

# 실시간 드론 관제 웹 어플리케이션 개발

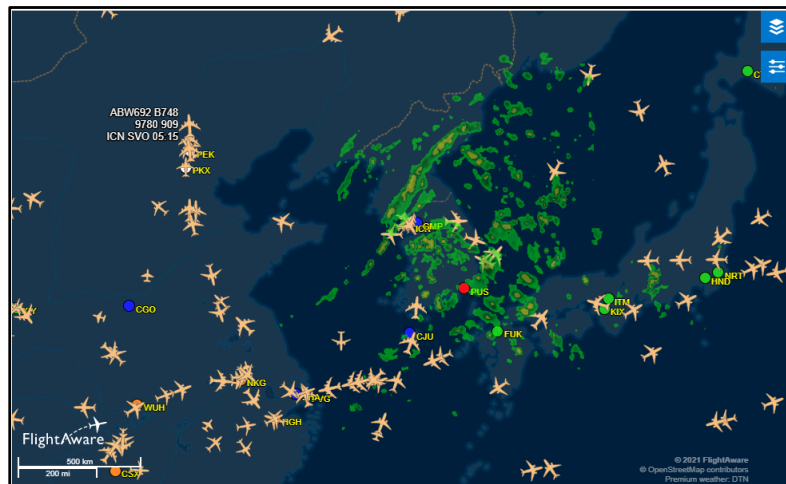
## 요약

최근 글로벌 기업인 구글, 아마존, DHL 등이 드론의 상업적 활용 사업을 선언하고 개인이 수월하게 운용할 수 있는 소형 드론이 개발되어 민수, 민간용 드론 시장이 매우 커지고 대중의 관심이 집중되고 있다. 점차적으로 상업적, 민간용으로 운용되는 드론의 개수가 증가한다면 여러 사용자가 드론의 위치 및 상태 정보를 얻을 수 있는 시스템이 필요할 것이다. 따라서 본 연구에서는 드론으로부터 실시간으로 정보를 받아와 디스플레이 하는 웹 어플리케이션을 제시하고자 한다.

## 1. 서론

### 1.1. 연구배경

무인항공기로 불리는 드론은 1900년대 초반 군사적 용도로 전투와 경찰의 목적으로 사용되기 시작하였다. 현재는 건설, 물류, 시설 관리, 재난구조, 교통 관측, 과학연구, 농업, 촬영, 취미 등 다양한 분야에서 활용되고 있고 최근 드론 관련 연구나 드론을 활용한 기술들이 쏟아져 나오고 있다. 아마존과 구글에서도 드론을 택배업에 활용할 계획을 세우고 있고 국내에서도 K-드론시스템 구축을 진행하며



드론 택배, 드론 배송 사업을 꿈꾸고 있는 추세이다.

위 사진은 항공 관제 시스템 중 하나로 전 세계에서 운행 중인 항공기의 정보를 실시간으로 모니터링 하는 시스템이다. 실시간 위치 뿐만 아니라 항공기와 항공편에 대한 정보까지 제공한다. 본 연구에서는 항공기가 아닌 드론을 대상으로 관제 웹 어플리케이션을 구축하고자 한다. 실시간으로 드론들의 위치를 추적하고 드론에 대한 정보 및 통계를 실시간으로 보여줄 것이다. 현재 본 연구를 진행하는

기업인 (주)트웬티온스에서 임무 특화 드론 훈련 시뮬레이터를 개발하여 활용하고 있다. 일반적인 상황에서 사용하는 드론이 아닌 어려운 상황이 바탕이 되는 시나리오에서 임무 특화적인 드론을 훈련할 수 있는 교육을 위한 가상 훈련 콘텐츠이다. 이후 본 프로젝트를 통하여 드론 데이터를 웹 어플리케이션에 전달하여 활용할 방안이며 더 나아가 국내 드론 정보를 등록하여 드론 관제 시스템을 구축하는 것을 목표로 한다.

## 1.2. 연구목표

사용자에게 실시간으로 드론의 위치를 보여주는 것을 목적으로 지도에 드론 위치 정보를 제공하여 사용자가 쉽게 눈으로 볼 수 있게 만들 것이며 웹으로 표현하여 어디서든 확인하도록 할 것이다.

첫 번째 목표는 지도에 드론 위치를 실시간으로 보여주는 것이다. 실시간으로 지도에 많은 데이터를 한 번에 표시하는 데 있어서 끊김이나 버벅거림을 최소화하여 사용자가 느끼기에 편하게 만드는 것이다.

두 번째 목표는 대용량 데이터를 저장하고 전송하는 데 있어서 빠르고 정확하게 데이터가 전송되어야 한다. 이를 위해 데이터를 안정적으로 저장하고 전송하는 시스템을 만드는 것이다.

## 2. 관련연구

### 2.1. PX4 Autopilot

본 프로젝트와 유사한 소프트웨어인 드론 관제 시스템은 Pixhawk의 오픈소스 PX4 Autopilot를 확장하여 개발하는 경우가 많은데, PX4 Autopilot은 오픈 소스로 만들어진 드론 또는 기타 무인 차량을 제어하기 위한 소프트웨어이며, 드론 개발자들을 위해 드론 어플리케이션 제작을 위한 유연한 도구들을 제공해준다.

이 소프트웨어는 드론 등과 같은 비행 차량에 대한 오픈 소스 소프트웨어 사용을 촉진하기 위해 만들어졌으며, Linux 재단에서 관리하는 비영리 조직인 Dronecode의 일부이다.

PX4의 주요 특징으로는 오픈 소스이며, 개발자를 위한 API와 SDK가 제공되어 있어 개발자 친화적이고 모듈식 구조를 취하고 있어 확장과 설정이 간편하다. 또한 BSD-3-Clause 라이선스 정책을 취하고 있어 상업적 사용 및 수정에 자유로우며 안정성이 보장되어 세계적으로 PX4 기반 시스템이 많이 배포되고 있다.

