| 20 | 20 | 2017-07-07 19:14:02 |
+-----+-----+
1 row in set (0.00 sec)

mysql> delimiter //
mysql> create procedure selectdata()
  -> select * from inputensor;
  -> end //
Query OK, 0 rows affected (0.00 sec)

mysql> delimiter ;
mysql> call selectdata();
+-----+-----+
| input_val | input_val | input_time |
+-----+-----+
| 20 | 20 | 2017-07-07 19:14:02 |
+-----+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql>

```

## 데이터 통신 :

```

Frnt/52/Right/119/Left/1000/

<type 'str'>
2017-07-18 12:39:19
data[1] -> 
52
data[2] -> 
119
data[5] -> 
1000

{'Front': '52', 'Time': '2017-07-18 12:39:19', 'Right': '119', 'Left': '1000'}
(queue size is : 23)

```

	<p>시연 영상 :</p> <p>스트리밍 영상 : <a href="https://youtu.be/mZtRV7dNnOs">https://youtu.be/mZtRV7dNnOs</a></p> <p>사이드 미러 영상 : <a href="https://youtu.be/KNyswvB_xVg">https://youtu.be/KNyswvB_xVg</a></p>
하드웨어 구성	<p>Android로 Client 구현</p> <p>Raspberry Pi로 Server, DB 구현 및</p> <p>Arduino로 입출력 센서 제어</p>
주요기능 및 진행 순서	<ol style="list-style-type: none"> <li>1. 아두이노에서 초음파거리센서 3개의 값을 각각 받아 시리얼 통신으로 라즈베리파이와 통신</li> <li>2. 라즈베리 파이에서 받은 데이터를 자료구조 Queue에 쌓는다.</li> <li>3. 3개의 센서 값이 전부 Queue에 쌓이면 Node.js(Rest api)에서 POST메서드로 DB에 Insert한다.</li> <li>4. Rest api에서 GET 메서드를 통해 DB에 있는 Data에 따라 해당 거리를 위험, 경고, 안전 등 3가지 케이스로 구분하여 측정 거리에 따른 이미지를 디스플레이에 출력한다.</li> <li>5. 핸들과 방향지시등에서의 조작을 DB에 Insert 시켜 방향지시등의 값에 따라 내부 디스플레이(Android)에 카메라 영상을 스트리밍 한다.</li> <li>6. DB를 암호화 하여 usb(UUID가 KEY)에 저장한다.</li> </ol>