

LoRaWAN 네트워크를 활용한 교내 환경 모니터링 서비스



지도교수 : 김종덕

팀명 : 물구나무서서 코딩하기

201924551 이 현

202055522 김은지

202055590 장서윤

목차

1. 과제 배경 및 목표	2
1.1. 과제 배경	2
1.2. 모니터링 서비스	2
1.2.1 스마트 쓰레기통	2
1.2.2 흡연부스 모니터링	3
2. 시스템 구성도	4
2.1. 스마트 쓰레기통	4
2.2. 흡연부스 모니터링	5
3. 사용 기술 및 개발 환경	6
4. 개발 일정 및 역할 분담	11
4.1. 개발 일정	11
4.2. 역할 분담	12

1. 과제 배경 및 목표

1.1. 과제 배경

부산대학교 안에 존재하는 다양한 시설들로 인해 발생할 수 있는 불편함이 존재하며, 이를 해결하기 위한 모니터링 서비스가 필요하다. 주기적인 모니터링이 필요한 서비스로는 주차장 자리 찾기, 흡연부스 모니터링, 스마트 쓰레기통 등이 있으며, 이러한 서비스들은 개별 특성과 요구사항을 가지지만, 동시에 운영되어야 한다. 서비스들의 효율적인 관리를 위해 교내 환경 모니터링 서비스를 개발하고자 한다.

해당 서비스를 운영함으로써 얻는 이점은 다음과 같다.

1. 여러 서비스를 통합적으로 관리할 수 있어 운영 및 유지보수의 효율성을 높인다.
2. 각 서비스는 필요한 자원만큼 할당받아 실행되므로, 자원의 효율적인 분배가 가능하다.
3. 새로운 서비스를 추가하거나 삭제할 때 용이하다는 장점이 있다.

각 서비스는 도커 컨테이너 내에서 관리되며, 컨테이너들을 효율적으로 관리하기 위해 쿠버네티스를 활용한다.

1.2. 모니터링 서비스

필요한 교내 모니터링 서비스들은 다음과 같다.

1.2.1. 스마트 쓰레기통

교내 환경미화원은 각 건물의 층마다 쓰레기통을 수거해야 한다. 쓰레기통이 다 차지 않았음에도 각 층에 있는 쓰레기통을 비워야 하는데, 이는 불필요한 인력이 낭비된다. 이를 해결하고자 쓰레기통의 적재량을 알려주는 서비스를 개발한다. 적재량을 알려줌으로써, 쓰레기통이 다 찼을 때 비워줄 수 있다. 이는 환경미화원의 동선을 최소화하여 효율적인 쓰레기통 관리가 가능하다.

기존 사례로 서울 관악구에서 실시한 스마트 쓰레기통 서비스를 들 수 있으나, 이는 3년 만에 중단되었다. 본 사례가 실패한 원인은 다음과 같다.

1. 환경미화원의 정해진 동선

환경미화원이 정해진 시간표대로 이동하기 때문에, 쓰레기통이 찼다는 알람이 오더라도 위치상 바로 처리하기 힘들다. 그러나 건물의 층마다 놓인 쓰레기통을 관리함으로써, 층을 일일이 돌지 않는 것으로도 큰 이점이 될 것이다.

2. 비용 대비 효과가 미미하다.

서울시에서는 길마다 쓰레기통 자체를 새로 설치하는 설치비용으로 약 900만원이 들었으나, 우리는 기존에 존재하는 쓰레기통에 센서를 부착하는 과정만 필요하므로 비용이 절약된다.

해당 서비스는 쓰레기통 뚜껑에 초음파센서를 부착하여 적재량을 확인하고, 센서 데이터는 네트워크를 통해 서버로 보내진다. 최종적으로 애플리케이션 상에서 각 쓰레기통의 적재량을 보여준다.

