

종합설계 프로젝트 수행 보고서

프로젝트명	오토바이 공유 시스템
팀번호	S1-4
문서제목	수행계획서() 2차발표 중간보고서() 3차발표 중간보고서() 4차발표 중간보고서() 최종결과보고서(O)

2020.12.04

팀원 : 한승우 (팀장)
권태민
최경호

지도교수 :

최경호

문서 수정 내역

작성일	대표작성자	버전(Revision)	수정내용	
2020.01.21	한승우(팀장)	1.0	수행계획서	최초작성
2020.03.01	한승우(팀장)	2.0	2차발표자료	설계서추가
2020.05.02	한승우(팀장)	3.0	3차발표자료	시험결과추가
2020.06.27	한승우(팀장)	4.0	4차발표자료	테스트, 코드추가
2020.12.04	한승우(팀장)	5.0	최종결과보고서	시험결과 수정

문서 구성

진행단계	프로젝트 계획서 발표	중간발표1 (2월)	중간발표2 (4월)	학기말발표 (6월)	최종발표 (10월)
기본양식	계획서 양식	계획서 양식	계획서 양식	계획서 양식	계획서 양식
포함되는 내용	I. 서론 (1~6)	I. 서론 (1~6)	I. 서론 (1~6)	I. 서론 (1~6)	I II III
	II. 본론 (1~3)	II. 본론 (1~4)	II. 본론 (1~5)	II. 본론 (1~7)	
	참고자료	참고자료	참고자료	참고자료	

이 문서는 한국산업기술대학교 컴퓨터공학부의
 “종합설계” 교과목에서 프로젝트 “오토바이 공유시스템” 을
 수행하는
 (S1-4, 한승우, 권태민, 최경호)들이 작성한 것으로 사용하기
 위해서는 팀원들의 허락이 필요합니다.

목 차

I. 서론

1. 작품선정 배경 및 필요성	1
2. 기존 연구/기술동향 분석	2
3. 개발 목표	3
4. 팀 역할 분담	4
5. 개발 일정	4
6. 개발 환경	5

II. 본론

1. 개발 내용	6
2. 문제 및 해결방안	8
3. 시험시나리오	9
4. 상세 설계	10
5. Prototype 구현	18
6. 시험/ 테스트 결과	24
7. Coding & DEMO	26

III. 결론

1. 연구 결과	
2. 작품제작 소요재료 목록	

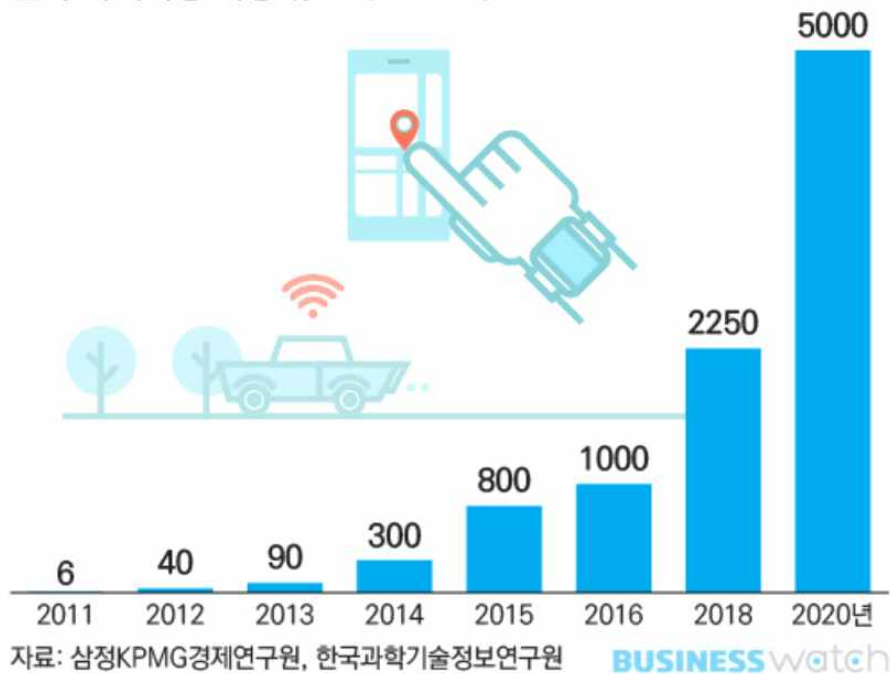
참고자료	
------------	--

I. 서론

1. 작품선정 배경 및 필요성

최근 카 셰어링 시장 규모가 지속적으로 커지고 있다. 카 셰어링이란 자동차를 빌려 쓰는 방법 중의 하나로 일종의 공유 경제 시스템이다. 렌터카와는 달리 주택가 근처에 보관소가 있으며, 주로 시간 단위로 필요한 만큼만 쓰고 차를 갖다주는 방식으로 이용한다. 시간 단위로 빌리기 때문에 간단하게 장을 볼 때나 짐을 옮길 때 등 종종 벌어지는 간단한 상황에서 이용할 수 있다. 밑의 사진은 한국과학기술정보연구원에서 조사한 자료로서 국내의 카셰어링 시장 규모가 커지고 있다는 지표를 나타내고 있다. 성장세는 가파르게 이루어져있다. 국내에서도 그렇지만 해외서는 더욱더 향상된 그래프를 나타내고 있다.

한국 카셰어링 시장 규모 [단위: 억원]



이처럼 이동수단을 셰어링에 대한 수요가 늘어나고 있지만 국내에 자동차와 전동 킥보드에 대한 서비스는 많이 있지만 아직 오토바이에 관련된 서비스가 없다. 면허는 있지만 오토바이를 구매하거나 소유하기 어려운 사람들이 자유롭게 이용 가능하고 차를 소유하고 있는 사람도 때로는 오토바이를 이용하고 싶을 때 편이 이용할 수 있다.

2. 기존 연구/기술동향 분석

- 쏘카(SOCAR)



스마트폰 앱을 통해 이용 시간을 선택하고, 내 근처 쏘카존을 선택하거나 쏘카를 원하는 장소로 부를 수 있다. 그리고 사용자는 원하는 차량과 면책 상품을 선택할 수 있다. 앱 안에 있는 스마트키를 이용하여 비상등과 경적 버튼을 통해 예약한 쏘카를 찾을 수 있고, 문 잠금과 문 열기가 가능하다. 운전면허와 결제카드를 등록하여 이용이 가능하다. 차량 이용이 끝나면 지정된 장소에 주차 후 앱 내 반납하기를 이용하여 반납한다.

- 킥 고잉(KICK GOING)



스마트 폰 앱에 접속하여 지도 상에 표시되어 있는 자신의 현재 위치와 가장 가까운 전동 킥보드를 찾는다. 이용할 킥보드에서 부착된 QR 코드를 인식한다. 사용자는 자유롭게 이용한 후에 반납할 때 지도에 표시된 곳에 주차한다.

3. 개발 목표

우선 궁극적인 목표를 말하자면, 무인으로 오토바이를 빌려 반납하는 서비스를 구현하는 것이 정확한 개발 목표라고 할 수 있다.

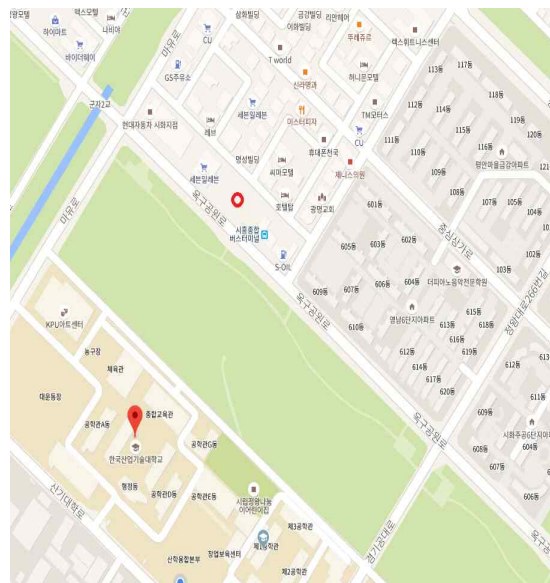
사용자와 오토바이 사이의 위치 정보를 사용자가 조회할 수 있게하여, 사용자가 오토바이를 선택하여 사용 할 수 있게 하고, 사용자가 오토바이 공유 시스템을 사용할 때 어플리케이션 기능중 하나인 탭박스 스마트 키를 이용하여 탭박스의 잠금장치를 제어하고 헬멧과 키를 이용 할 수 있는 시스템을 개발하는 것이다.