국내 기관들에서 운용하는 임무 전용 드론 관제 시스템은 이 Pixhawk Autopilot을 확장하여 개발된 소프트웨어를 사용하고 있다. 개별적인 드론 조종/설정 등을 하는데 효율적인 도구이나, 대량의 무인기/드론 정보 조회, 통계 기능을 제공하기에는 기능이 부족하다고 볼 수 있다.

## 2.2. 웹소켓 통신

### 2.2.1. Polling and Long Polling

Polling 방식은 브라우저가 일정한 주기로 요청하여 결과를 확인하는 방식으로 실시간 메시지 전달의 중요성에 따라 요청주기를 조절할 수 있지만, 주기가 짧아지면 서버에 무리를 줄 수 있으며, 주기가 길면 실시간성이 떨어진다.

Long Polling 방식은 브라우저 요청에 대한 응답을 서버 이벤트 발생 시점에 받는 방식으로 Polling 방식에 비해 불필요한 트래픽을 줄일 수 있으나, 이벤트가 자주 발생하는 경우에는 Polling과 다른 점이 없다.

Polling, Long Polling 방식은 데이터 전송 과정(HTTP 요청/응답)에서 불필요한 헤더 등으로 인해 네트워크 오버헤드를 발생시키며, 연결 대기시간에 따른 성능저하도 추가로 발생한다.

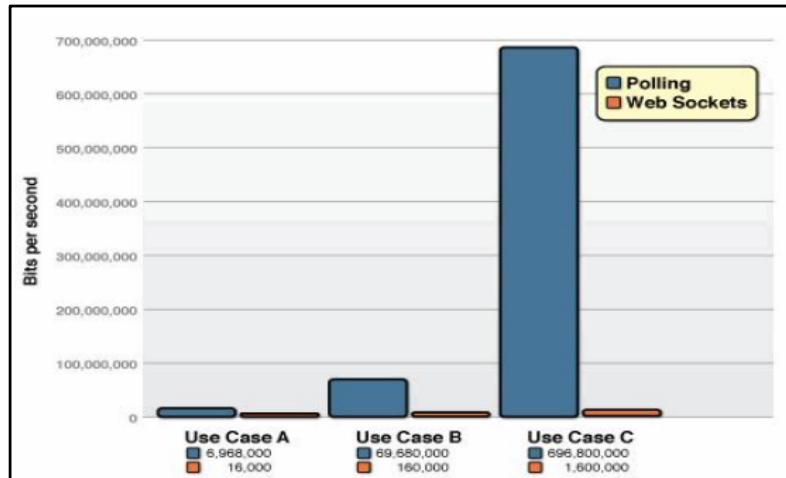
### 2.2.2. 웹 소켓(Web Socket) 통신

웹 소켓(Web Socket)은 순수 웹 환경에서 웹 클라이언트 – 웹 서버간 실시간 양방향 통신을 지원하는 프로토콜이다. 기존 브라우저의 통신 방식은 HTTP 요청(HTTP Request)에 대한 HTTP 응답(HTTP Response)을 받아서 브라우저의 화면을 지우고 받은 내용을 새로 표시하는 방식이다. 내용은 지우고 다시 그리면 브라우저의 깜빡임이 생기게 되며 이러한 깜빡임 없이 원하는 부분만 다시 그리고 실시간으로 사용자와 상호작용하는 방식이 웹 소켓(Web Socket)이다.

HTTP 프로토콜이 아닌 웹 소켓(Web Socket) 프로토콜을 사용하면 기존의 Polling 기법이나 Long Polling 기법의 불필요한 요청으로 인한 부하를 줄일 수 있다. 웹 소켓(Web Socket) 방식은 최초 연결 시 HTTP 요청(HTTP Request)의 Upgrade 헤더를 사용하여 웹 서버에 요청하고, 서버는 이를 확인하고 토큰을 생성하여 응답한다. 이러한 웹 소켓(Web Socket) 핸드셰이크 과정에서만 요청과 응답 헤더가 있고 그 이후 데이터 송수신에는 최소 2바이트 크기의 프레임을 사용하므로 네트워크의 부하가 감소하게 된다.

### 2.2.3. Polling과 웹 소켓(Web Socket) 방식 비교

웹 소켓(Web Socket) 방식은 처음 연결이 되면 연결이 지속되는 것에 비해 Polling 방식은 요청에 대한 응답이 될 때마다 재연결이 되며, 이로 인해 불필요한 네트워크 오버헤드가 발생한다.



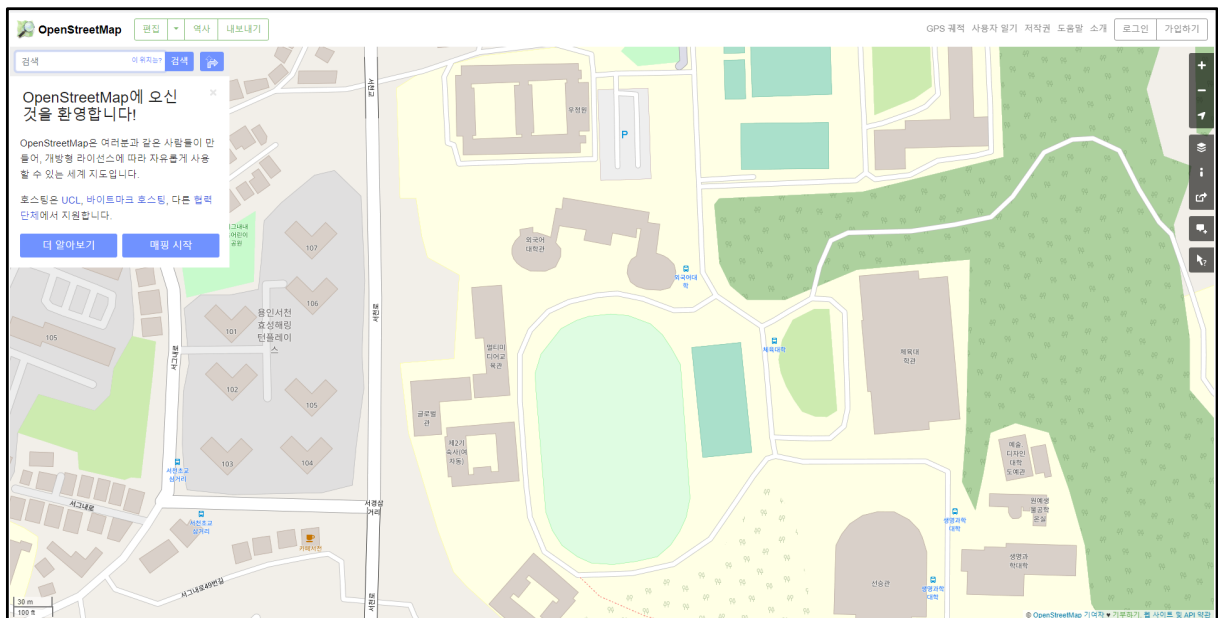
위 그림을 보면 Polling 방식과 웹 소켓(Web Socket) 방식의 대규모 이벤트에 따른 불필요한 네트워크 오버헤드 차이를 볼 수 있다.