1.2.2. 흡연부스 모니터링

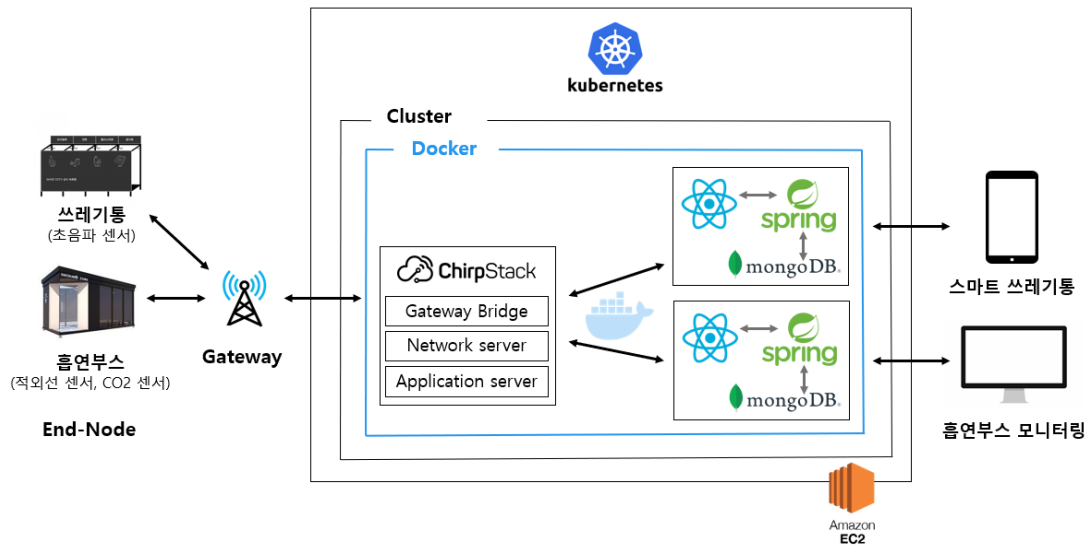
흡연부스는 흡연자의 쾌적한 흡연 공간 제공과 비흡연자의 간접흡연으로 인한 피해 방지를 위해 사용된다. 그러나 흡연부스 내, 공기질 문제로 인해 일부 흡연자는 부스 내부에서 흡연을 기피하고 부스 외부에서 흡연하게 되며 비흡연자에게 피해가 발생하고 있다. 이러한 문제를 해결하기 위해, 흡연부스 모니터링 서비스를 개발하고자 한다.

이 서비스는 먼저 교내 흡연부스의 위치와 상태를 지도로 시각화하여 흡연자의 편의를 제공한다.

적외선 센서를 사용하여 흡연부스 사용 인원을 측정하여 알려줌으로써 흡연부스별 이용률에 따른 사용 여부를 결정할 수 있도록 한다.

또한 공기질 센서를 사용해 공기질을 모니터링하여 적정 공기질이 넘었는지 여부를 조회할 수 있도록 한다.