사용자와 오토바이 사이의 위치 정보 데이터를 받기 위해서 사용자의 단말기, 즉 스마트폰과 오토바이 탭박스 내에 부착되어 있는 아두이노를 통해 GPS 데이터 값을 받게 되는데, 아두이노의 GPS 모듈을 통하여 개발하고, 구글MAPS API를 통해 가시화할 것이다.

또한 탭박스 스마트 키를 이용하여 탭박스의 잠금장치를 제어하는 방법에 있어서는 블루투스 모듈을 활용한다. 사용자의 스마트폰과 아두이노 사이에 블루투스 연결을 완료한 후, 어플리케이션을 통해 사용자는 스마트키 실행 UI를 탭하게 되면 스마트키 실행 신호 코드가 전송되어 아두이노에서는 신호 코드에 맞는 이벤트를 처리하게 한다.



< 앱을 통한 탭박스 잠금제어 >



< 구글MAPS API를 통해 가시화된 모습 >

4. 팀 역할 분담



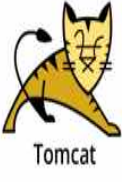


	한승우	권태민	최경호
탐박스	탐박스 내부에 아두이노 연결	탐박스 내부에 아두이노 연결	오토바이 탐박스 설치
아두이노 및 센서	아두이노 설계	아두이노 센서 값 서버 전송	아두이노 센서 값 서버 전송
APP	어플리케이션 UI 제작	어플리케이션 주요 기능 개발	어플리케이션 주요기능 개발
서버	서버와 DB 연동 서버와 아두이노 연동	서버와 APP 연동	서버와 APP 연동
DB	DB Schema 설계 저장된 DATA APP 연동	DB에 서버 데이터 저장	DB에 서버 데이터 저장

5. 개발 일정


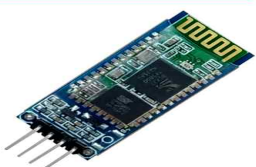
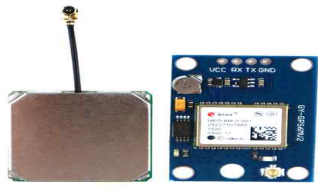
	추진 사항	12월	1월	2월	3월	4월	5월	6월	7-9 월
추진 일정	주제에 따른 사전조사, 제안서작성 및 제출								
	보드, 센서 테스트								
	하드웨어 설계 및 제작								
	Application 개발								
	모듈 통합 및 구현								
	통합 테스트 및 보완								
	중간 보고서 제출								
	발표 및 시연								

6. 개발 환경

- SW 개발환경

				
안드로이드 스튜디오	아두이노 IDE	Apache Tomcat	php	My SQL
개발사: 구글 언어: Java	개발사 : 아두이노 소프트웨어 언어 : C	개발사 : 아파치 언어 : Java	개발사: php 라이센스 종류 : 웹 서버	개발사: MySQLAB 종류: DB

- HW 개발환경

		
아두이노 우노+WIFI D1 R1 보드(ESP8266)	HC-06	GPS 모듈
동작 전압 : 3.3V 디지털 핀 : 11개 아날로그 핀 : 80uA	모델명 : HC-06 핀 구성 : 6핀 규격 : 43x15x7mm 신호범위 : 약 10m	모델명: UART GPS NEO-6m 초기 보드레이트 : 9600bps 전압 : 3V to 5V

II. 본론

1. 개발 내용

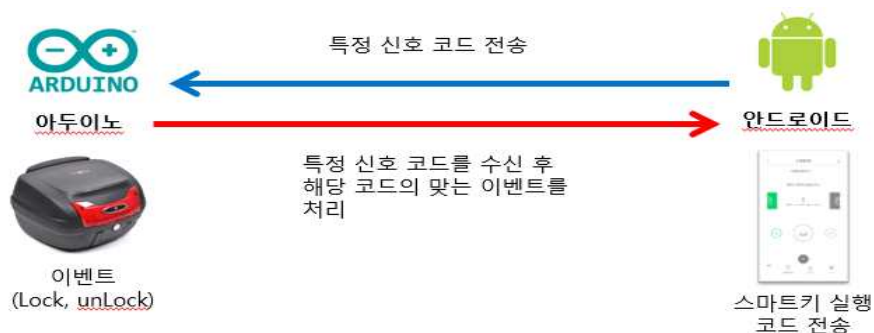
오토바이 공유시스템 개발은 크게 세 가지로 나눌 수 있다.

오토바이의 상태나 위치를 알려주는 아두이노 개발과 어플리케이션 개발 마지막으로 DB와 서버를 개발하는 것이다.

어플리케이션과 서버 또 아두이노에서 서버로 데이터 유동이 발생하게 되며 고객의 스마트폰 단말기를 통해 어플리케이션과 아두이노 사이에서 블루투스 연결이 발생하게 된다.

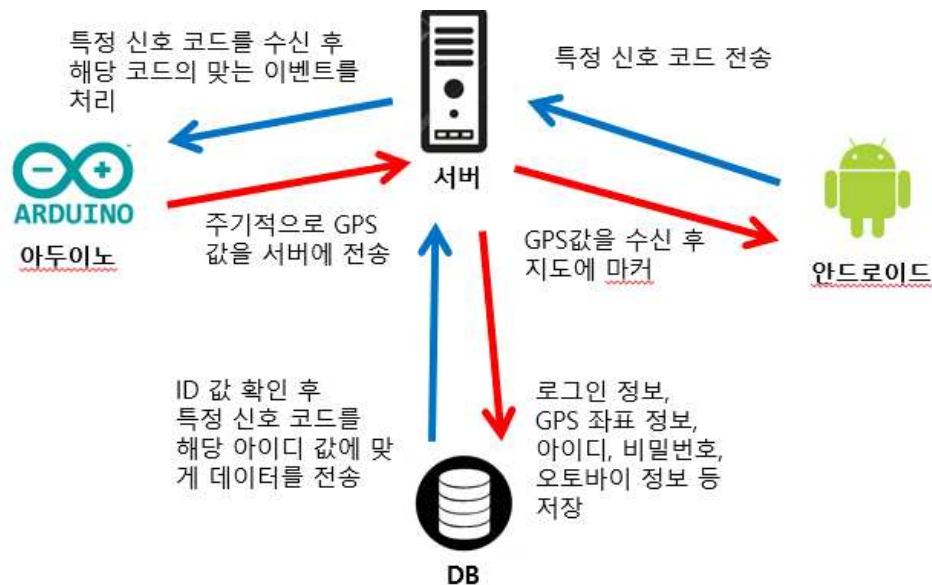
아두이노 개발은 오토바이의 위치 정보와 어플리케이션의 기능중 하나인 스마트 키를 통한 탑박스 제어를 위해 하게되는데, 아두이노 우노 보드에 GPS 센서, 블루투스 센서, 솔레노이드 잠금 장치모듈을 연결하여 GPS 센서는 데이터 값을 주기적으로 서버에 전송하도록 개발한다. 또한 블루투스 통신을 통한 잠금제어 이벤트 발생기능을 개발하는 것이고, 솔레노이드 잠금 장치는 블루투스 연결이 완료된 후 잠금제어 이벤트 발생을 읽어 기능을 수행하도록 개발한다.

다음으로 어플리케이션 개발이다. 어플리케이션은 GPS 정보를 활용하여 서버에 사용자 위치와 가까운 오토바이의 위치를 지도에 표기하고 블루투스 통신을 통해 탑박스 스마트 키 기능을 사용한다. 마지막으로 사용자는 오토바이의 예약 정보를 통해 비어있는 시간대에 예약을 할 수 있고 예약하여 사용 시간이 되면 사용자는 스마트 키의 권한을 부여받을 수 있다.



