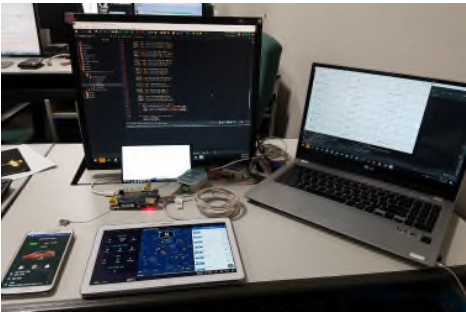
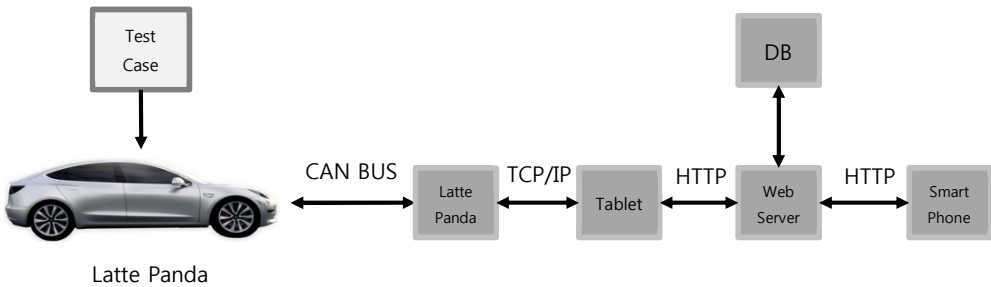


프로젝트명	전기자동차 클러스터		
개발기간	2018.05.10 ~ 2018.06.04		투입인원 4 명
시스템 구조	O/S	Windows 10, CentOS 5.0, Android3.0	
	Client	HTML5, CSS, JavaScript, jQuery, AJAX, Highcharts, Bootstrap	
	Web Server	Web server – Apache Tomcat 9.0 Development Tools – JDK 1.8.1.Eclipse Oxygen, Android Studio	
	Data Server	RDBMS – Oracle 11g DFS – Hadoop 1.2.1 Hadoop Echo System – Hive 1.0.1	
개요	<ul style="list-style-type: none">- 자동차(LattePanda)에 발생하는 데이터를 CAN통신으로 송수신한다.- 자동차에서 발생한 데이터를 TCP/IP 방식으로 받아 Tablet에 표시하고 제어한다.- Tablet은 받은 데이터를 Webserver에 http방식으로 송수신 하고 DB에 저장- Spring MVC 구현- 스마트폰으로 Webserver에 http방식으로 송수신해서 자동차를 제어한다.		
역할	프로젝트 관리	개인역량에 따른 개발 인원 모듈화로 효율성 극대화	
	분석	사용자가 자주 사용하는 값을 분석(실내온도, 전비) 해당 차량의 충전내역을 분석 현재위치 기준 가까운 충전소 추천	
	설계	Android 화면 설계, Client MVC 구조 설계, DB ERD 설계 Cent OS환경, Hadoop환경	
	구현	Login : 로그인(DB와 연동 확인, 로그인 기록 저장) Register : 회원가입 Reset : 비밀번호 재설정 Main : 클러스터(Google Map API, Contents, 차량정보) Charge : 충전 관리(충전량 설정, 충전시간 예약) Control : 차량 온도 제어, 사용자 정보(에어컨, 히터, 온도조절) Location : 현재 위치	
	프레젠테이션	프로젝트 발표용 PPT제작 및 발표	
내용			