구 분	동접 브라우저	네트워크 처리량
HTTP Request/Response (Polling)	1,000	6.6Mbps
	10,000	66Mbps
	100,000	665Mbps
HTML5 WebSocket	1,000	0.015Mbps
	10,000	0.153Mbps
	100,000	1.526Mbps

위 그림을 통해서 알 수 있듯이 Polling 방식과 웹 소켓(Web Socket) 방식의 트래픽 차이는 약 435배 정도 차이를 보이며, Polling 방식은 매번 일어나는 요청/응답으로 인해 대기시간이 추가된다.

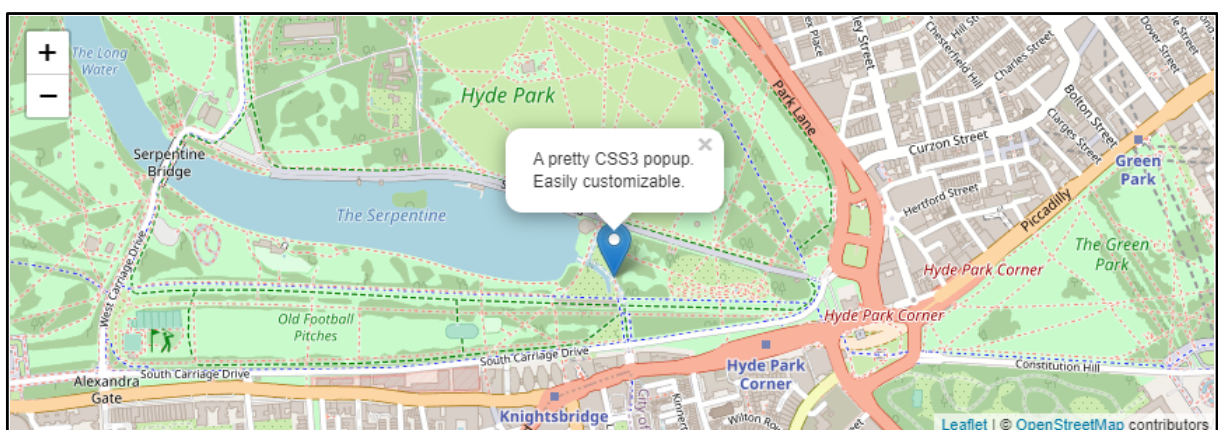
## 2.3. 지도

### 2.3.1 OpenStreetMap



OpenStreetMap(OSM)은 영국에서 출범한 오픈 소스 방식의 참여형 무료 지도 서비스이다. 개방형 공동 작업 프로젝트로 전 세계의 다양한 사람들이 집단 지성을 이용하여 만들어진다. 빠르게 업데이트 되고 무료이면서 공개적으로 사용할 수 있다는 장점이 있고 웹사이트나 GPS장치 오프라인 등에서도 지도 데이터를 사용할 수 있는 GPS좌표와 연동하는 API도 제공한다. 실제 유명한 모바일 게임인 '포켓몬 고'를 국내에서 서비스 하기 위하여 구글 지도 대신 OSM을 사용하였다.

### 2.3.2 Leaflet



Leaflet은 인터랙티브한 지도 개발을 지원하는 오픈 소스 자바스크립트 라이브러리로 Google Map, Open Street Map, Bing Map 등을 쉽게 사용하여 가볍고 간단한 Mapping을 할 수 있게 한다. Tile Map방식을 기반으로 하고 GeoJson형식을 지원하여 공간정보 자료도 지도 위에서 활용할 수 있다. 또한 다양한 Plugin이 존재하고 데스크탑 및 모바일 플랫폼에서 효율적으로 작동한다.

## 2.4. 프론트엔드

### 2.4.1. Vue.js

Vue.js는 MVVM(Model View ViewModel) 패턴에서 ViewModel 레이어에 해당하는 프론트엔드 개발을 위한 프로그레시브 프레임워크이다.

Vue.js의 특징으로는 DOM을 조작하는데 있어 가상 DOM을 사용하며, DOM에 바인딩된 데이터가 변경되었을 때, 복제된 가상 돔에서 변경된 DOM만 추적하여 상태를 변경시킴으로써 DOM 조작에 관한 오버헤드를 최소화 시킨다. 그리고 Vue에서는 기본적으로 컴포넌트간 통신이 단방향 흐름을 가지며 추가적인 설정을 통하여 양방향 데이터 바인딩을 구현할 수도 있어 데이터 조작에 있어 이점을 가진다.

또 다른 특징으로는 상태 관리 및 라우트 관리를 위한 Vue.js 공식 라이브러리가 있으며, Vue 프레임워크와 같이 View 레이어에만 초점을 맞추어 다른 라이브러리나 기존의 프로젝트와의 통합이 매우 간편하다. 또한 템플릿 기반 마크업 언어를 사용하여 HTML을 따로 수정 없이 사용할 수 있으며 문법이 간결하고 초기 학습 러닝커브가 낮아 기존의 웹 개발자들이 사용하기 용이하다는 장점을 가진다.