<블루투스 통신을 통한 탑박스 잠금제어권 확보>

마지막으로 DB와 서버 개발이다. Tomcat과 MySQL을 통해 서버와 DB를 구축한다. 오토바이 DB에는 오토바이의 기본정보와 오토바이만의 예약정보를 저장하고 GPS 정보를 가져온다. 사용자 DB로는 사용자의 기본적인 정보와 오토바이를 사용할 수 있는 자격인 면허 사진이나 GPS 정보를 가져올 수 있게한다. 또한 관리자 DB에는 오토바이의 정보들과 사용자들의 개인정보나 이용 내역 등을 관리할 수 있게 한다.



<서버와 아두이노, 사용자간의 GPS 데이터의 유동>

2. 문제 및 해결방안

요금 관련 문제 해결방안

10분 기준, 모닝 기준

모닝 연비 12.6km

2020.01.28 (원/리터) 1570원

모닝 km당 124.6원

쏘카 : 10분당 1200원 km당 170원

그린카 : 10분당 12000원 km당 170원

킵고잉 : 기본요금 1000원(5분) + 분당 100원

일반 스쿠터 연비 평균 30km

2020.01.28 (원/리터) 1570원

km당 52.3 원

3. 시험시나리오

1. 사용자는 스마트폰 앱을 통해 회원가입을 하고 운전면허증을 등록한다.
2. 자신의 GPS 정보를 통해 근처에 있는 오토바이의 위치를 확인한다.
3. 사용하려고 하는 오토바이의 정보와 사용할 수 있는 시간 정보가 나오고 예약이나 사용할 수 있다.
4. 예약을 하게 되면 예약한 오토바이와 블루투스 연결을 할 수 있는 권한이 생기고, 블루투스 연결설정이 완료되면 어플리케이션을 통해 스마트 키 기능을 사용할 수 있다.
5. 탑박스 스마트 키를 사용하여 탑박스의 잠금상태를 제어할 수 있게 되고, 탑박스 내의 헬멧을 사용할 수 있다. 동일하게 키박스 스마트 키를 사용하여 키박스의 잠금 상태를 제어할 수 있으며, 키박스 내의 와이어로 연결되어 있는 오토바이 시동키를 사용할 수 있다.
6. 사용이 끝나면 오토바이 시동키를 키박스에 넣은 후 반납하기 버튼을 누르면 결제 금액을 어플리케이션을 통해 확인할 수 있고, 결제한 후 지정된 장소에 반납한다.



< 시나리오 2 >



< 시나리오 3 >



< 시나리오 4 >

4. 상세설계

가입하기

1. 앱에서 각종정보 입력
2. 아이디 중복 검사를 하면 서버와 연동을 통해 db에 같은 아이디가 있는지 검사
3. 카드 정보 입력
4. 가입 버튼을 누르면 서버랑 연동을 통해 db에 저장

예약하기

1. app에서 핸드폰 gps를 통해 자신의 위치정보를 표시
2. 사용자의 위치정보를 통해 서버에 저장되어 있는 근처 야타존을 표시
3. 야타존을 누르게 되면 야타존에 사용할수 있는 오토바이 정보를 표시
4. 오토바이를 누르게 되면 오토바이를 사용할 수 있는 시간대를 표시
5. 사용할 수 있는 오토바이의 시간대를 설정한다.
6. 시간에 맞게 10분당 1000원의 가격으로 가격을 보여준다.
7. 예약하기 버튼을 누른다.
8. app 예약정보를 통해 서버와 연동을 해서 예약 테이블, 오토바이 테이블, 사용자 테이블을 갱신한다.

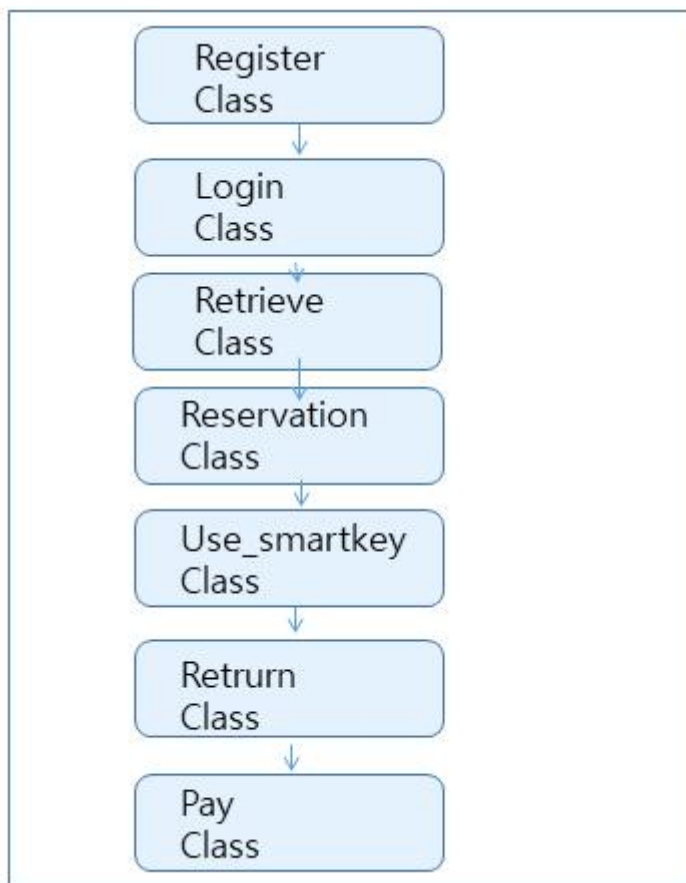
사용하기

1. 사용시간이 되면 오토바이의 대한 권한을 가지게 한다.
2. 오토바이의 아두이노와 블루투스 통신을 한다.
3. app의 스마트키를 이용하여 오토바이의 탑박스를 lock/unlock을 할 수 있다.
4. unlock버튼을 클릭하면 아두이노의 솔레노이드 장치가 작동되어 실제 키를 사용할 수 있다.
5. 사용자는 오토바이를 이용한다.
6. 오토바이 gps모듈을 통해 위치정보가 계속해서 갱신되면서 이동되는 경로를 통해
km당 100원의 가격으로 이용요금이 갱신된다.

반납

1. 사용을 마치면 사용자는 대여했던 야타존에 다시 가져다 놓는다.
2. 키를 탭박스 안에 넣고 app에 반납하기 버튼을 클릭한다.
3. 아두이노의 gps정보를 통해 야타존에 오토바이가 있는 걸 확인한 뒤 반납이 되었다고 표시한다.
4. 거리에 비례한 이용요금을 통해 사용자의 카드정보를 통해 요금이 부과된다.

