
자율주행 실시간 데이터 이력 관리 APP 개발

□ APP 서비스를 통한 자율주행 데이터의 수집/가공/공유 플랫폼화

AS-IS

데이터 수집

- 1) 개인/기업/기관별 자체적인 데이터 수집 방식 이용
- 2) 자동화되지 않은 기능으로 인해 데이터 수집할 경우 번거로움이 발생함
- 3) GPS 송/수신률이 좋지 않은 음영지역이나 통신환경에 따라 데이터 공백이 발생함

데이터 가공

- 1) 수집한 데이터 확인과 관리를 위해 별도의 가공 및 추가 시각화 작업이 필요함
- 2) 자동화되지 않은 작업으로 데이터를 가공할 경우 휴먼 에러가 발생할 가능성이 존재함
- 3) 가공한 데이터의 통합과 관리의 활용성이 불편함

데이터 공유

- 1) 수집&가공한 데이터의 공유 절차 및 방법이 불편함
- 2) 별도 플랫폼의 부재로 개인/기업/기관별 데이터 현황 파악과 공유가 어려움

TO-BE

데이터 수집

- 1) 수집 데이터 항목 표준화와 기능 자동화로 효율적이고 편리한 데이터 수집이 가능함
- 2) 위치 데이터 보정을 통해 데이터의 공백을 최소화함
- 3) 다중 차량의 주행 데이터를 동시에 수집 및 처리 가능


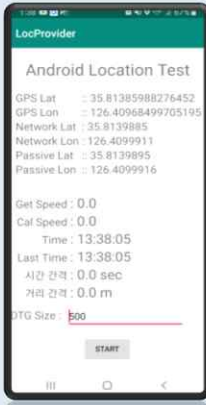
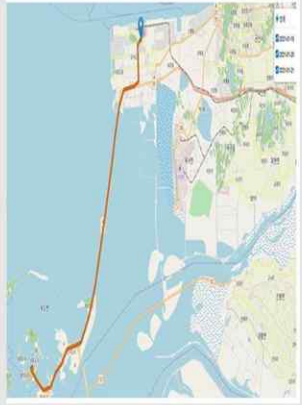
데이터 가공

- 1) 별도 추강 작업 없이 실시간으로 주행데이터를 시각화함
- 2) 휴먼 에러 가능성을 배제함으로 정확한 데이터 제공이 가능함
- 3) 차량에 따른 누적 기간/누적 주행거리 등 데이터 통합 관리 및 선별적인 데이터 확인이 가능

데이터 공유

- 1) 시간/공간적 제약 없이 데이터 현황 확인과 공유 가능
- 2) 개인/기업/기관 간 데이터 공유 및 거래 플랫폼 제공으로 새로운 BM 운용

1. 개발과제의 최종목표

구분	내용	
최종 목표	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>로그인 화면</p>  </div> <div style="text-align: center;"> <p>GPS 수집 화면</p>  </div> <div style="text-align: center;"> <p>Web 표출 화면</p>  </div> </div> <p style="text-align: center; margin-top: 10px;">[연구개발 주요 목표]</p>	
	1. 스마트 디바이스 GPS 데이터 수집 및 표출	- 스마트 디바이스 GPS 데이터와 차량 DTG 데이터를 취합하여 얻은 데이터를 Web으로 표출함
	2. 스마트 디바이스 GPS 데이터와 차량 DTG 데이터 융합	- 스마트 디바이스 GPS 데이터와 차량 DTG 데이터를 취합하여 표준편차를 구하는 알고리즘 기준을 수립하고 Web을 이용하여 그래프를 통해 표출함
주요 내용	[연구개발 주요 내용]	
	1. 스마트 디바이스 GPS 데이터 추출	- Android App을 통해 스마트 디바이스로 GPS 데이터를 추출한 데이터를 Back-end 서버로 전송함 - Log-in 기능을 통해 차량 관련 데이터를 구분하며 어느 차량의 GPS 데이터를 추출할지 결정함
	2. 데이터 시각화 Web 표출	- Android App에서 추출한 GPS 데이터를 가공하여 Back-end 서버 Django를 통해 Web 브라우저를 사용하여 map에 표출함 - REST API를 통해 차량을 구분하여 알맞은 정보를 map에 표기함 - DTG 데이터와 스마트 디바이스 데이터를 취합하여 Web 화면에 표출함
	3. 스마트 디바이스 GPS 데이터와 DTG 데이터 융합	- DTG 데이터와 스마트 디바이스 데이터를 융합하여 표준편차를 구하는 알고리즘을 설계하고 수립함