그리고 React, Angular와 같은 다른 주요한 프론트엔드 프레임워크에 비해 상대적으로 기본 구성요소가 간단하여 상대적으로 가볍고, 렌더링 속도가 빠르다는 이점을 가져 stateofjs에서 진행한 지난 5년간의 자바스크립트 개발자 설문에서 80~90%의 개발자가 만족하는 결과를 가지고 있다.

### 2.4.2. Nuxt.js

Nuxt.js는 Vue를 기반으로 한 웹 개발을 좀 더 쉽고 강력하게 만들어주는 오픈소스 프레임워크이다. Nuxt의 주요 특징은 추상화된 Nuxt 전용 설정을 통하여 Vue, Webpack, Babel, Server 등의 웹 어플리케이션을 만들기 위한 설정을 간편하게 할 수 있다. 기본적으로 설정된 값은 일반적인 상황에서 Vue.js 웹 어플리케이션을 개발하기 위한 최적의 값으로 되어 있으며, 모든 설정 값은 확장 및 오버라이드가 가능하다.

Nuxt의 가장 큰 장점은 SPA(Single Page Application)으로 구성된 Vue 어플리케이션을 SSR(Server Side Rendering)로 프로젝트를 구성할 수 있다는 것이다. SSR을 통하여 Nuxt 프로젝트는 사용자의 초기 진입 경험을 개선할 수 있고, SEO(Search Engine Optimization)을 설정할 수 있다. Nuxt는 프로젝트를 생성할 때 SPA, Universal, SSG(Static Site Generate) 모드를 선택할 수 있는데 Universal 모드는 최초의 페이지 접근 시 서버에서 렌더링하여 로드되고 Rehydration 이후에는 SPA의 방식을 취하여 SSR과 SPA의 장점을 모두 취하고 있다. 그리고 Universal 모드로 프로젝트를 구성할 시, 프론트엔드 서버가 활성화 되어 http 통신에서의 header 등의 서버 레벨에서 설정할 수 있었던 것들을 프론트엔드 개발에서도 가능해진다.

또한 Nuxt는 Vue 어플리케이션과 달리 정해진 폴더 구조가 있으며 Nuxt에서 정해놓은 규칙대로 개발할 필요성이 있다. 이 규칙 중 가장 특징적인 부분이 파일 시스템 기반 라우팅과 저장소 구조인데,

생성한 폴더 및 파일의 이름을 기반으로 계층적 라우트 및 저장소 구조를 자동으로 생성해주어 해당 설정의 수고로움을 덜어준다.

이러한 Nuxt의 장점으로 많은 개발자들이 웹 프론트 엔드 개발의 프레임워크로 Nuxt를 채용하고 있으며 stateofjs의 2020년 자바스크립트 프레임워크 설문결과에서 88%의 높은 만족도를 보여주었다.

## 2.5. 백엔드

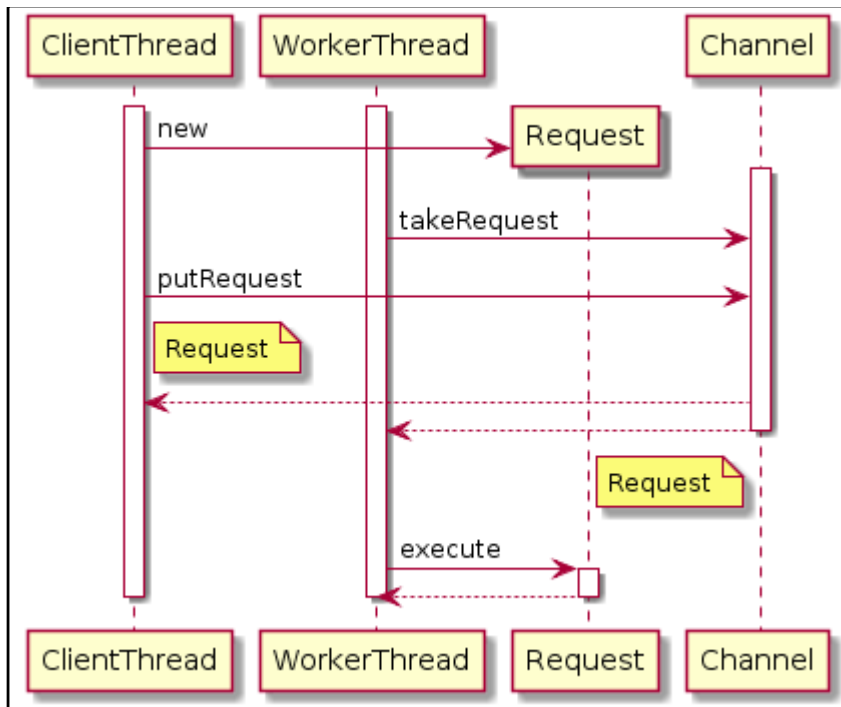
### 2.5.1. 요구사항

국내 모든 드론을 실시간으로 사용자에게 보여줘야 한다. 또한 드론의 위치, 경로, 속도 등 해당 드론의 상세 정보들을 담고 있어야 한다.