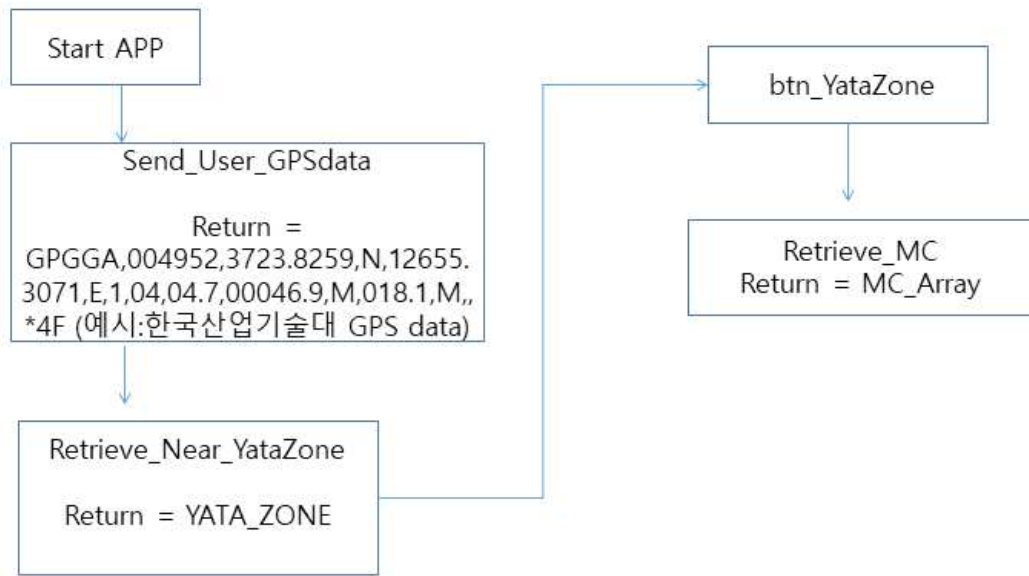
App



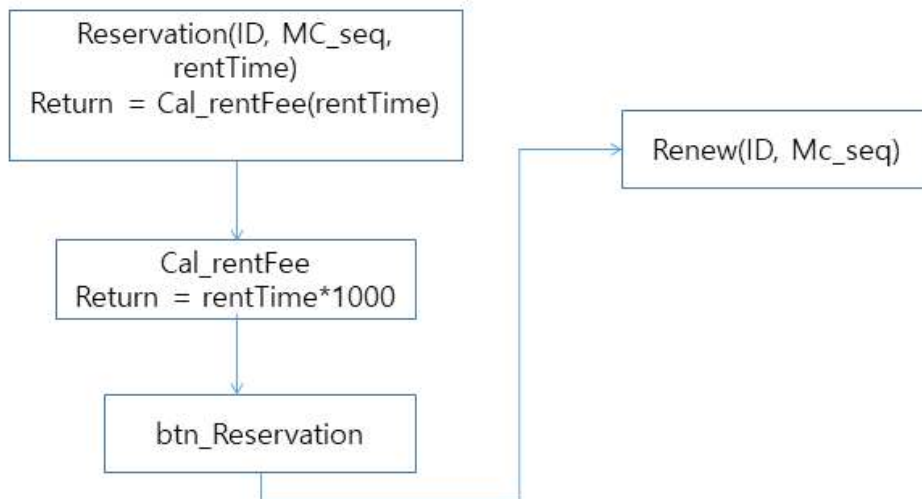
Register Class : 사용자의 각종 정보를 기입 하여 회원 가입을 하는 클래스

Login Class : 서버에 저장된 사용자의 아이디와 패스워드를 통해 로그인 하는 클래스

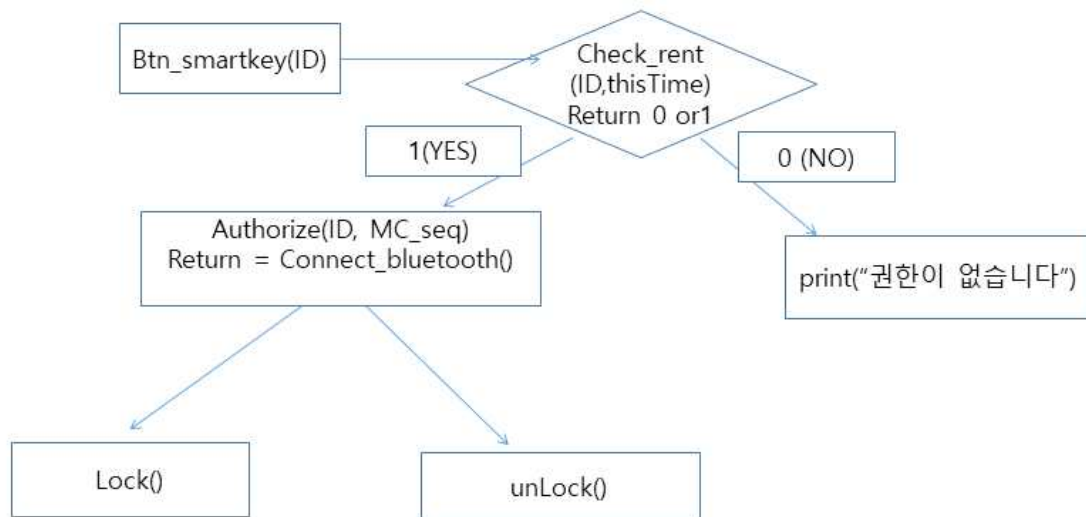
Retrieve Class : 사용자와 가용한 오토바이를 조회하는 클래스



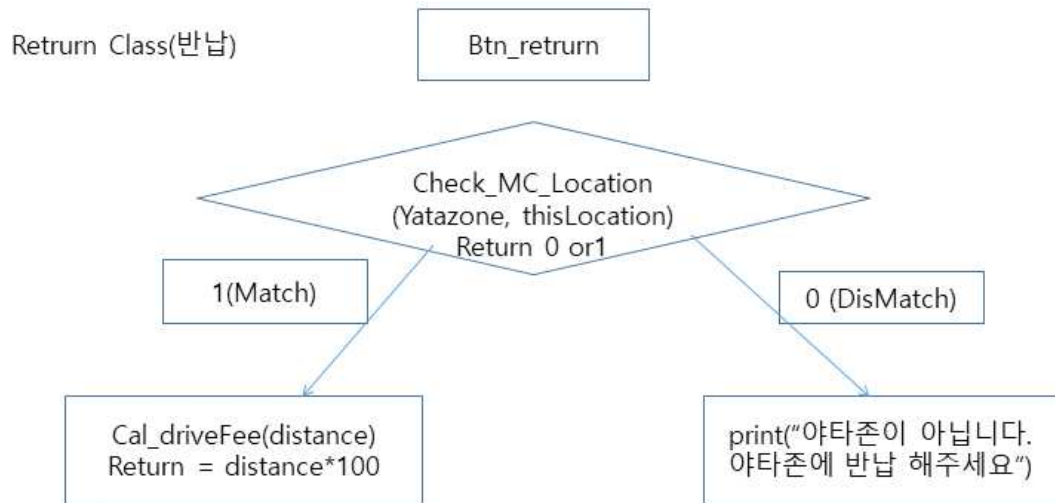
Reservation Class : 사용자가 오토바이를 예약 하는 클래스



Use_smartkey Class : 오토바이의 키박스를 제어하는 스마트키를 이용하는 클래스



Retrun Class : 사용을 종료하고 반납하는 클래스



Pay Class : 기본요금과 이용요금을 납부하는 클래스

DB

Member Table
ID
PW
LICENSE

Card Table
CARD_NUM
ID

Mc Table
MC_Seq
MC_NUM
YATA_ZONE
FEE

Rent Table
RENT_Seq
TIME
MC_Seq
ID

Payment Table
PAYMENT_Seq
RENT_Seq
CARD_NUM
ID

회원 테이블

No	컬럼	자료형	PK	비고
1	ID	Varchar(25)	O	아이디
2	NAME	Varchar(25)		이름
3	PW	Varchar(25)		비밀번호
4	PHONE	Varchar(25)		전화번호
5	ADDRESS	Varchar(25)		주소
6	LICENSE_TYPE	Varchar(25)		면허종류
7	LICENSE_NUM	Varchar(25)		면허번호
8	LICENSE_START	Date		면허 발급일
9	LICENSE_END	Date		갱신 만료일
10	BIRTH	Varchar(8)		생년월일

카드 테이블

No	컬럼	자료형	PK	비고
1	CARD_NUM	Varchar(25)	O	카드번호
2	CARD_PW	Varchar(25)		카드 비밀번호
3	CARD_MONTH	Varchar(2)		만료 월
4	CARD_YEAR	Varchar(2)		만료 년
5	ID(FK)	Varchar(25)		아이디

오토바이 테이블

No	컬럼	자료형	PK	비고
1	MC_SEQ	INT	O	고유번호
2	MC_NUM	varchar(25)		오토바이 번호
3	MC_CC	INT		오토바이 cc
4	YATA_ZONE	varchar(25)		대여,반납 구역
5	RENT_FEE	INT		대여 요금
6	DIRVE_FEE	INT		주행 요금
7	MC_IMG			오토바이 이미지

대여 테이블

No	컬럼	자료형	PK	비고
1	RENT_SEQ	INT	O	예약번호
2	START_TIME	DATETIME		대여시간
3	END_TIME	DATETIME		반납시간
4	MC_SEQID (FK)	INT		오토바이 번호
5	ID (FK)	INT		아이디
6	FEE	INT		금액

결제 테이블

No	컬럼	자료형	PK	비고
1	PAYMENT_SEQ	INT	O	고유번호
2	PAYMENT_FEE	INT		결제금액
3	RENT_SEQ (FK)	INT		예약번호
4	CARD_NUM (FK)	varchar(25)		카드 번호
5	ID (FK)	varchar(25)		아이디

Aduino

Send_GPSdata : GPS 모듈을 이용하여 서버에 주기적으로 위치

Connect_bluetooth : 블루투스 모듈을 통해 핸드폰과 연결

Disconnect_bluetooth : 블루투스 모듈을 통해 핸드폰과 연결을 해제

Lock/ Unlock: 솔레노이드 장치를 제어

5. Prototype 구현

< 시나리오 >

: 회원가입과 로그인을 실시한 후에 앱을 실행시킨 사용자의 GPS정보를 기반으로 주변 야타존을 지도에 마킹해 보여준다. 사용자는 자신이 이용하고 싶은 위치에 개설되어 있는 야타존에서 오토바이의 상태를 조회하여 사용할 수 있다. 그 후 이용시간이 되면, 블루투스 통신이 가능하게 되어, 스마트폰 단말에서 어플의 스마트키 기능을 이용하여 키박스에 부착되어있는 잠금장치를 제어할 수 있게 되고, 오토바이를 이용할 수 있다.