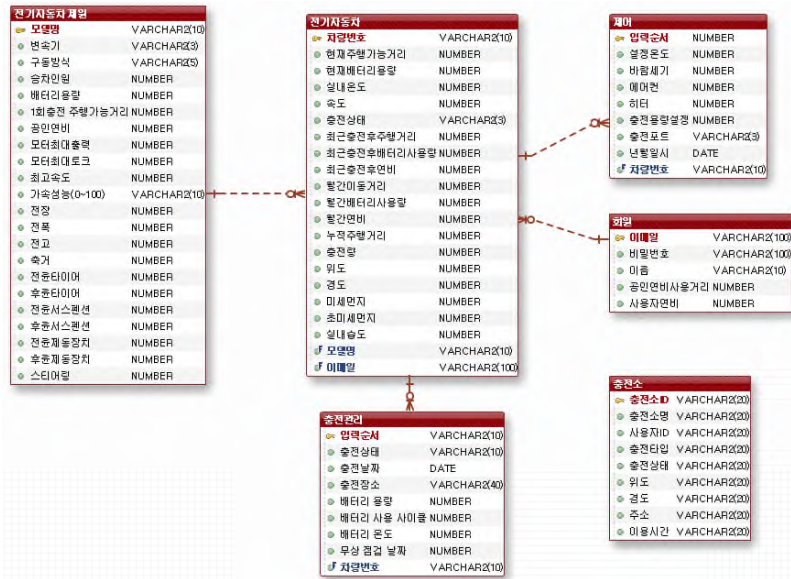
시스템 구조

IoT System	Web Server	Data Server
<ul style="list-style-type: none"> OS : Windows 10 Tool : Eclipse Oxygen JDK 1.8.0_171 Application OS : Windows 10 Tool : Android Studio 3.0 	<ul style="list-style-type: none"> OS : Windows 7 Web Server Apache Tomcat 9.0 Development Tools JDK-1.8.0_171 Eclipse Oxygen Android Studio 	<ul style="list-style-type: none"> OS : Windows 7 RDBMS Oracle 11g DFS Hadoop 1.2.1 Hadoop Echo System Hive 1.0.1

시스템 구조(시각화)



데이터베이스 구조(ERD)