2. 개발과제의 단계별 목표

구분	기술명	개발목표	기술개발 내용 및 범위	개발기관
1차 년도	① 데이터 이력 관리 시스템 개발	○ DB 설계 및 구축	○ Data Modeling ○ Postgre SQL 구축 ○ Django Server 구축	디엔에이 모티브
	② 추출된 GPS 주행 데이터 시각화 화면 개발	○ Front Server 구축 ○ Front Server 호출	○ Python Folium을 이용한 map 시각화 정보 전달 ○ leaflet.js를 이용한 map에 표 기 및 시각화	
	③ 시험차량 데이터 구분을 위한 표출 화면 개발	○ App Login 기능 개발	○ 차량 데이터를 구분하게 만 들어주는 Data Modeling ○ Log-in을 위한 화면 개발	
	④ 실시간 데이터 전송 및 저장 시스템 구축	○ Back-end Server 구축	○ Django를 이용하여 REST API 구현	
	⑤ 자율주행 데이터 실시간 이력 관리용 APP 개발	○ Android App 개발	○ Android Studio를 이용 한 App 개발	

□ 단계별 상세 개발 내용

① 자율주행 데이터 실시간 이력 관리 APP 개발

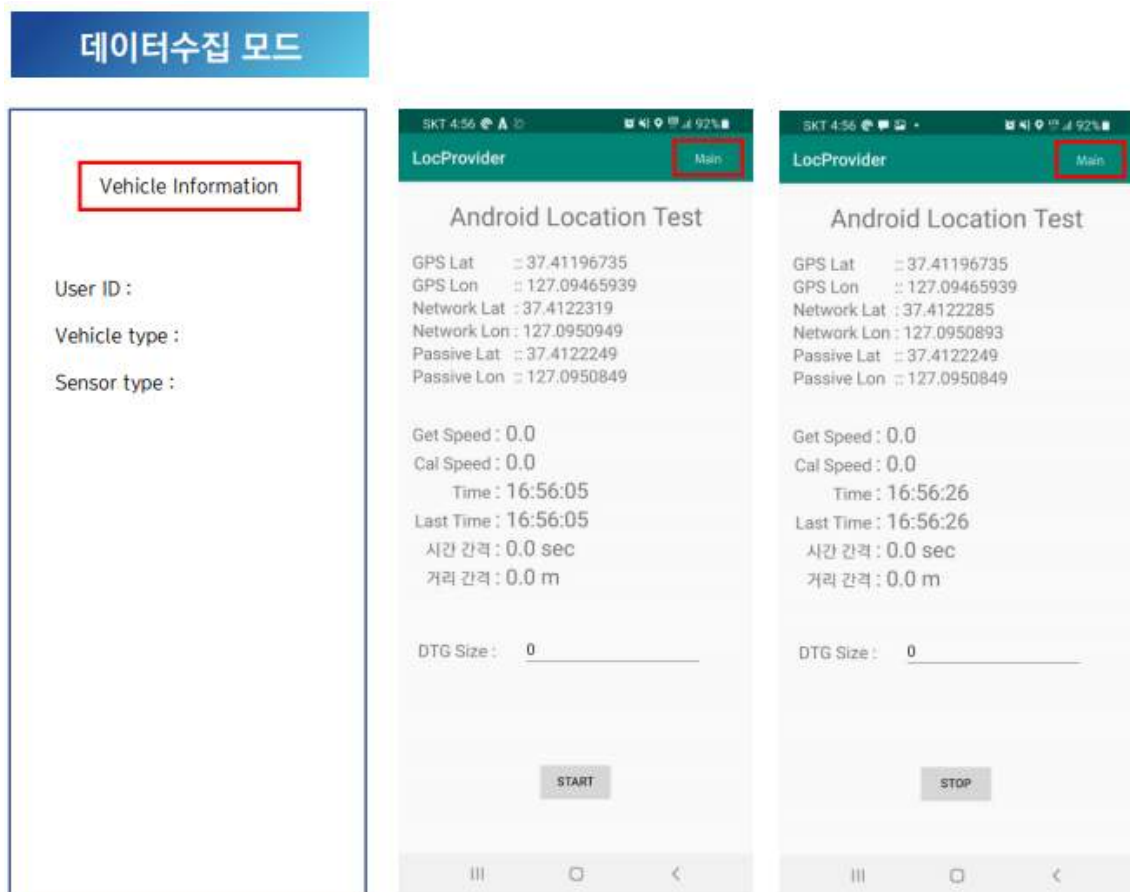
○ WEB/APP 메인 화면

- APP 실행 시 첫 메인 화면은 Open UI(스프링콩 BI) 후 로그인 페이지 표출
- 로그인 페이지와 회원가입 페이지로 분리하고 관련 메뉴 제공
- 회원가입 시 ID>Password 외 e-mail 정보를 추가로 수집하여 ID,PW 분실 시에 e-mail를 통한 확인 및 재설정을 위한 UI 구현
- ID>Password 찾기 UI 구현
- 로그인 후 모드(데이터 수집mode, 모니터링 mode) 선택 페이지로 이동

- * 데이터수집모드: GPS/ 속도 등 데이터 추출과 Back-end 서버로 실시간 전송
- * 모니터링모드: DTG정보와 스마트디바이스 간의 실시간 정보를 시각화하여 표출