< 구현된 부분과 아직 구현되지 않은 부분 >

	구현된 부분	아직 구현되지 않은 부분
SW	로그인, 회원가입 지도에 자신의 위치 지도에 야타존 마킹 야타존에 있는 오토바이 조회 로그인된 아이디와 오토바이를 통해 예약 예약시간을 통해 예약	APP 디자인 주행 요금에 대한 코딩 예약 중복을 막는 코딩 오토바이의 예약 관련된 DB
HW	안드로이드와 아두이노 블루투스 통신 블루투스 통신을 통해 락, 락해제 기능	GPS 통신을 통해 움직인 거리 측정 안드로이드와 아두이노 GPS 통신
비고	스트리밍 기능 -> 유튜브와 연동	

< 화면별 기능, 시나리오 설명 >

▶ 로그인, 회원가입 화면

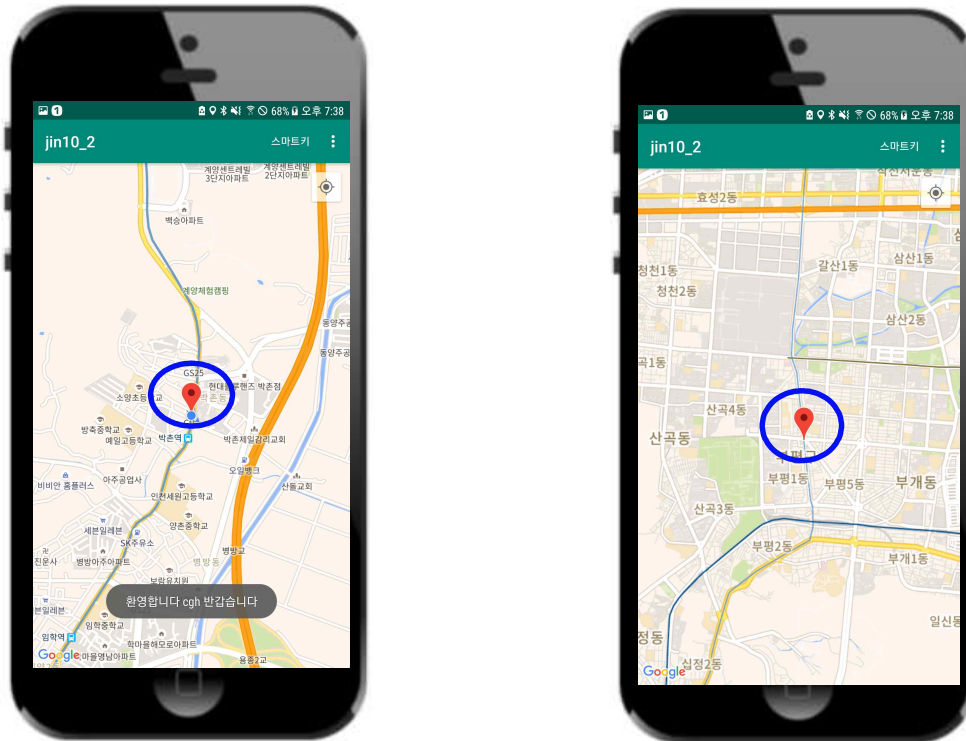


						userID	userPassword	userName	userPhone	
		수정		복사		삭제	cgh	1234	최경호	1063367184
		수정		복사		삭제	hsw	1234	한승우	105724818454
		수정		복사		삭제	qqq	qqq	qqq	01010101
		수정		복사		삭제	ee	ee	ee	ee
		수정		복사		삭제	fff	ffff	fff	0

<DB에 사용자 정보가 저장됨>

- 1) 기능 : 로그인과 회원가입의 경우 mySQL과 PHP - TOMCAT을 이용하여 구현했다. 앱의 기본화면은 우측화면이고, 필요하다면 가입하기를 눌러 회원가입이 가능하게끔 구현하였다. 아이디, 패스워드, 이름과 휴대폰 번호를 기입하게 되면 가입할 수 있으며, 아이디는 중복확인을 통해 중복된 아이디가 아니면 가능하게 된다. 또한 가입하게 되면 사용자의 정보가 mySQL에 실시간으로 저장된다.

▶ 메인화면

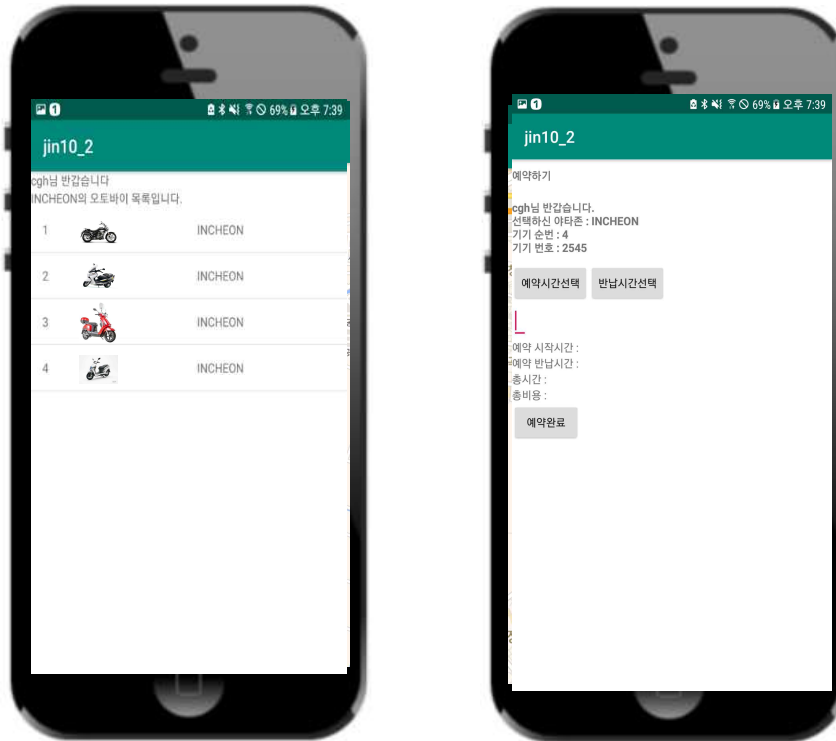


<div><div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div></div><div></div></div>							mcSeq	mcNum	yata_zone
<input type="checkbox"/>		수정		복사		삭제	1	1234	INCHEON
<input type="checkbox"/>		수정		복사		삭제	2	3234	INCHEON
<input type="checkbox"/>		수정		복사		삭제	3	6444	INCHEON
<input type="checkbox"/>		수정		복사		삭제	4	2545	INCHEON
<input type="checkbox"/>		수정		복사		삭제	5	5324	SEOUL