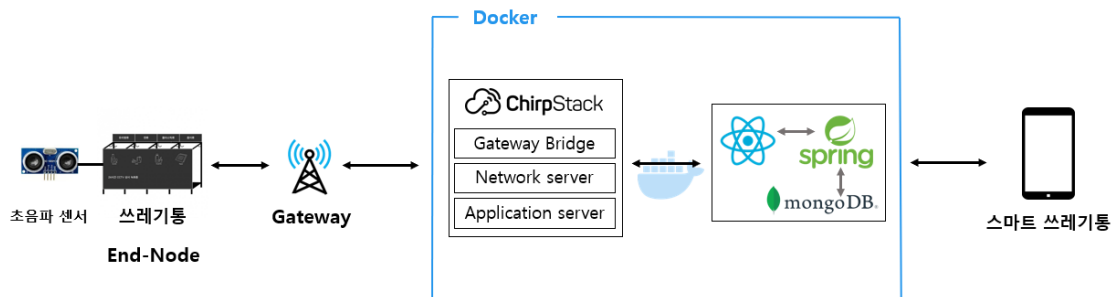
2. 시스템 구성도



[그림 1] 시스템 구성도

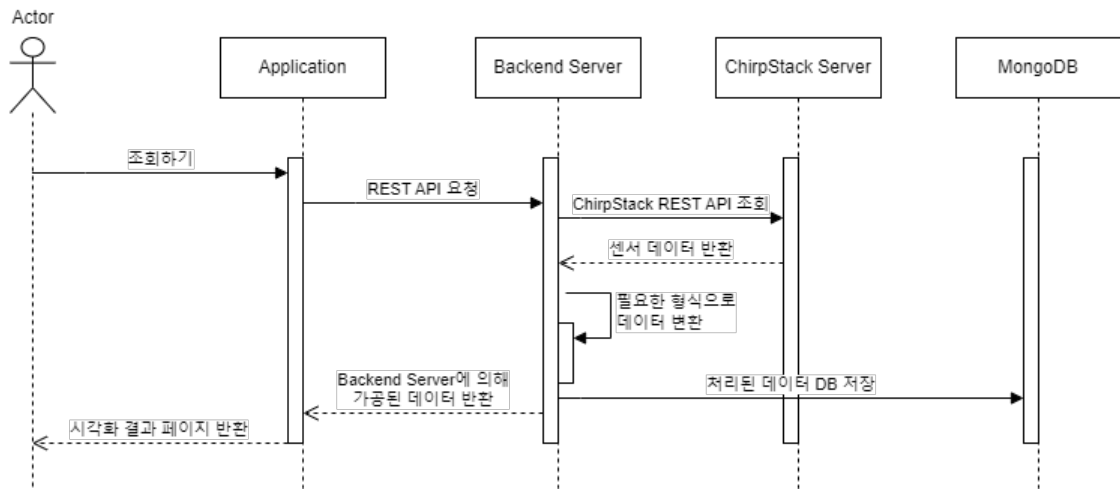
시스템 구성도는 [그림1]과 같다. 서비스와 ChirpStack은 도커 컨테이너로 구성되어 있으며, 컨테이너들은 쿠버네티스에 의해 효율적으로 관리된다. 운영 및 유지보수의 효율성을 높이고, 추후에 새로운 서비스를 빠르게 추가할 수 있다.

2.1. 스마트 쓰레기통



[그림 2] 스마트 쓰레기통 시스템 구성도

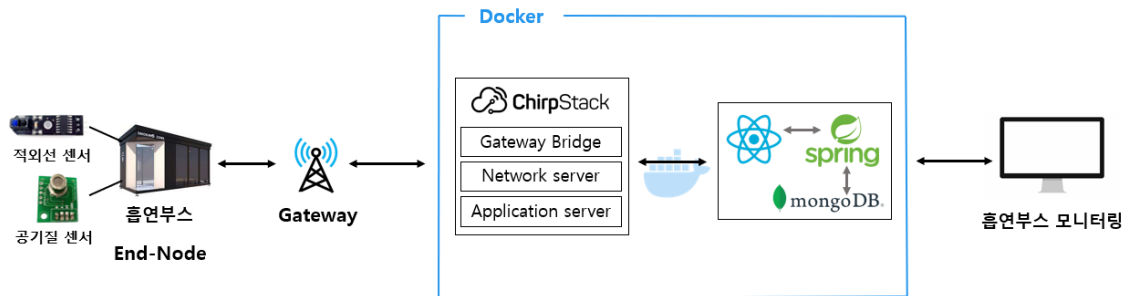
먼저 스마트 쓰레기통 서비스는 쓰레기통 뚜껑에 부착된 초음파 센서를 통해, 쓰레기통의 적재량을 확인하며 해당 정보를 Gateway로 보낸다. 다음으로 Gateway와 Network Server 간의 인터페이스 역할을 해주는 Gateway Bridge가 정보를 송수신하여 Network Server 혹은 Gateway에서 이해할 수 있는 형식으로 변환하여 전송한다. 마지막으로 Application Server에 데이터를 전송하여 최종적으로 사용자가 애플리케이션 형태로 결과를 확인할 수 있다.