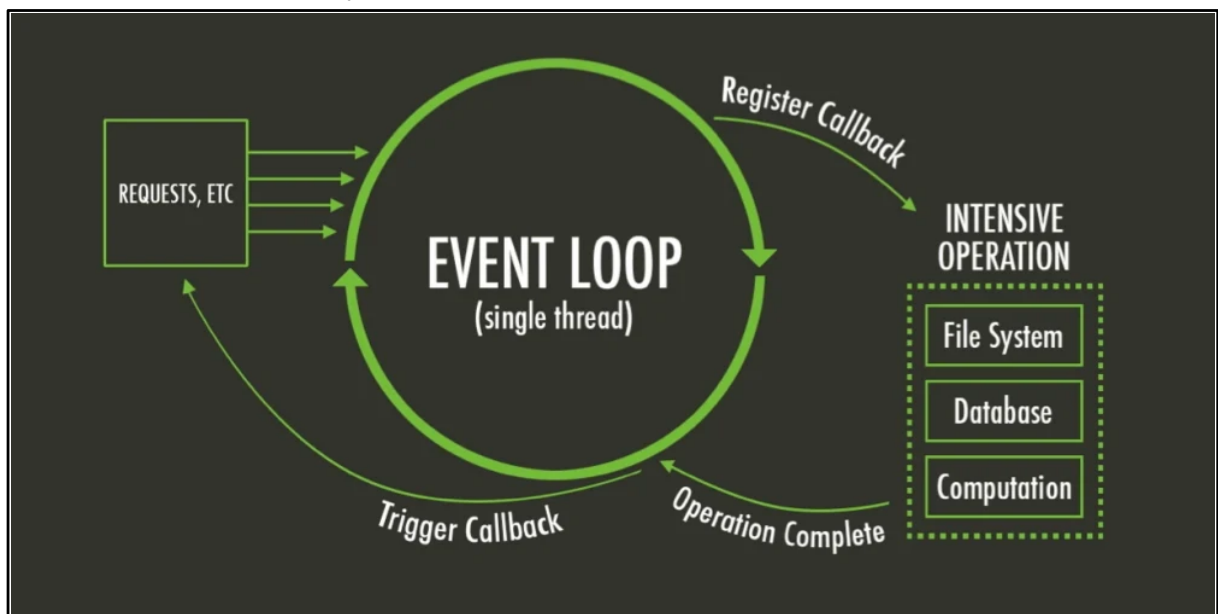
### 2.5.2. Node.js Event loop

대용량의 데이터를 실시간으로 전송하면서 빠르고 손실없이 제공하며, 또한 기록되어야 한다.

하나의 프로세스에 있어서 작업 처리시간이 오래 걸리거나, 다양한 곳에서 같은 로직을 처리해야 하는 상황이 발생한다. 이러한 상황에서는 Producer/Consumer패턴, Worker Thread 패턴을 사용하여 큰 데이터 흐름을 사용자에게 오랜시간동안 기다리지않고 뒤에서 처리하는 패턴들이 있다. 대용량의 데이터는 사용자에게 전달되어야 하며 전달되는도중 해당 데이터는 추가적으로 데이터베이스에 저장되어야한다. 데이터베이스에 저장되는 시간동안 사용자에게 데이터 전달에 있어서 시간이 지연 될 것이다. 데이터를 전송하는 프로세스에서 데이터를 저장하는 기능은 worker로 전달만 하여 데이터 전송지연이 없도록 처리하며 worker는 데이터 전송과는 별개로 저장하는 작업을 진행한다.



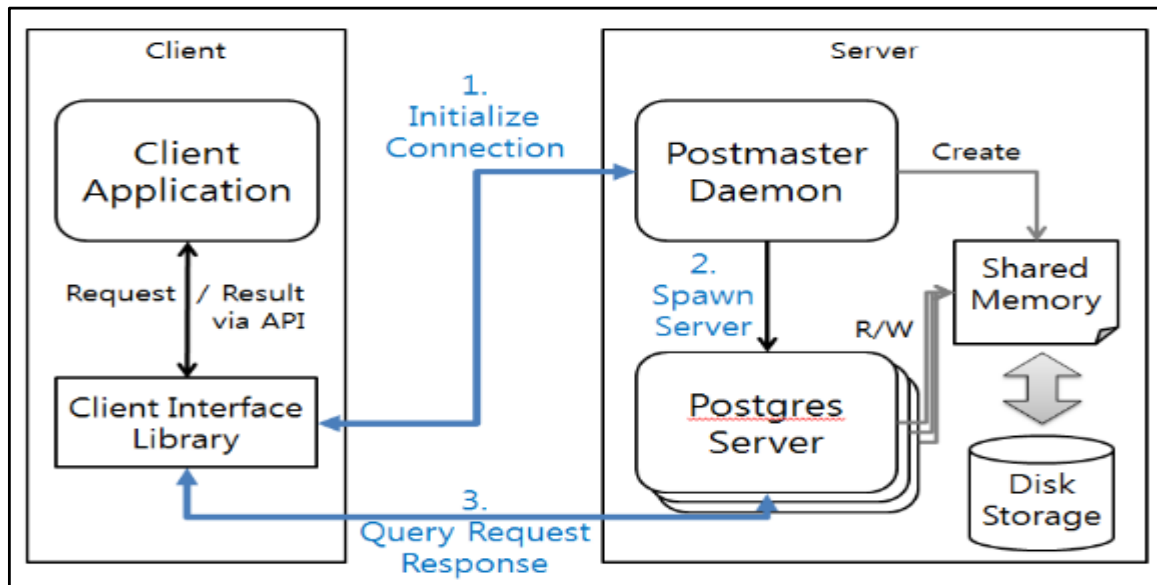
이러한 패턴들에 대해서 Node.js는 Event Loop라는 기능을 제공한다. Node.js는 Single Thread 기반의 Non-Blocking I/O 방식으로 이와같은 패턴을 코드로 해결할 수 있다. 데이터를 제공하면서 기록하는 부분을 별도의 Event Loop로 전송하여 별도로 데이터를 저장할 수 있다.