<설정해놓은 야타존>

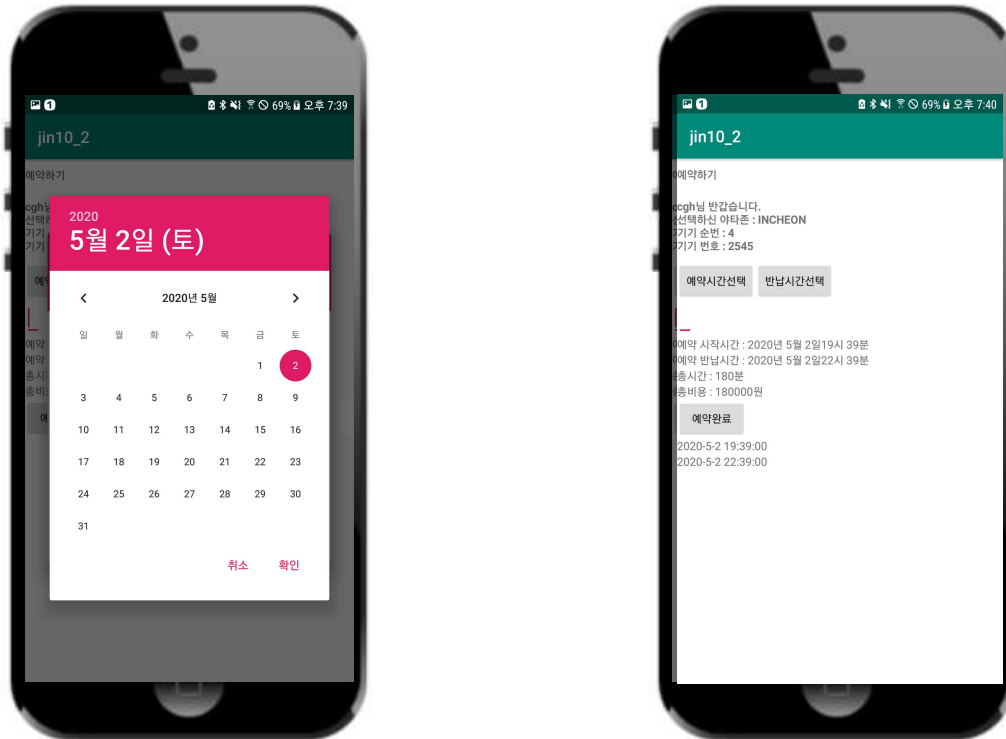
- 1) 기능 : 로그인 후에 위 사진과 같은 메인 화면으로 전환된다. 메인 화면에서는 구글맵 API를 사용하여 GPS 데이터를 받아와 사용자의 위치가 표시되고, 기존 등록되어 있는 야타존을 우측사진과 같이 확인 할 수 있다. 또한 위치정보 허용여부를 사용자가 선택할 수 있다.
표시되어 있는 야타존을 클릭하게 되면, 야타존에 존재하는 오토바이 목록을 조회할 수 있다.

▶ 오토바이 조회 화면



- 1) 기능 : 야타존에 존재하는 오토바이 목록을 조회할 수 있고, 좌측사진에서 우측에 INCHEON으로 명시되어 있는 것은 사용자가 선택한 야타존임을 명시해주기 위해 추가했다. 또한 화면에 나타난 오토바이를 탭하게 되면 선택한 야타존, 기기에 대한 정보를 확인할 수 있고, 예약 시간과 반납시간을 선택할 수 있다.

▶ 오토바이 예약 화면

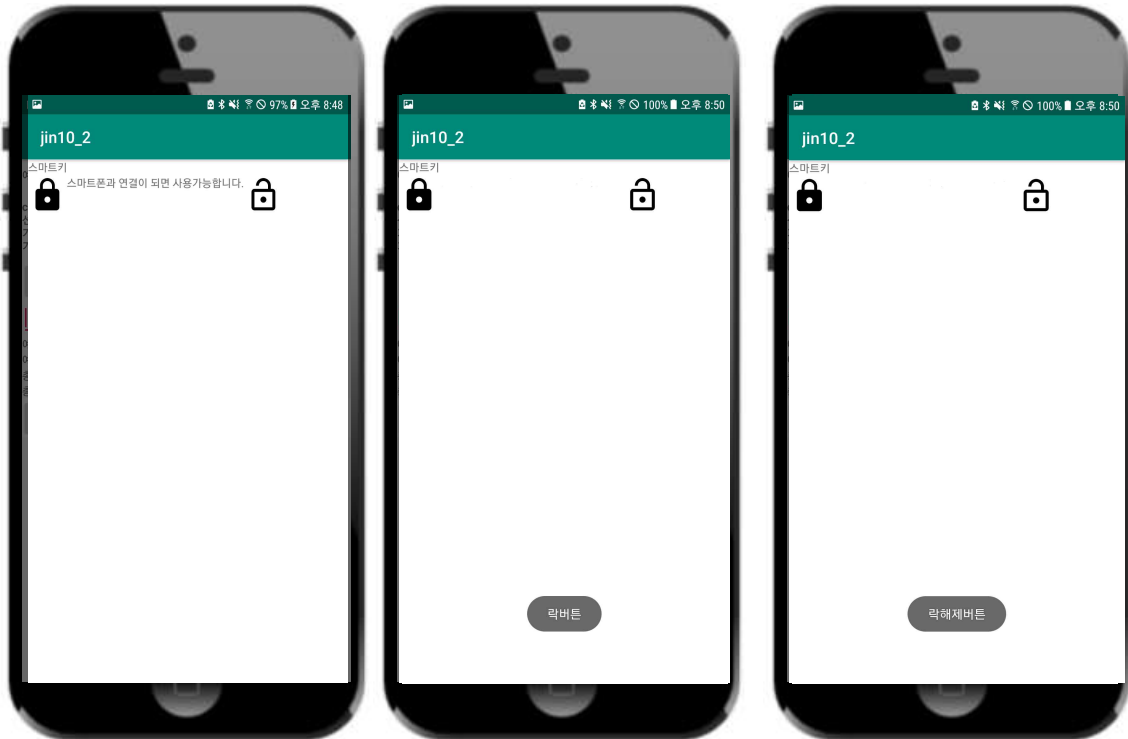


	rentSeq	startTime	endTime	rentFee	userID	mcSeq
수정 복사 삭제	1	2020-05-01 18:00:00	2020-05-01 20:00:00	2000	cgh	1
수정 복사 삭제	3	2020-05-02 16:48:00	2020-05-02 16:48:00	2000	cgh	1234

<DB에 저장된 예약정보>

- 1) 기능 : 예약시간선택 버튼을 탭하게 되면 왼쪽사진과 같은 화면이 나타나고 동일하게 반납시간선택 버튼을 탭해도 왼쪽사진과 같은 화면이 나타난다. 또한 예약시간과 반납시간이 확인가능하게 표시되고, 예약완료 버튼을 탭하게 되면 예약이 마무리된다. 예약이 마무리되면, DB에 예약 정보가 실시간으로 저장되게 된다.

▶ 스마트키 기능 화면



< 스마트키 기능 구현 동영상 >

CTRL + 클릭 or https://www.youtube.com/watch?v=YMIlgcy5_-A

- 1) 기능 : 블루투스 연결이 완료된 후 사용자의 단말은 스마트키 기능을 이용할 수 있게된다. 블루투스 연결이 완료되면 스마트폰과 연결이 되면 사용가능하다는 메시지는 지워지게 되고, 검게 색칠된 좌물쇠를 탭하면 솔레노이드 잠금장치는 잠기게 되고 반대로 풀려있는 좌물쇠를 탭하게 되면 잠금장치가 해제된다.