[그림 3] 스마트 쓰레기통 시퀀스 다이어그램

스마트 쓰레기통 서비스의 전체적인 동작 흐름은 [그림 3]와 같다. 사용자가 애플리케이션에서 쓰레기통 적재량 조회를 요청하면 백엔드 서버에 구현된 API가 호출되며, 마지막으로 ChirpStack REST API를 호출하여 데이터를 조회한다. 조회 결과를 DB에 저장한 후 관련 데이터들을 시각화하여 사용자에게 보여주게 된다.

2.2. 흡연부스 모니터링



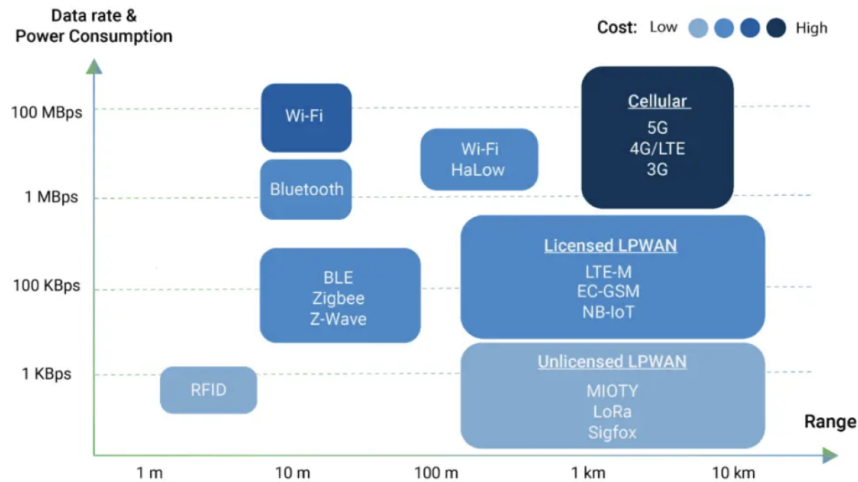
[그림 4] 흡연부스 모니터링 시스템 구성도

적외선 센서를 활용하여 흡연부스 이용 인원을 파악하며, 공기질 센서를 통해 공기질을 파악한다.

2.1 스마트 쓰레기통 서비스와 동일한 방법으로 Gateway가 센서의 정보를 받아오며, ChirpStack을 거쳐 마지막으로 사용자가 애플리케이션 형태로 결과를 확인할 수 있다. 시퀀스 다이어그램은 [그림 3]의 스마트 쓰레기통 서비스의 동작 과정과 동일하다.

3. 사용 기술 및 개발 환경

3.1 LoRaWAN



[그림 5] 무선 네트워크 종류

교내 환경 모니터링 서비스 내에서 흡연부스 서비스와 쓰레기통 서비스를 위한 무선 네트워크 통신으로 LoRaWAN을 사용한다.

무선 네트워크 통신 기술 중 LoRaWAN을 채택한 이유는 다음과 같다.

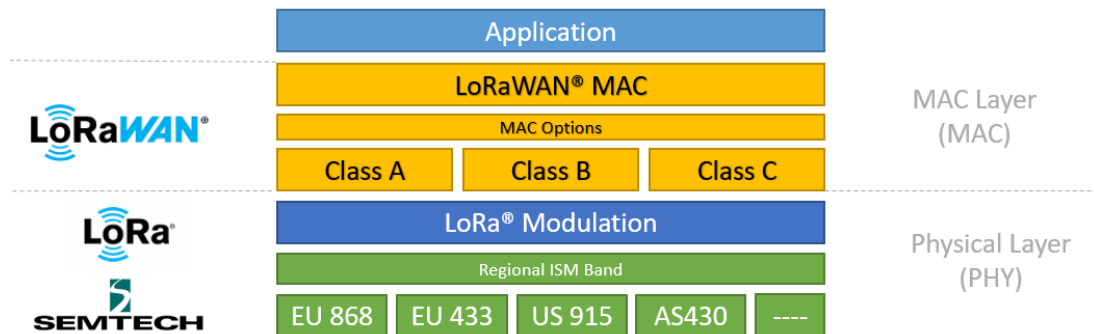
1. 모니터링 서비스이므로 오랜 시간 동안 주기적으로 데이터를 받아와야 하고, 많은 End-Node가 필요하다. 이때 높은 전력을 사용하는 와이파이, 블루투스는 적합하지 않다.
2. 흡연부스 모니터링 서비스의 경우, 멀리 떨어진 흡연부스들을 효과적으로 모니터링하기 위해 장거리 통신을 지원해야 한다.
3. 스마트 쓰레기통의 경우, 실내에서도 통신이 가능해야 한다.