### 2.5.3. PostgreSQL

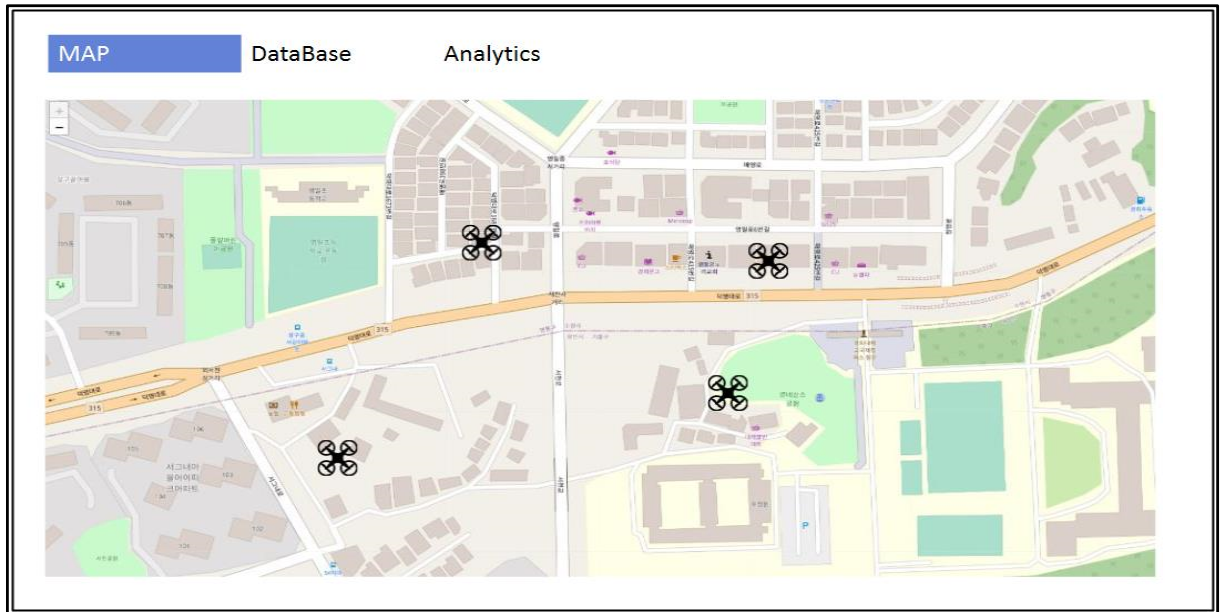
PostgreSQL은 객체관계형 데이터베이스 관리 시스템(ORDBMS)이다. 이것은 미국 캘리포니아에 있는 버클리 대학 컴퓨터 과학부에서 개발한 POSTGRES, Version 4.2를 기반으로 개발되었다. POSTGRES는 훗날 몇몇 상용 데이터베이스에 영향을 준 많은 개념들을 개척했다. PostgreSQL은 이 원래의 버클리 소스를 기반으로 확장된 오픈 소스이다. 이것은 표준 SQL 기능을 대부분 지원하고 있다. 또한, 복합 쿼리, 참조키, 트리거, 업데이트 가능한 뷰, 트랜잭션, 다중 버전 병행 제어 등의 진보적인 기능도 제공하고 있다. 그리고 PostgreSQL에서는 다양한 방법으로 기능을 확장 할 수 있다. 예로, 사용자는 자료형, 함수, 연산자, 집계 함수, 인덱스 방법, 프로시저 언어등을 직접 새로 만들 수 있다. PostgreSQL의 프로세스 구조를 간단히 살펴보면 클라이언트가 인터페이스 라이브러리를 통해 서버와의 연결을 요청하면, Postmaster 프로세스가 서버와의 연결을 중계한다. 이후 클라이언트는 할당된 서버와의 연결을 통해 질의를 수행한다.



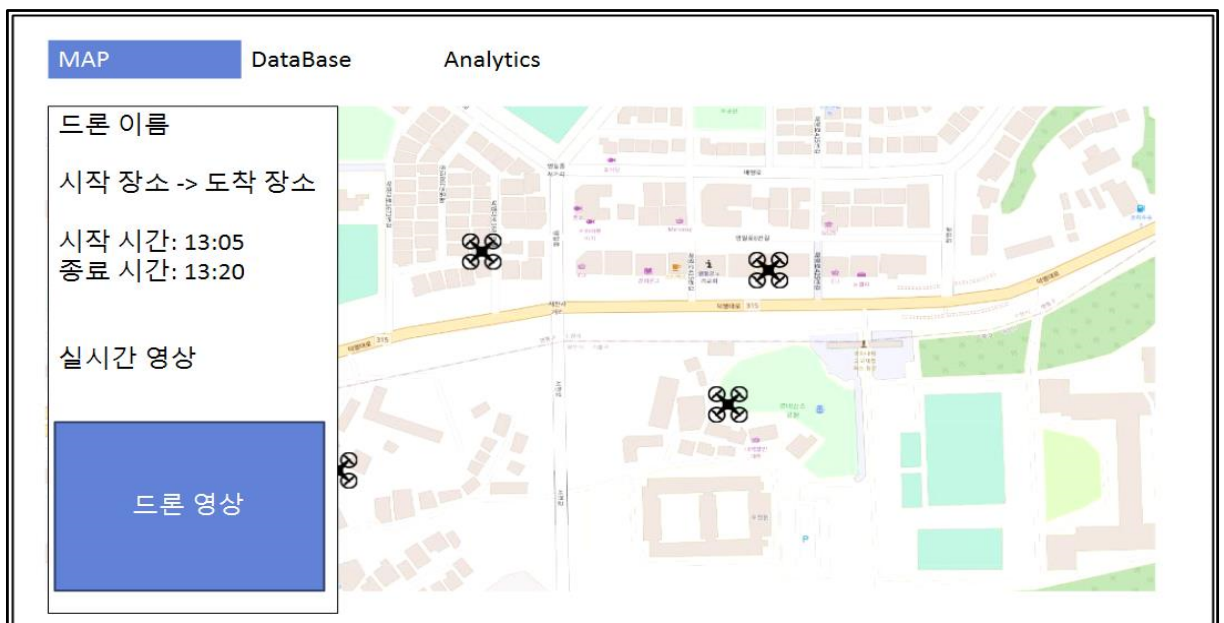
### 3. 프로젝트 내용

#### 3.1. 시나리오

##### 3.1.1. Main Map UI



[그림 1] 메인 화면



[그림 2]드론 아이콘 클릭시

[그림 1]는 웹 어플리케이션의 메인 화면이다. 메인화면에는 지도화면과 현재 비행하고 있는 드론의 움직임을 지도위에 보여준다. 드론 아이콘을 클릭하면 [그림 2]화면이 출력된다. [그림 2]는 클릭한 드론에 대한 정보를 왼쪽 레이아웃에 표시해주도록 한다.