6. 시험/테스트 결과

<테스트 케이스>

▶ 회원가입

테스트 대상	테스트 조건	예상 결과	실행 상태
회원가입	회원가입을 할 때 ID가 중복된 상태로 회원가입을 진행한다.	Toast 메시지로 중복된 ID 값이라고 출력한다.	완성
	비밀번호가 특수문자를 포함한 8문자 미만인 채로 회원가입을 진행한다.	Toast 메시지로 특수문자를 포함한 8문자 이상 입력하라고 출력한다.	완성
	정상적으로 사용자 정보 입력한다.	DB에 저장된다.	완성
	올바르지 않은 휴대폰 번호입력한다.	Toast 메시지로 다시 입력하라고 출력한다.	미완성

▶ 로그인

테스트 대상	테스트 조건	예상 결과	실행 상태
로그인	올바른 사용자 ID와 패스워드를 입력한다.	DB에 있는 회원 데이터를 비교하여 로그인에 성공한다.	완성
	올바르지 않은 사용자 ID와 패스워드를 입력한다.	Toast 메시지로 다시 입력하라고 출력한다.	완성

▶ GPS

테스트 대상	테스트 조건	예상 결과	실행 상태
Google Maps	지도 상에 현재 위치가 출력된다.	지도 상에 현재 위치가 정확하게 출력된다.	완성
	‘야타존(오토바이가 주차된 공간)’ 이 지도 상에 표시된다.	‘야타존(오토바이가 주차된 공간)’ 이 지도 상에 정확히 표시된다.	완성
	예약한 오토바이의 이동 경로를 통해 GPS 값이 저장된다.	GPS 값이 DB에 저장된다.	미완성

▶ 예약

테스트 대상	테스트 조건	예상 결과	실행 상태
예약	이미 예약된 시간에 오토바이를 예약한다.	Toast 메시지로 이미 사용 중인 시간이라고 출력한다.	완성
	예약 중이지 않은 시간에 오토바이를 예약한다.	Toast 메시지로 정상적으로 예약되었습니다라고 출력한다.	완성

▶ Bluetooth

테스트 대상	테스트 조건	예상 결과	실행 상태
스마트키	사용할 오토바이 근처에 가서 블루투스 연결을 한다.	정상적으로 작동하여 키박스가 열린다.	완성

7. Coding & Demo

▶ Coding

(1) 회원가입

- 아이디, 패스워드, 이름, 휴대폰 번호를 입력받아 Register.PHP 파일과 연동하여 DB에 접근하여 데이터를 삽입한다.

안드로이드 - signupRequest.java

```
import java.util.Map;

// 회원가입 요청 받는부분
public class signupRequest extends StringRequest {

    //서버 url 설정(PHP 파일연동)
    final static String URL = "http://jin10.dothome.co.kr/Register.php";
    private Map<String, String> map;

    public signupRequest(String userID, String userPassword, String userName, String userPhone, Response.Listener<String> lis
        super(Method.POST, URL, listener, errorListener);

        map = new HashMap<>();
        map.put("userID", userID);
        map.put("userPassword", userPassword);
        map.put("userName", userName);
        map.put("userPhone", userPhone);
    }

    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        return map;
    }
}
```

안드로이드 - signupActivity.java

```

Response.Listener<String> responselister = (response) -> {
    try {
        JSONObject jsonObject = new JSONObject(response);
        boolean success = jsonObject.getBoolean( name: "success");
        if(success){
            Toast.makeText(getApplicationContext(), text: "회원등록 성공", Toast.LENGTH_SHORT).show();
            Intent intent = new Intent( packageContext: signupActivity.this, LoginActivity.class);
            startActivity(intent);
        }else{
            Toast.makeText(getApplicationContext(), text: "회원가입 실패", Toast.LENGTH_SHORT).show();
            return;
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
};
signupRequest signupRequest = new signupRequest(userID, userPassword, userName, userPhone, responselister, errorListener: null);
RequestQueue queue = Volley.newRequestQueue( context: signupActivity.this);
queue.add(signupRequest);
}

```

(2) 로그인

- 아이디, 패스워드를 입력받아 Login.PHP 파일과 연동하여 DB에 접근하여 아이디에 맞는 패스워드와 비교하여 같을 경우 메인화면으로 넘어 간다.

안드로이드 - LoginActivity.java

```

btn_login.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String userID= edit_id.getText().toString();
        String userPassword= edit_password.getText().toString();

        Response.Listener<String> responselister = (response) -> {
            try {
                JSONObject jsonObject = new JSONObject(response);
                boolean success = jsonObject.getBoolean( name: "success");
                if(success){
                    String userID = jsonObject.getString( name: "userID");
                    String userPassword = jsonObject.getString( name: "userPassword");
                    Toast.makeText(getApplicationContext(), text: "로그인 성공", Toast.LENGTH_SHORT).show();
                    Intent intent = new Intent( packageContext: LoginActivity.this, MainActivity.class);
                    intent.putExtra( name: "userID", userID);
                    startActivity(intent);
                }else {
                    Toast.makeText(getApplicationContext(), text: "로그인 실패", Toast.LENGTH_SHORT).show();
                    return;
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
        };
    }
};

```

PHP - Login.PHP

```

<?php
$con = mysqli_connect("localhost", "jin10", "rudgh0730!", "jin10");
mysqli_query($con, 'SET NAMES utf8');

$userID = $_POST["userID"];
$userPassword = $_POST["userPassword"];

$statement = mysqli_prepare($con, "SELECT * FROM USER WHERE userID = ? AND userPassword = ?");
mysqli_stmt_bind_param($statement, "ss", $userID, $userPassword);
mysqli_stmt_execute($statement);

mysqli_stmt_store_result($statement);
mysqli_stmt_bind_result($statement, $userID, $userPassword, $userName, $userPhone);

$response = array();
$response["success"] = false;

while(mysqli_stmt_fetch($statement)) {
    $response["success"] = true;
    $response["userID"] = $userID;
    $response["userPassword"] = $userPassword;
    $response["userName"] = $userName;
    $response["userPhone"] = $userPhone;
}

echo json_encode($response);

?>

```

(3) 구글맵API

- 구글 API 맵을 활용하여 핸드폰 상에 지도를 띄우고 사용자의 현재위치를 마커로 표시한다
- 마커를 통해 오토바이 있는 야타존을 표시한다.

안드로이드 - MainActivity.java

```
LocationCallback locationCallback = new LocationCallback() {  
    @Override  
    public void onLocationResult(LocationResult locationResult) {  
        super.onLocationResult(locationResult);  
  
        List<Location> locationList = locationResult.getLocations();  
  
        if (locationList.size() > 0) {  
            location = locationList.get(locationList.size() - 1);  
            //location = locationList.get(0);  
  
            currentPosition  
                = new LatLng(location.getLatitude(), location.getLongitude());  
  
            String markerTitle = getCurrentAddress(currentPosition);  
            String markerSnippet = "위도:" + String.valueOf(location.getLatitude())  
                + " 경도:" + String.valueOf(location.getLongitude());  
  
            Log.d(TAG, msg: "onLocationResult : " + markerSnippet);  
  
            //현재 위치에 마커 생성하고 이동  
            setCurrentLocation(location, markerTitle, markerSnippet);  
        }  
    }  
};
```

```
public boolean onMarkerClick(Marker marker){  
    String zone = marker.getSnippet();  
    Intent intent = new Intent( packageContext: MainActivity.this, popup.class);  
    intent.putExtra( name: "zone",zone);  
    intent.putExtra( name: "userID",userID);  
    startActivity(intent);  
    return true;  
}
```



```

//런타임 퍼미션 요청 대화상자나 GPS 활성 요청 대화상자 보이기전에
//지도의 초기위치를 서울로 이동
setDefaultLocation();

LatLng YATA_INCHEON_POSITION = new LatLng( v: 37.498374, v1: 126.722280);

MarkerOptions markerOptions = new MarkerOptions();
markerOptions.position(YATA_INCHEON_POSITION);
markerOptions.title("야타존");
markerOptions.snippet("INCHEON");
mMap.addMarker(markerOptions);
mMap.setOnMarkerClickListener(this);

LatLng YATA_SIHEUNG_POSITION = new LatLng( v: 37.344703, v1: 126.732042);
markerOptions.position(YATA_SIHEUNG_POSITION);
markerOptions.title("야타존");
markerOptions.snippet("SIHEUNG");
mMap.addMarker(markerOptions);
mMap.setOnMarkerClickListener(this);

//런타임 퍼미션 처리
// 1. 위치 퍼미션을 가지고 있는지 체크합니다.
int hasFineLocationPermission = ContextCompat.checkSelfPermission( context: this,

//여기부터는 런타임 퍼미션 처리를 위한 메소드들
private boolean checkPermission() {

    int hasFineLocationPermission = ContextCompat.checkSelfPermission( context: this,
        Manifest.permission.ACCESS_FINE_LOCATION);
    int hasCoarseLocationPermission = ContextCompat.checkSelfPermission( context: this,
        Manifest.permission.ACCESS_COARSE_LOCATION);

    if (hasFineLocationPermission == PackageManager.PERMISSION_GRANTED &&
        hasCoarseLocationPermission == PackageManager.PERMISSION_GRANTED ) {
        return true;
    }

    return false;
}

```

(3) 예약

- 선택한 야타존에 있는 오토바이와 예약 시간을 선택해서 특정 오토바이와 시간에 예약을 해서 DB에 저장 한다.