○ 데이터수집 모드

- Input 정보 입력창을 표출하여 User ID(차량별 부여 ID)를 입력
 - * 신규 입력 및 계정정보에 따른 기존 등록 ID 선택 옵션 제공
 - * 차량타입/ 센서타입(옵션화)
- 정보 입력 후 START 버튼 입력 시 데이터 수집 시작
- STOP 버튼을 입력하면 데이터 수집을 멈추고 현재까지 수집한 데이터 저장
- Main 화면으로 돌아가기 혹은 모드 전환 기능 구현



○ 모니터링 모드

- 로그인 계정에 따른 수집ID(차량ID) 선택 옵션
- 선택한 ID에 따른 주행 시간 별 데이터 선택 옵션
- 선택한 데이터를 이용한 지도에 맵 드로잉(맵핑)과 주행 경로 시각화
- 전체 날짜(시간) 데이터 선택 시 전체 주행경로를 이어서 표출함
 - * 날짜(시간) 파일 간 위치 이동으로 경로 공백 발생 시 임의로 경로를 연결하여 표출하지 않음
 - * 저장 파일 간 데이터상에 위치 공백값은 공백으로 처리함
- 선택한 날짜(시간)에 대한 데이터 용량, 수집 시간 표시
- 선택한 날짜(시간)에 대한 날씨 정보, 주행거리 표시
- 선택한 수집ID(차량ID)에 대한 전체 누적 데이터 용량, 수집 시간 표시
- Main 화면으로 돌아가기 혹은 모드 전환 기능 메뉴