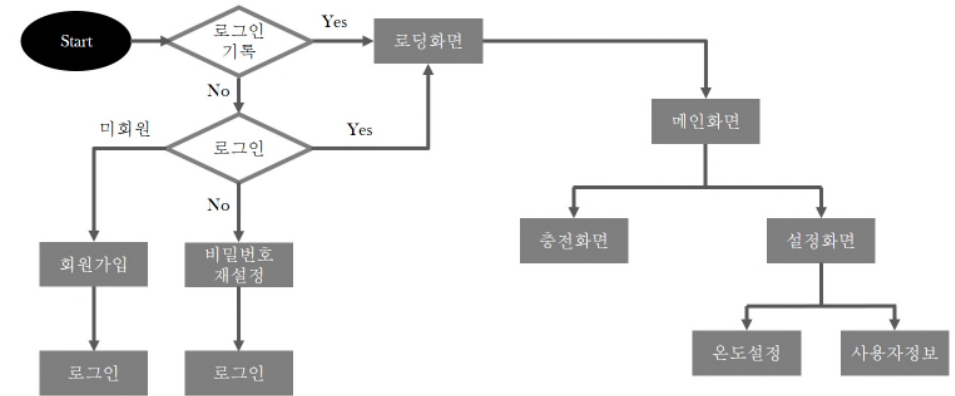


파일 구조

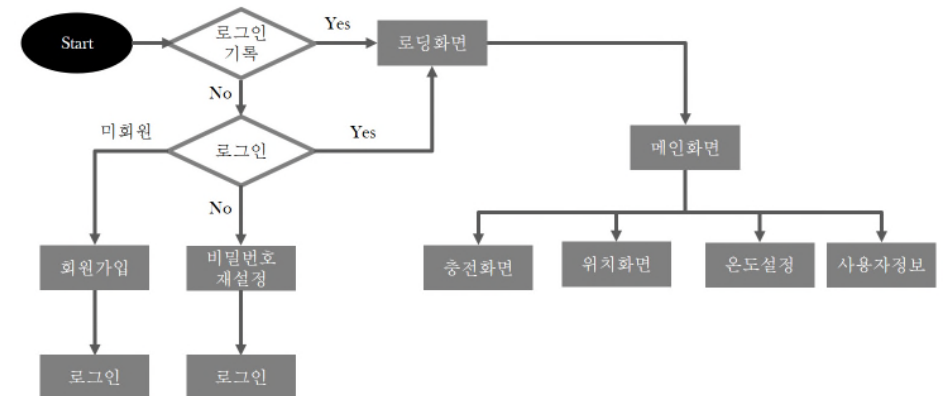
구분	파일	내용
Android App (Tablet)	LoginActivity.java	로그인 화면
	ResetActivity.java	비밀번호 재설정 화면
	SignupActivity.java	회원가입 화면
	MainActivity.java	차량정보 표시 및 목적지까지 거리 계산, 콘텐츠
	ControllActivity.java	온도 설정 및 제어 화면
	ProfileActivity.java	사용자 정보 화면
	ChargeActivity.java	충전 설정 화면(충전예약설정, 충전량 조절)
Spring MVC	MainController.java	Mybatis 로 데이터를 Get, Post 방식으로 데이터를 송수신
	Profile.jsp	Bootstrap 과 HighChart 를 활용하여 사용자 화면 구성
Android App (Smartphone)	LoginActivity.java	로그인 화면
	ResetActivity.java	비밀번호 재설정 화면
	SignupActivity.java	회원가입 화면
	MainActivity.java	메인화면(차량정보 표시- 충전량, 주행가능거리...)
	ChargeActivity.java	충전 설정 화면
	ProfileActivity.java	사용자 정보 화면
	TempActivity.java	온도 설정 및 제어 화면
	LocationActivity.java	차량 현재 위치 표시

화면 설계(Flow chart)

Tablet

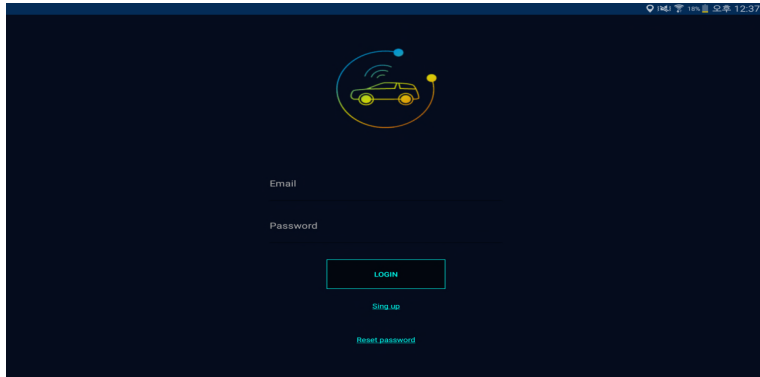


SmartPhone



실행 화면(Login)

HTTP 통신으로 서버를 통해 로그인 정보를 확인



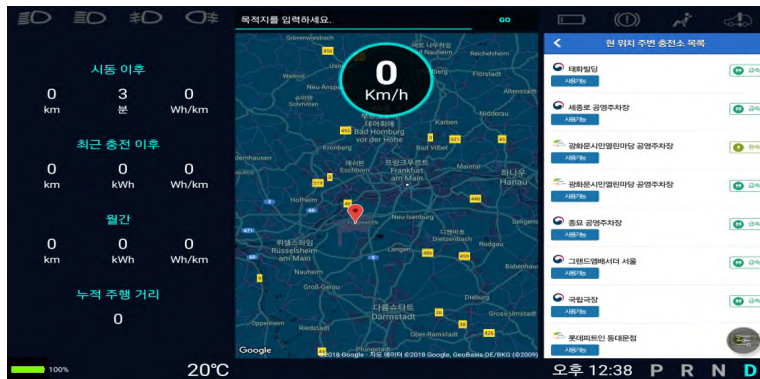
실행 화면(Main)