### 3.1.2. DataBase UI

MAP

DataBase

Analytics

DataBase - Drone Table

드론이름

드론종류

드론제조사

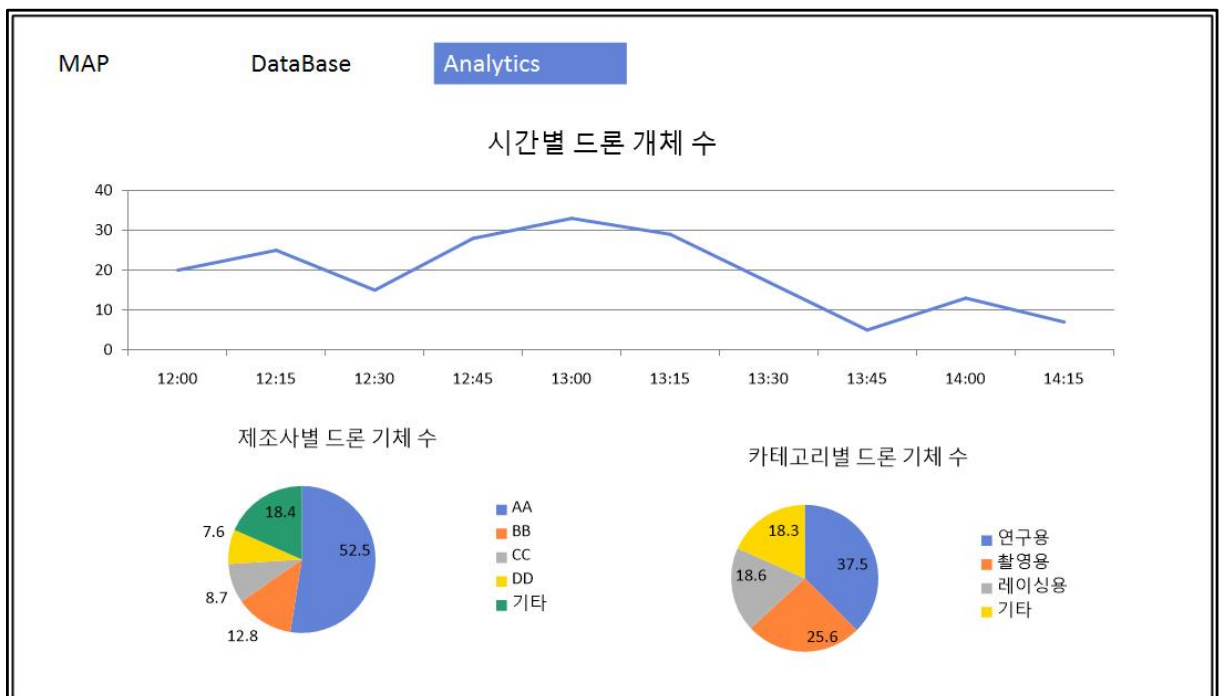
시리얼번호

검색된 드론 리스트

NO	드론 이름	드론제조사	드론 종류	시리얼번호
1	<a href="#">TEST1</a>	AA	촬영용	1111-11111
2	<a href="#">TEST2</a>	BB	연구용	2222-22222
3	<a href="#">TEST3</a>	CC	레이싱용	3333-33333

DataBase는 드론에 대한 데이터를 보여주는 UI이다. 초기화면은 드론기체에 대한 이름만이 리스트의 형태로 구현이 되어있으며, 기체 이름 클릭시에는, 드론에 대한 상세정보를 화면에 출력해준다. 그리고 화면 상단에는 필터링 옵션이 들어가있고, 각기 입력한 정보에 따라 이에 맞는 드론의 리스트가 화면에 출력된다.

### 3.1.3. Analytics UI



Analytics는 드론의 통계정보를 담은 화면이다. 화면 중앙에는 꺾은선 그래프가 표시되도록 하며, 하단에는 원형 차트가 2개정도 배치되도록 한다. 꺾은선 그래프의 값에 접근이 이것이 무엇을 의미하는지에 대해 스티커 메시지 형태로 보여주도록 한다.

## 3.2. 요구사항

### 3.2.1. Main Map에 대한 요구사항

- 드론 아이콘은 현재 움직이는 방향에 따라서 각기 다르게 표시해주도록 한다. 이 때 방향은 데이터 벡터 계산을 이용하여 구한다.
- 드론 클릭 시에는 레이아웃 왼쪽에 구현하도록 하며 클릭한 아이콘에 대한 정보만을 보여주도록 한다. 표시될 정보로는 드론 기체의 간략한 정보와 비행 스케줄 정보이다.
- 드론의 소켓 로그가 30초이상 Response가 오지 않으면 해당 드론 지도에서 삭제하며 이동거리 또한 초기화 시켜주도록 한다.
- 검색 기능의 경우 드론의 모델명을 이용하되, 중복된 내용이 있을 경우 필터링을 허용하도록 한다. 이때 필요한 필터링으로는 드론의 무게, 제원, 종류 등이다.

### 3.2.2. DataBase에 대한 요구사항

- DataBase는 대량의 데이터를 관리하고 복잡한 쿼리문을 잘 처리하기 위한 PostgreSQL을 사용한다.
- 필요한 데이터 테이블로는 회원, 드론, 실제 운행, 비행 스케줄 4가지로 구성이 되어있다. 또한, 회원-드론기체의 매핑을 위한 테이블과 회원-비행스케줄의 매핑을 위한 테이블이 구성되어있다.
- 회원 테이블에는 아이디를 기본키 속성으로 가지며 이름, 연락처, 소속을 속성으로 가진다.
- 드론기체 테이블은 ID를 기본키 속성을 포함하여 모델명, 제조사, 종류, 모델사진, 제원, 무게 등을 속성으로 갖는다. 이때, 비어있는 정보가 많을 수 있으므로, 빈 정보는 Custom, Anonymous를 Default 값으로 갖도록 한다.
- 실제 운행 데이터에는 ID(PK) 드론기체, 비행스케줄 ID와 함께 드론위치좌표(위도 및 경도), 현재 속도와 고도를 속성을 갖는다. ID는 시퀀스로 생성하며, 드론기체 ID와 비행스케줄 ID는 외래키와 함께 NULL 값을 허용하도록 한다.
- 비행 스케줄 데이터로는 아이디(PK), 시작 시간과 종료 시간, 시작 위치 좌표(위도,경도), 종료 위치 좌표(위도, 경도)를 속성으로 갖는다.
- 회원과 드론 기체는 1대N관계를 가지며, 회원과 비행 스케줄 또한 1대N 관계를 가진다. 이 관계는 Software Maintenance를 위해 외래키로 넣지 않고 각각 별도의 매핑 테이블로 구현하도록 한다.