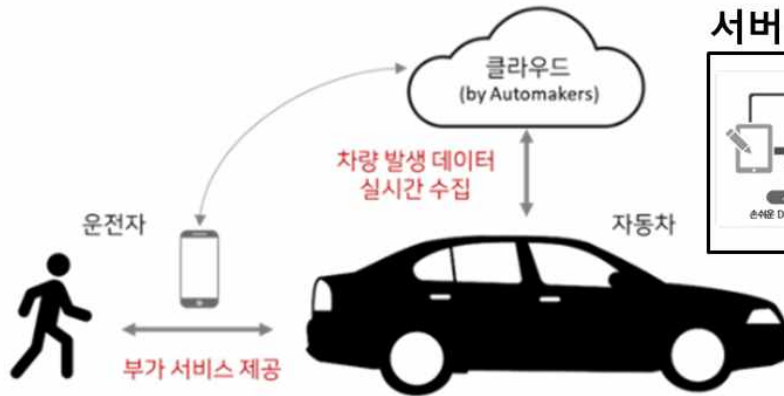
모니터링 모드



3. 개발과제의 내용

□ 데이터 이력 관리 시스템 개발

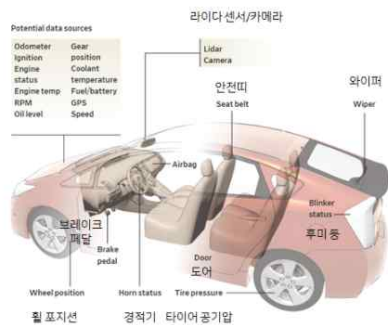
○ Database 서버 구축 및 Data Modeling



서버구축



- Data Modeling은 aquerytool을 이용하여 데이터를 분석에 용이하도록 설계

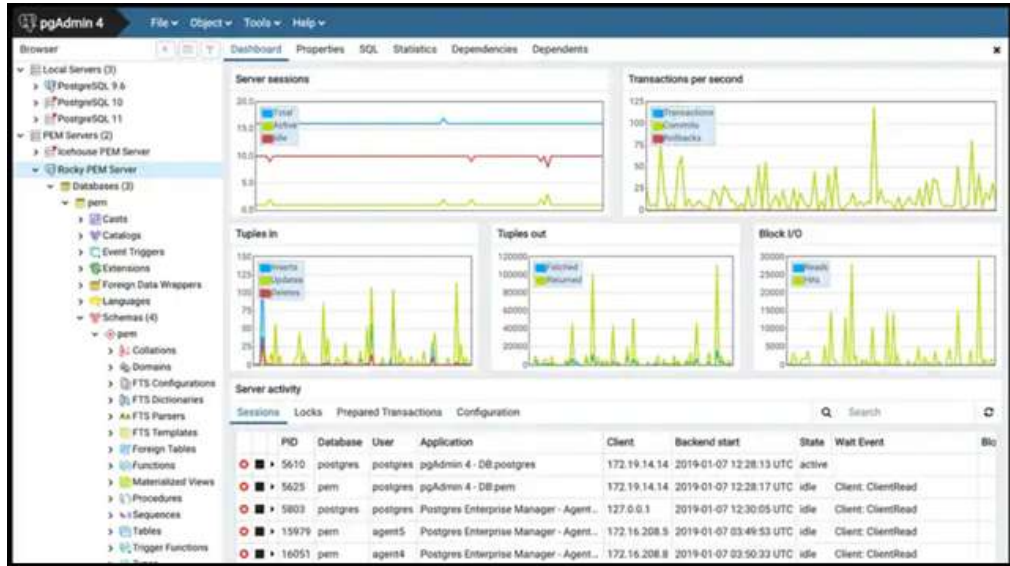


Source: Otonomo, The Wall Street Journal



`id`	INT	NOT NULL	AUTO_INCREMENT,
`account`	VARCHAR(20)	NOT NULL	COMMENT '계정 이름',
`password`	VARCHAR(50)	NOT NULL	COMMENT '비밀번호',
`kor_name`	VARCHAR(20)	NOT NULL	COMMENT '한글명',
`eng_name`	VARCHAR(20)	NOT NULL	COMMENT '영어명',
`status_id`	INT	NOT NULL	COMMENT '셀러상태(분리)',
`seller_category_id`	INT	NOT NULL	COMMENT '셀러 속성(분리)',
`created_at`	DATETIME	NOT NULL	DEFAULT CURRENT_TIMESTAMP COMMENT '셀러 프로필 이미지',
`seller_image`	VARCHAR(500)	NOT NULL	COMMENT '셀러 배경이미지',
`background_image`	VARCHAR(500)	NULL	COMMENT '셀러 소개',
`intro`	VARCHAR(50)	NOT NULL	COMMENT '셀러 상세 소개',
`detail_intro`	VARCHAR(500)	NULL	COMMENT '고객센터전화번호',
`center_phone`	VARCHAR(15)	NOT NULL	COMMENT '택배주소',
`post_address`	VARCHAR(34)	NOT NULL	COMMENT '택배상세주소',
`post_detail_address`	VARCHAR(20)	NOT NULL	COMMENT '고객센터 시작',
`center_start`	VARCHAR(8)	NOT NULL	COMMENT '고객센터 종료',
`center_end`	VARCHAR(8)	NOT NULL	COMMENT '배송 정보',
`delivery_info`	VARCHAR(200)	NOT NULL	COMMENT '교환/환불 정보',
`exchange_info`	VARCHAR(20)	NOT NULL	DEFAULT False COMMENT '마스터/셀러 0
`is_master`	TINYINT	NOT NULL	DEFAULT False COMMENT '소프트 딜리트
`is_deleted`	TINYINT	NOT NULL	
PRIMARY KEY (id)			

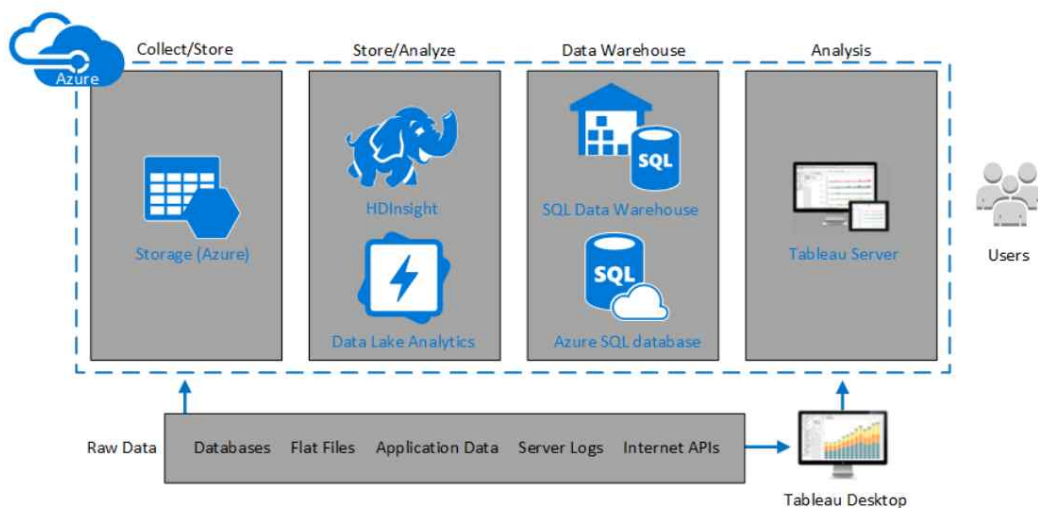
- Database 서버는 Postgre SQL 프로그램을 사용하여 구축