CAN 통신으로 받은 주행관련 데이터를 클러스터 화면으로 표시 및 데이터베이스에 저장

좌측 : 전기자동차 배터리 관련 데이터

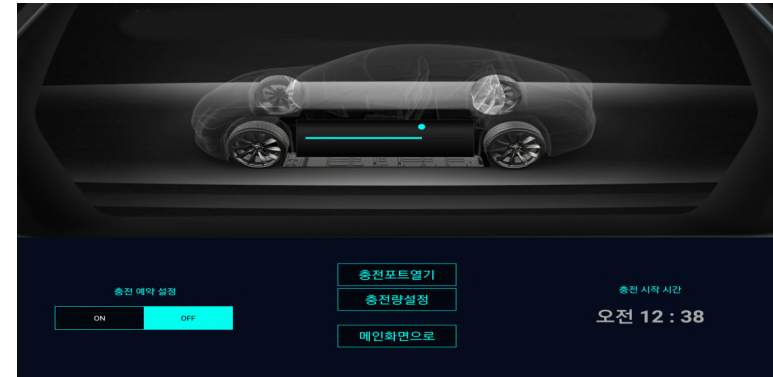
중앙 : 속도 및 자동차 위치(GoogleMap API)

우측 : 충전정보 및 컨테츠 정보를 시각화 표시



실행 화면(충전설정)

충전화면으로 배터리 최대 충전량 설정 및 충전 시간 예약기능



실행 화면 (온도 설정)

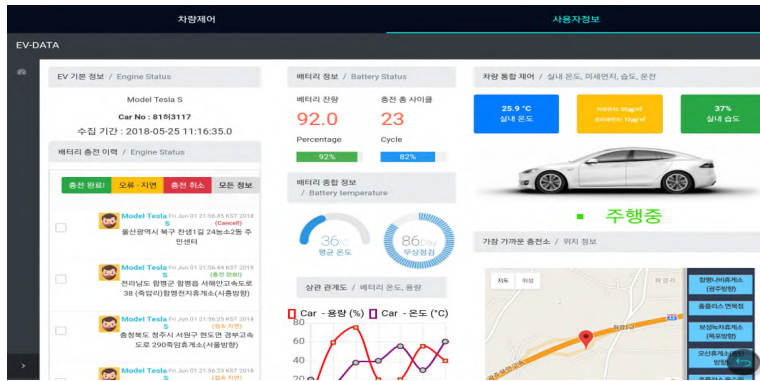
자동차 실내 온도 설정 에어컨 및 히터 ON/OFF 희망온도 설정한 데이터를

TCP/IP 와 CAN 통신으로 자동차와 양방향 통신 후 설정값 적용



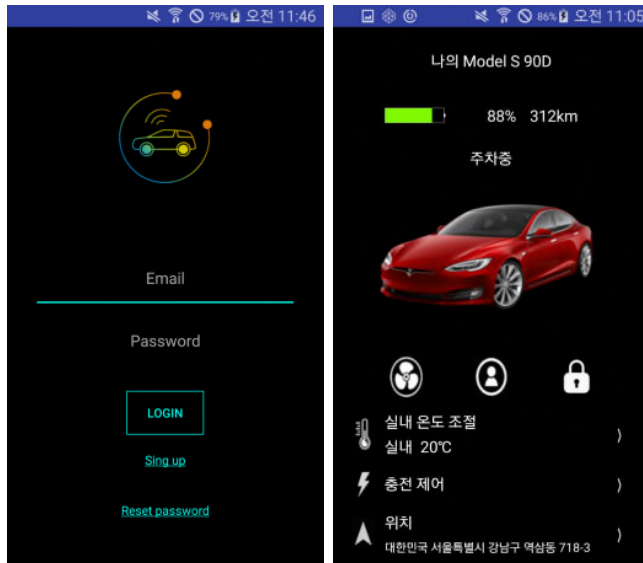
실행 화면 (사용자 정보)

자동차 배터리 충전이력, 충전 사이클, 배터리 온도, 충전소 위치등 정보를 종합해서 제공



실행 화면(로그인, 메인)

스마트폰 앱으로 웹서버를 통해서 현재 자동차의 정보제공 및 공조장치 및 도어 제어



실행 화면 (온도 설정)

태블릿 앱의 실내온도 설정 화면과 동일한 기능을 수행하며 HTTP 통신을 통해 값을 송수신

태블릿 앱과 스마트폰 앱 설정 값이 양쪽에서 조절 시 마지막 값으로 동일하게 설정됨



실행 화면(충전 및 사용자 정보)

전기자동차 충전정보를 확인 및 제어하고 사용자 정보를 제공