### 3.2.3. Analytics 그래프에 대한 요구사항

- Analytics의 그래프는 하이차트로 구현하도록 한다. 차후 웹에서 뿐만 아니라 모바일 애플리케이션에서도 호환이 가능하도록 하기 위함이다.
- 그래프로 보여 줄 자료로는 드론 개체 수, 드론 비행 시간 등이다. 꺾은선 그래프로는 개체수에 대한 통계를 표현해주도록 하며, 원형차트로는 카테고리별과 종류별로 개체 수의 비율을 표현해준다.

## 4. 향후 일정 및 역할 분담

### 4.1. 백엔드

- 필수기능에 한해서 메인과 서브로 나눠 서로 역할 분담.
  - 대용량 데이터를 실시간으로 효율적으로 통신 할 수있는 효율적인 구조 방안 - 공통
  - 데이터 베이스 구조 설계 및 구성 - 김대철(Main), 배상현(Sub)
  - 실시간 드론 데이터 전송 - 배상현(Main), 김대철(Sub)
  - 실시간 데이터 로그 데이터 처리 - 배상현(Main), 김대철(Sub)
  - 드론 상세 데이터 전송 및 저장 - 김대철(Main), 배상현(Sub)
  - 호출 API 구성 및 구현 - 김대철(Main), 배상현(Sub)
- 추가적인 기능
  - 분석 데이터 확보 및 저장
  - 날씨 관련 데이터 확보 및 저장
  - 드론비행 금지/허용 구역 데이터 확보 및 저장

### 4.2. 프론트엔드

1. 웹 페이지 UI/UX 설계 : 고다경
  - 사용자 요구사항에 맞는 UI 구성
  - 버튼, 컴포넌트의 효율적 배치 등의 UX 설계
  - UI/UX 설계를 바탕으로 페이지 제작
2. Socket 통신 핸들링 : 윤준현
  - 대용량 데이터 통신을 위한 소켓 통신 구조 최적화
  - 백엔드와의 소켓 통신 핸들링
  - 소켓 데이터 정제 ( 드론 좌표 계산, 벡터 계산 등)
3. Open Street Map 활용: 신승민
  - OSM API 활용 (웹 페이지에 지도 표시 및 현재 위치 감지)
  - 소켓 데이터를 OSM에 가시화

- 드론 상세 레이아웃 등 드론 모니터링 화면 UI 구성

## 5. 결론 및 기대효과

본 연구의 어플리케이션은 가상현실 속 드론 혹은 실제 드론의 실시간 위치와 상태(배터리 상태, IMU 및 GPS 신호, 목적지 정보 등)를 추적하여 사용자에게 웹으로 정보를 제공한다. 제공된 정보를 통하여 운용 중인 드론들을 파악함으로써 드론 관련 사고 및 오용을 예방할 수 있고, 사고 발생시 위치 파악 및 즉각 대응이 가능할 것이다.

또한 실시간으로 수집한 드론 데이터를 빅데이터화하여 분석 및 시각화를 통해 드론 산업 현황을 파악할 수 있을 것이다.

현재 드론 비행의 허가 및 관리에 대한 법제화가 되지 않은 상태이지만, 앞으로 드론 비행을 하기 위해 신고를 하고 실시간 정보를 제공해야 하게 될 것으로 보고 여러 방면으로 유용하게 활용될 것이라고 예측한다.

## 6. 참고문헌

- [1] FlightAware 홈페이지: <https://ko.flightaware.com/>
- [2] OpenStreetMap 공식 홈페이지 : <https://www.openstreetmap.org/>
- [3] Leaflet 공식 홈페이지 : <https://leafletjs.com/>
- [4] Nuxt 공식 홈페이지 : <https://ko.nuxtjs.org/>
- [5] Vue 공식 홈페이지 : <https://kr.vuejs.org/v2/guide/>
- [6] px4 공식 홈페이지 : <https://px4.io/>
- [7] PostgreSQL 공식 문서: <https://postgresql.kr/docs/>
- [8] 프론트엔드 개발자 만족도 설문조사 :  
<https://2020.stateofjs.com/en-US/technologies/front-end-frameworks/>
- [9] 캡틴 판교의 Vue.js 입문 :  
<https://joshua1988.github.io/web-development/vuejs/vuejs-tutorial-for-beginner/>
- [10] MicroSoft의 MVVM 패턴 :  
<https://docs.microsoft.com/ko-kr/archive/blogs/msgulfcommunity/understanding-the-basics-of-mvvm-design-pattern>
- [11] 구글의 웹 렌더링 :  
<https://developers.google.com/web/updates/2019/02/rendering-on-the-web>
- [12] 한눈에 살펴보는 PostgreSQL: <https://d2.naver.com/helloworld/227936>
- [13] 손만근, 박기성, 공용해 (2015). 웹 소켓을 이용한 병원 접수 시스템 개선에 관한 연구. 한국컴퓨터정보학회논문지 , 20(1), 185-195

- [14] 안광규, 이택근, 김강석, 홍만표 (2013). HTML5 WebSocket을 기반으로 한 실시간 예약 시스템 개선 방안 관련 연구. 한국정보과학회 학술발표논문집, 962-964
- [15] 서준오, 김철원 (2018). IoT환경에서 MQTT와 WebSocket을 활용한 실시간 사물제어 시스템 설계 및 구현. JKIECS, vol. 13, no. 03, 517-524, 2018