- Backup Schedule 설정

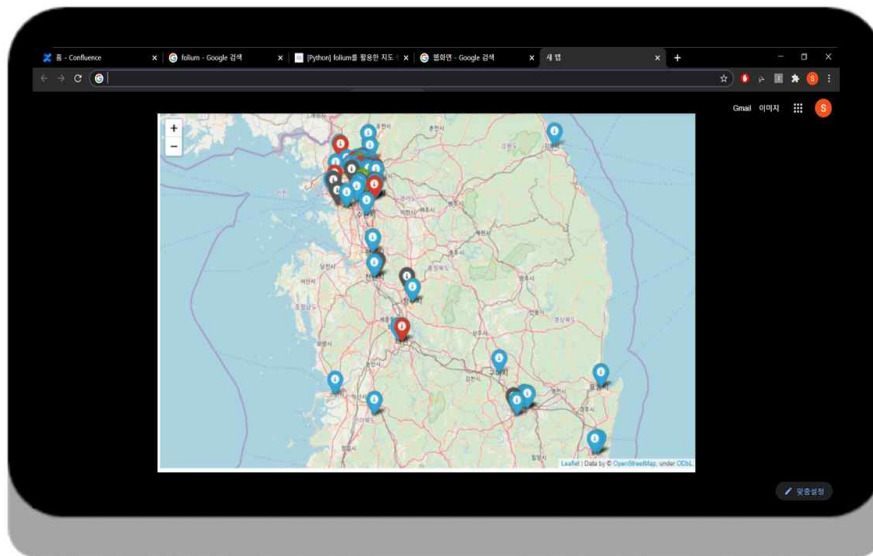


- Azure Server를 이용한 Cloud Platform 구성 및 Setting

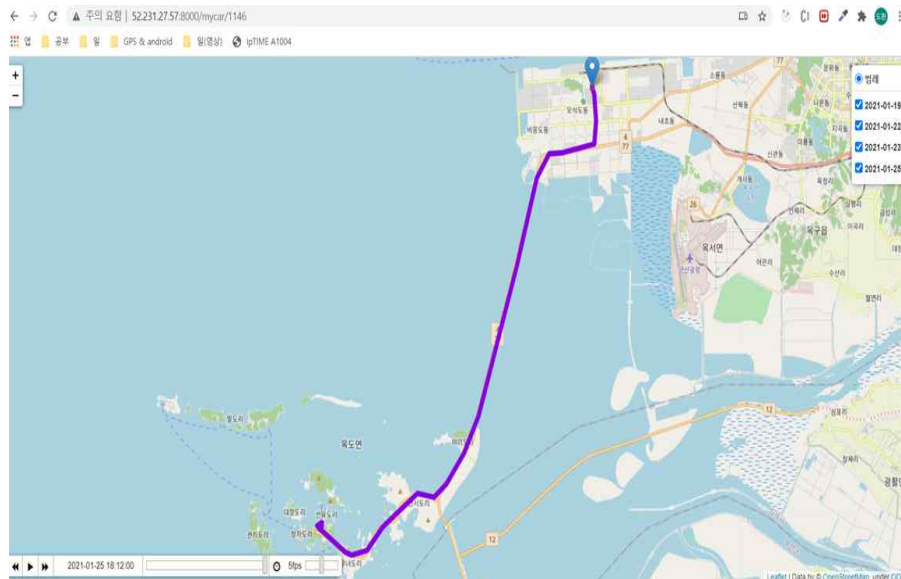


□ 추출된 GPS 주행 데이터 시각화 화면 개발

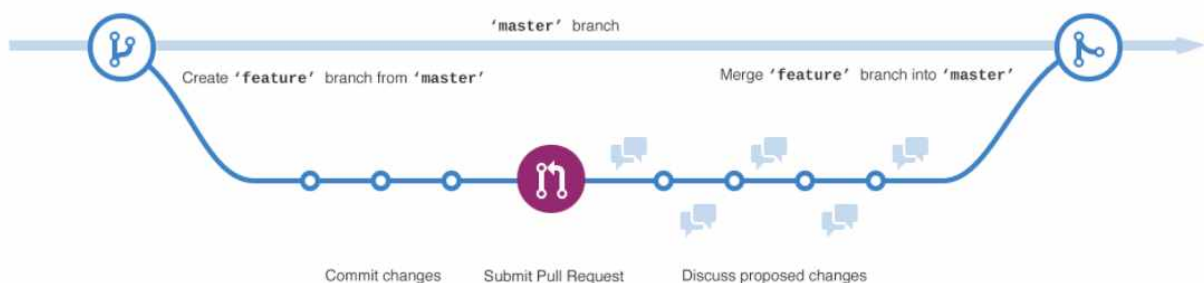
- Folium과 Leaflet.js를 이용한 Map 시각화 화면 개발



- Python의 시각화 도구인 Folium으로 Leaflet.js에 데이터를 전달하여 도식화



- Leaflet.js은 웹 Map과 상호작용을 위한 Javascript 라이브러리
- Leaflet.js와 HTML으로 Map화면 개발
- * Source code는 Github를 통한 Issue 관리