안드로이드 - popup.java

```
String TAG_image_res = "image_res";
try {
    JSONObject jsonObject = new JSONObject(mJsonString);
    JSONArray jsonArray = jsonObject.getJSONArray(TAG_JSON);

    for(int i=0;i<javascriptArray.length();i++){

        JSONObject item = jsonArray.getJSONObject(i);

        String MC_seq = item.getString(TAG_MC_seq);
        String MC_num = item.getString(TAG_MC_num);
        String yata_zone = item.getString(TAG_yata_zone);
        String image_res = item.getString(TAG_image_res);
        MCDData mcData = new MCDData();

        mcData.setMC_seq(MC_seq);
        mcData.setMC_num(MC_num);
        mcData.setYata_zone(yata_zone);
        mcData.setImage_res(image_res);

        adapter.addItem(MC_seq, MC_num, yata_zone, image_res) ;

    }

    mListViewList.setAdapter(adapter);
    mListViewList.setOnItemClickListener((parent, view, position, id) → {
        MCDData item = (MCDData) parent.getItemAtPosition(position) ;
```

안드로이드 - rent.java

```
//db에 보낼 값
endTime = ""+dp.getYear()+"-"+(dp.getMonth()+1)+"-"+dp.getDayOfMonth()+" "+"+tp.getHour()+" "+tp.getMinute()+"00";
}
}

button_real_end.setOnClickListener((v) → {
    Toast.makeText(getApplicationContext(), text: "버튼 클릭", Toast.LENGTH_SHORT).show();
    InsertData task = new InsertData();
    task.execute("http://" + IP_ADDRESS + "/Insert.php", startTime, endTime, rentFee, userID, mcSeq);

    Intent intent_rent_check = new Intent( packageContext: rent.this,rent_check.class);
    //Intent intent_smartkey = new Intent(MainActivity.this,smartkey_confirm.class);
    intent_rent_check.putExtra( name: "userID",userID);
    startActivity(intent_rent_check);
});
}
```

```

img_lock.setOnClickListener(new Button.OnClickListener() {
    @Override
    public void onClick(View view) {
        Toast.makeText(getApplicationContext(), text: "락버튼", Toast.LENGTH_SHORT).show();
        if(mThreadConnectedBluetooth != null) {
            mThreadConnectedBluetooth.write( str: "1");
        }
    }
});

img_lock_open.setOnClickListener(new Button.OnClickListener() {
    @Override
    public void onClick(View view) {
        Toast.makeText(getApplicationContext(), text: "락해제버튼", Toast.LENGTH_SHORT).show();
        if(mThreadConnectedBluetooth != null) {
            mThreadConnectedBluetooth.write( str: "2");
        }
    }
});
}

```

(4) 스마트키

- 아두이노와 블루투스 통신을 하여 열림 버튼을 누르면 특정 값을 보내고 닫힘 버튼을 두르면 특정 값을 보내 아두이노 서보모터가 동작하여 잠금, 해제가 가능하게 한다.

안드로이드 - smartkey.java (블루투스기기를 정하는 코드)

```

}
void listPairedDevices() {
    if (mBluetoothAdapter.isEnabled()) {
        mPairedDevices = mBluetoothAdapter.getBondedDevices();

        if (mPairedDevices.size() > 0) {
            AlertDialog.Builder builder = new AlertDialog.Builder( context: this);
            builder.setTitle("장치 선택");

            mListPairedDevices = new ArrayList<String>();
            for (BluetoothDevice device : mPairedDevices) {
                mListPairedDevices.add(device.getName());
                //mListPairedDevices.add(device.getName() + "\n" + device.getAddress());
            }
            final CharSequence[] items = mListPairedDevices.toArray(new CharSequence[mListPairedDevices.size()]);
            mListPairedDevices.toArray(new CharSequence[mListPairedDevices.size()]);

            builder.setItems(items, new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int item) {
                    connectSelectedDevice(items[item].toString());
                }
            });
        }
    }
}

```

안드로이드 - smartkey.java (이미지 버튼을 누르면 데이터를 아두이노에 보내는 코드)

아두이노 - 어플리케이션에서 보낸 값을 통해 서보 모터를 동작을 하게하는 코드

```
1 //블루투스 모듈(HC-06)을 이용한 서보모터(SG90) 제어
2 #include <SoftwareSerial.h> //시리얼 통신 라이브러리 호출
3 #include "Servo.h" //서보 라이브러리
4
5 int blueTx=15; //Tx (블투 보내는핀 설정)
6 int blueRx=13; //Rx (블투 받는핀 설정)
7 SoftwareSerial mySerial(blueTx, blueRx); //시리얼 통신을 위한 객체선언
8 String myString=""; //받는 문자열
9 Servo myservo; //서보객체
10
11 void setup() {
12   Serial.begin(9600); //시리얼모니터
13   mySerial.begin(9600); //블루투스 시리얼 개방
14   pinMode(13,OUTPUT); //Pin 13을 OUTPUT으로 설정 (LED ON/OFF)
15   myservo.attach(12); //서보 시그널 핀설정
16   myservo.write(0); //서보 초기각도 0도 설정
17 }
18
19 void loop() {
20   while(mySerial.available()) //mySerial 값이 있으면
21   {
22     char myChar = (char)mySerial.read(); //mySerial int형식의 값을 char형식으로 변환
23     myString+=myChar; //수신되는 문자열을 myString에 모두 붙임 (1바이트씩 전송되는 것을 모두 붙임)
24     delay(5); //수신 문자열 끊김 방지
25   }
26
27   if(!myString.equals("")) //myString 값이 있다면
28   {
29     Serial.println("input value: "+myString); //시리얼모니터에 myString값 출력
30
31     if(myString=="1") //myString 값이 'on' 이라면
32     {
33       digitalWrite(13, HIGH); //LED ON
34       myservo.write(60); //각도 60도로 움직임
35     } else if(myString=="2") {
36       digitalWrite(13, LOW); //LED OFF
37       myservo.write(0); //각도 0도로 움직임
38     } else {
39       for(int i=0;i<5;i++){
40         digitalWrite(13, HIGH); //LED ON
41         delay(300);
42         digitalWrite(13, LOW); //LED OFF
43         delay(300);
44       }
45     }
46   }
```

1. 연구 결과

본 연구에서는 직접 개발한 ‘YATA’ 어플리케이션과 함께 블루투스 모듈, GPS 센서 등을 사용해 만든 잠금 장치를 바탕으로 앱 내의 기능들을 사용해보았다. 현재 시중에 있는 대여 자전거 또는 대여 킥보드 시스템들과 함께 오토바이를 좋아하는 사람들의 니즈를 충족시켜 줄 수 있을 것이라 예상된다. 앞으로 모빌리티 산업의 전망을 보았을 때 해외의 여러 사례 등을 차용하여 앞으로 몇 년 안에 친환경적 기술을 바탕으로 한 시스템들이 상용화될 것이라고 예상한다. 실제로 국내의 모기업에서 전기 배터리를 이용한 이륜차 모빌리티 사업에 관한 소식들이 들려오고 있으니 앞으로의 전망에 주목할 필요가 있다.

2. 작품제작 소요재료 목록

순번	소요재료
1	노트북
2	3D 프린터로 제작한 잠금 장치
3	아두이노 우노+WIFI D1 R1 보드(ESP8266)
4	HC-06 (블루투스 모듈)
5	GPS 모듈
6	점퍼선
7	브레드 보드
8	서보 모터

참고자료

순번	참고자료
1	조효성, and 이혁준. "안드로이드 기반의 블루투스 디바이스 응용 설계 및 구현." 한국ITS학회논문지 11.1 (2012): 72-85.
2	이재길(Jae-Gil Lee), and 구본우(Bon-Woo Koo). "스마트폰과 GPS 모듈 기반의 위치추적 서비스." 한국컴퓨터정보학회 학술발표논문집 20.1 (2012): 15-17.
3	홍석원. "사용자의 이동정보를 활용한 클라이언트 인증 기반의 임베디드 보안 컨트롤러 설계." 디지털융복합연구 18.3 (2020): 163-169.
4	편집부(편집자). " ‘공유 전동킥보드’ 국내 동향과 그 기대효과." KISO 저널 -.36 (2019): 31-36.