**차량 데이터 생성 시뮬레이터 개발**

이 프로젝트의 목표는 가상의 차량 데이터를 필요한 만큼 생성해내는 시뮬레이터를 개발함으로써, 차량 IoT 플랫폼을 개발하는데 있어서 부족한 테스트 데이터를 보충해주는 것이다. 개발 기간은 총 4달(9~12월)이며, 목표를 세분화시켜 단계별로 진행해 나갈 계획이다.

**Overview**

**Problem**

1. **실제 차량 및 단말기 사용 제약** : 차량 IoT 플랫폼 구축 시 필요한 다량의 테스트 데이터를 생성하는데 실제 차량 및 단말기를 필요한 만큼 등록시켜 사용하기에는 제약이 있다.
2. **개발 아키텍처 선정 기준이 불명확** : 시뮬레이터 개발에 사용할 개발 언어와 프레임워크, 개발 아키텍처 등을 고를 기준이 명확하지 않다.
3. **가상의 데이터라 구체화시키기 어려움** : 실제 하는 데이터를 가공하는게 아니고 아예 가상의 데이터를 뽑아내야 하기 때문에 시뮬레이터를 설계하는데 어려움이 있다.

**Goals**

1. **시뮬레이터로 가상의 차량 데이터 생성** : 가상의 차량과 OBD를 시뮬레이터에 등록시켜 필요한 만큼의 가상 차량 데이터를 생성해 Rest API형식으로 제공 해준다.
2. **다양한 아키텍처 분석 후 선정** : 다량의 데이터 생성을 위해 병렬 처리에 유리한 아키텍처를 선정하여 자원을 효율적으로 관리할 수 있도록 한다.
3. **단계별 목표 설정** : 최종 목표를 단계별로 세분화시켜 설정하고, 한 단계씩 목표를 달성해 나가며 점차 설계를 구체화시켜 나간다. (개발 기간은 총 2달 -10~11월-)

* 1차 목표 (Version 1)
* 실시간 주행 데이터와 주행이력 데이터를 생성/사용하기 위해 SmartFleet의 API와 동일한 API를 구현한다.
* 현재 SmartFleet에 등록된 실제 차량 id로 가상 데이터 생성 후 전송
* 해당 데이터를 가져와 지도에 이동경로를 그렸을 때 trip이 문제 없이 잘 그려져야 함

- 한 trip이 비효율적이게 중구난방 그려져선 안됨(최단거리는 아니어도 됨)

- 거리에 비례해서 출발시간과 도착시간 차이가 적절해야 함

* 2차 목표 (Version 2)
* 시뮬레이터에 가상 차량과 obd를 등록시킨 후, 해당 차량으로 가상 데이터 생성 후 전송
* 해당 데이터를 지도에 표시했을 때 trip이 문제 없이 잘 그려져야 함
* 3차 목표 (Version 3)
* 시뮬레이터에 다량의 가상 차량을 등록시킨 후, 동시에 다량의 가상 데이터 생성 후 전송
* 해당 데이터를 지도에 표시했을 때 각각의 trip이 문제 없이 잘 그려져야 함
* 전송 과정에 문제가 없어야 함 : 다량의 데이터를 한번에 전송하는 데 있어서 서버 쪽에 무리는 없는지 확인

**Out of Scope**

1. **실시간 데이터 보류** : ‘시작 지점’과 ‘끝 지점’을 설정해 현재 운행중인 실시간 데이터를 뽑아내는 것은 일단 보류한다.

**Context**

**Use Cases**

1. **개발환경**
   * **Spring Boot사용** : 처리속도와 언어 친숙도를 고려하여 Spring Boot를 사용하여 개발한다.
   * **Database** : 차량, 운전자, 단말기, 주행정보 등의 데이터를 join하여 조회해야 하므로 RDB(MariaDB)를 사용한다.
   * **Java** : Oracle의 Java 유료 정책에 의해 Open JDK를 사용하여 개발한다.
2. **Front-end**

* **UI** : 데이터 생성 및 삭제를 위한 사용자 화면은 web으로 한다.
* **데이터 생성 주기 설정** : trip데이터 또는 실시간 데이터가 생성되는 주기를 설정할 수 있어야 한다.(단위 : 초, 최소와 최대는 직접 설정이 가능하도록)
* **실시간 주행 데이터** : 실시간 주행 데이터 생성 시 조작도 : 차량 대수 설정 → 시작 버튼 → 주행 시작 → 종료 버튼 → 주행 종료. 시작부터 종료까지를 한 trip으로 본다.
* **주행이력 데이터** : 주행이력 데이터 생성 시 조작도 : 차량 대수 설정 → 생성 버튼 → 출발, 도착 위치 랜덤으로 한 trip씩 생성된다.
* **차량 위치 set 기능** : set하고 싶은 위치의 위도와 경도, 차량 대수 값을 설정하면 입력한 차량 대수가 전부 입력한 위치로 이동할 수 있다.

1. **Back-end**
   * **trip 데이터 한 번에 생성** : trip 단위로 운행 이력 데이터 생성 시 한 번에 생성하여 저장한다.
   * **실시간 주행 데이터 생성** : 실시간 주행 데이터 생성 시 데이터가 생성 되는 대로 저장한다.
   * **차량 위치 범위 제한** : 차량의 위치는 지정된 위도와 경도의 범위 내에서만 생성되도록 한다.
   * **Trip 데이터 객체** : ‘GPS정보(위도, 경도 등), 시간 정보, 이벤트성 데이터’가 기본적으로 들어가 있다.
   * **이전 운행 이력 시간과 안 겹치게 생성** : trip 데이터를 계속 생성할 때 이전 운행 이력 시간과 겹치지 않아야 한다.
   * **데이터 전송 후 응답 대기** : 실패한 전송에 대해 재전송 또는 삭제 여부를 선택할 수 있는 기능을 제공한다.
2. **Test**
   * 각 Version의 개발단계에서 단위테스트 계획서를 작성하고 테스트를 진행한다.
   * 각 Version 개발 완료 후 산출물 작성 단계에서 통합테스트 계획서를 작성하고 테스트를 진행한다.

**Assumptions**

1. **실제 데이터와 다름없이 가상데이터를 받아 볼 수 있어야 함** : 시뮬레이터에서 가상 데이터를 생성해 저장하고 해당 데이터는 REST API를 통해 외부에서 받아볼 수 있게 된다.(API는 SmartFleet과 동일해야 한다.) → 개발팀에서 해당 데이터를 누락없이 정상적으로 받아볼 수 있어야 한다.
2. **신뢰성 있는 데이터** : 만들어진 데이터는 신뢰성 있는 데이터여야 하며, 현실에서 만들어질 법 한 trip정보가 생성 되어야 한다.

**Tasks and Timeline**