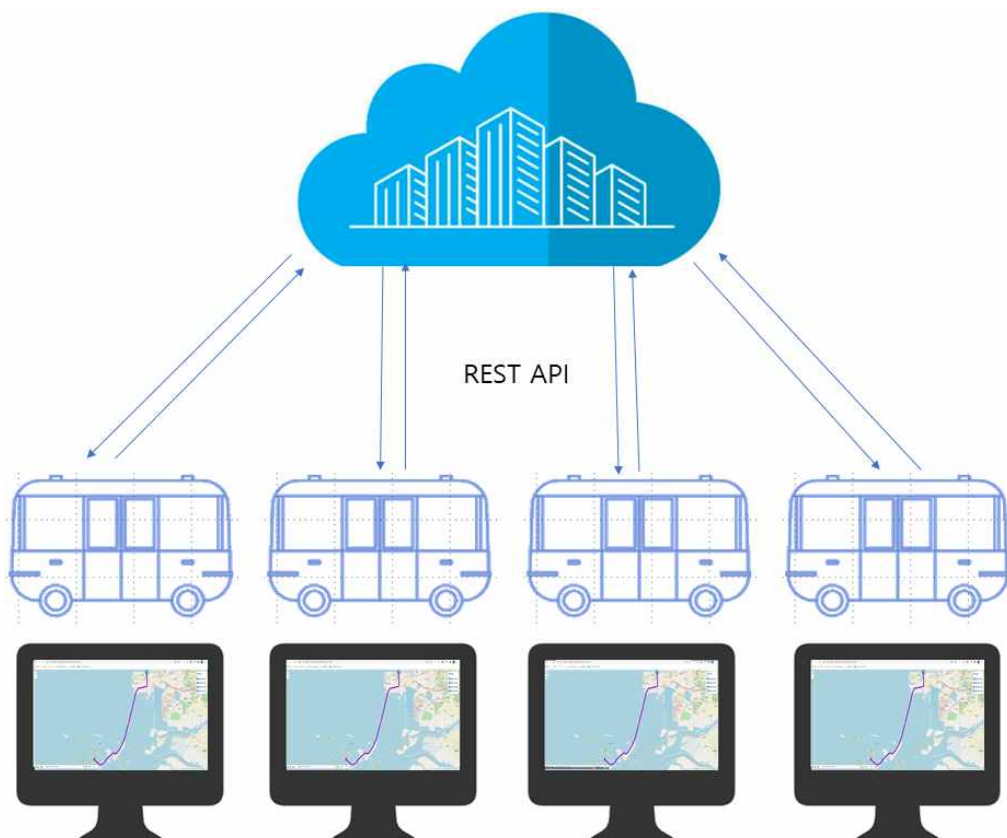
□ 차량 데이터 구분을 위한 표출 화면 개발

○ User authentication 화면 디자인



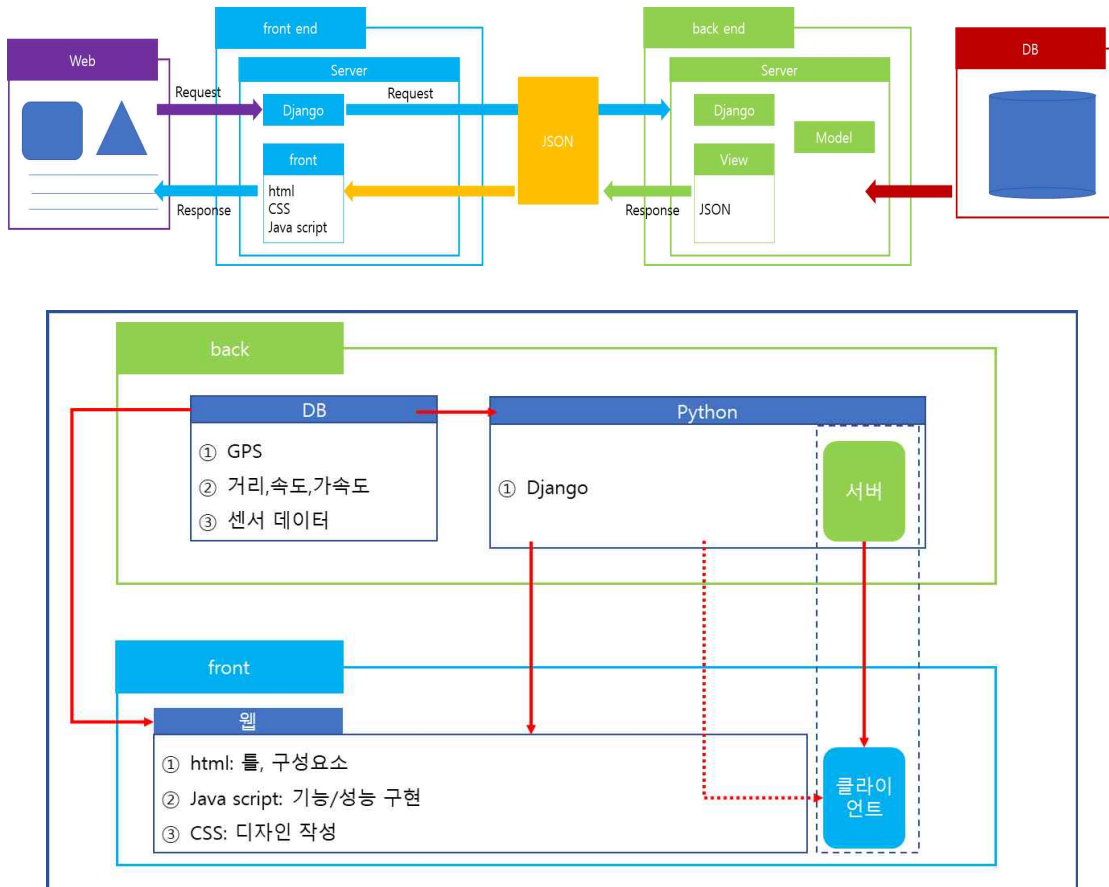
- User 및 Vehicle Location 구분을 위한 작업

* User authentication별 화면 구성을 다르게 설정

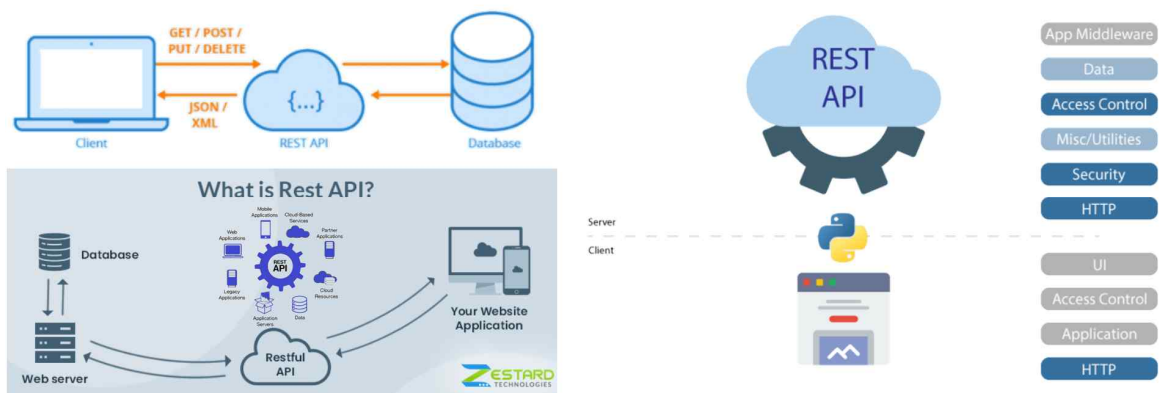


□ 실시간 데이터 전송 및 저장 시스템 구축

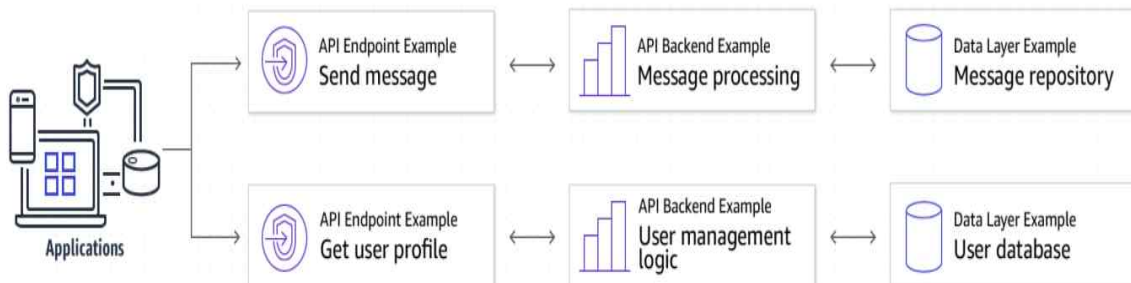
○ Back-end Server 구축



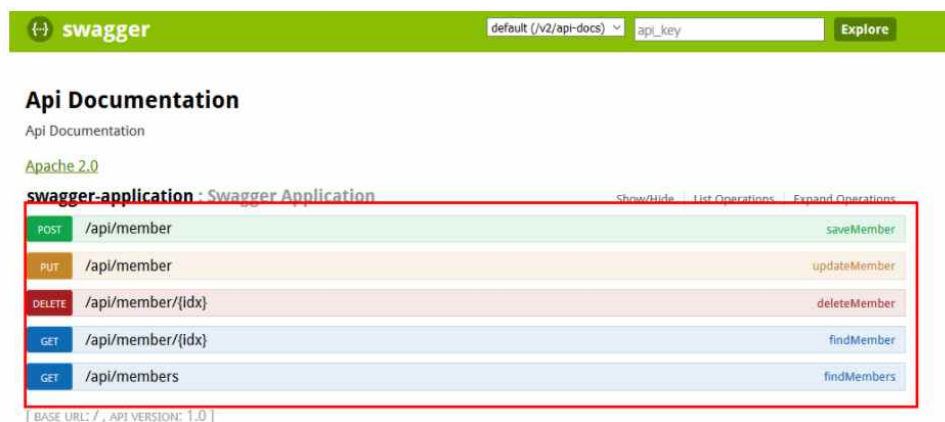
- Django를 이용하여 REST API 기반으로 개발
- REST(RESTful) API란 REST 아키텍처의 제약조건을 준수하는 애플리케이션 프로그래밍 인터페이스이며 REST는 Representational State Transfer의¹⁾ 약자임
- API 또는 애플리케이션 프로그래밍 인터페이스(Application Programming Interface)는 애플리케이션 소프트웨어를 구축하고 통합하는 정의 및 프로토콜 세트임



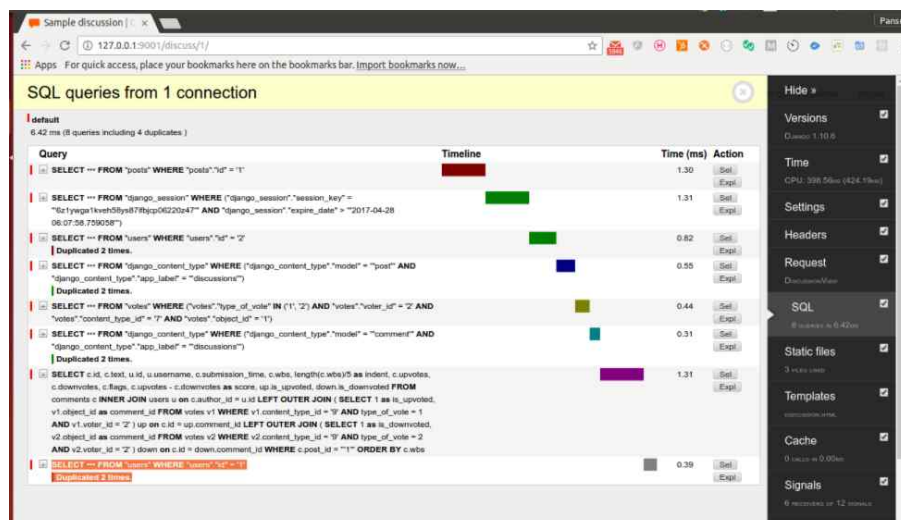
- API 관리는 개발자와 회사가 안전한 환경에서 API를 빌드하고, 분석하며, 운영하고, 확장할 수 있게 하는 도구 및 서비스 세트로 구성함
- 단순한 수준에서 API(애플리케이션 프로그래밍 인터페이스)는 개별 소프트웨어 애플리케이션 사이의 통신을 가능하게 하고 개발자는 서로 다른 회사와 서비스의 API를 연결하여 특정 결과를 실현할 수 있음



- 모니터링을 위한 RESTful 표준 방식의 API 관리 및 샘플데이터와 호출 테스트를 제공하는 기능을 주는 오픈 소스 swagger, django debug toolbar 사용
- Swagger는 간단한 설정으로 프로젝트에서 지정한 URL들을 HTML 화면으로 확인할 수 있게 해주는 프로젝트임

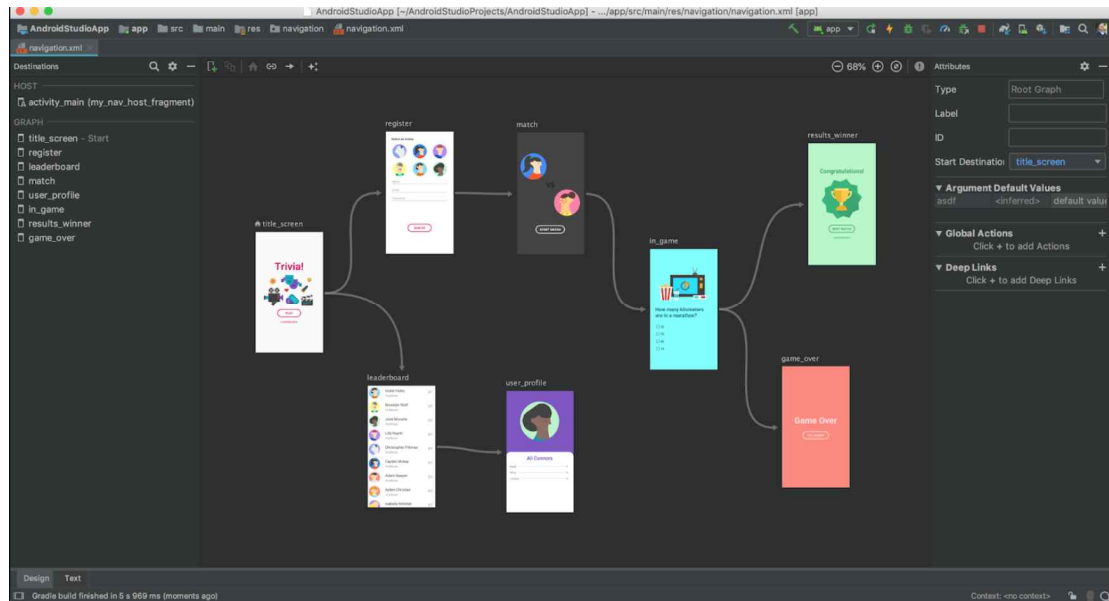


- django Debug Toolbar를 사용하면 웹 브라우저에서 해당 웹페이지에서 사용된 HTTP 헤더, Setting, SQL문들, 템플릿 계층 구조 등 매우 다양한 디버깅 정보를 쉽게 파악할 수 있음



□ 자율주행 실시간 데이터 이력 관리용 APP 개발

○ Android Studio를 이용한 APP 개발



- Java 기반으로 Login 화면 개발 및 GPS Location 정보 추출 화면 개발

* 스마트디바이스는 GPS provider를 기반으로 데이터를 가져와 속도 계산을 적용