LoRaWAN은 이와 같은 사항을 충족시키는 저전력, 장거리를 지원해 주고 주파수가 낮아 반사와 굴절에 강하여 실내 통신이 가능한 특성을 가진다.

저전력, 장거리를 지원하는 또 다른 기술로는 NB-IoT, Sigfox가 있는데, 그중에서 LoRaWAN을 사용하는 이유는 다음과 같다.

1. IoT 특성상 배터리 수명이 중요한데, NB-IoT는 LoRaWAN에 비해 다소 높은 전력을 소모한다.
2. NB-IoT, Sigfox 둘 다 네트워크가 제공하는 커버리지가 정해져 있다. 특히 NB-IoT는 해당 네트워크를 제공하는 통신사 커버리지에 의존하는 등, 커버리지를 변경할 수 없다.

즉, 적절한 위치에 Gateway를 배치하여 커버리지를 확보할 수 있으며, 부산대 한정으로 필요한 범위에 맞게 커버리지를 맞출 수 있다.



[그림 6] LoRa 서비스가 사용하고 있는 계층

3.2. ChirpStack

ChirpStack은 LoRaWAN 네트워크 구축을 위해 다양한 기능을 제공하는 오픈 소스이다. LoRaWAN 네트워크는 End-Node, Gateway, Network server, Application server로 이루어져 있으며, Gateway와 End-Node를 중앙집중식으로 관리한다. 또한 LoRaWAN 네트워크 구축을 위한 인터페이스를 제공한다.

ChirpStack 구성요소는 다음과 같다.

먼저 End-Node의 경우, 실제 데이터를 수집하는 센서가 존재하며, 수집된 데이터는 Gateway로 보낸다. Gateway로 들어온 데이터는 Gateway Bridge를 통해 Network server로 보내진다.

Gateway Bridge의 경우, Gateway로부터 받은 데이터를 ChirpStack에서 사용하는 데이터 형식으로 변환한다. 즉, ChirpStack에서 사용할 수 있도록 공통 데이터 형식으로 바꾸는 작업을 Gateway Bridge가 담당한다.

Network server는 LoRaWAN 네트워크 서버를 구현하며, Application server는 외부 서비스를 연결한다.

ChirpStack은 커스터마이징 옵션을 제공하여 우리가 제공하는 서비스에 맞게 조정할 수 있으며, 데이터와 네트워크를 보호하는 기능과 프로토콜을 제공하므로 보안과 안정성을 강조하는 학교 내부 서비스로 사용하기에 적합하다.

3.3. Kubernetes

쿠버네티스는 환경 모니터링 서비스 내에서 여러 서비스를 관리할 수 있는 컨테이너 오케스트레이션 플랫폼이다. 쿠버네티스를 활용하는 이유는 다음과 같다.

환경 모니터링 서비스 내에 여러 서비스를 구동시킬 때 단일 물리적 서버에서 여러 애플리케이션을 실행하면 리소스 할당 문제가 발생할 수 있다. 쿠버네티스는 환경 모니터링 서비스 내에서 여러 서비스를 단일 머신에서 실행함으로써 리소스 제약을 줄이고 컴퓨팅 성능을 최적으로 활용할 수 있도록 함으로써 이 문제를 해결할 수 있다. 환경 모니터링 시스템의 각 서비스에는 서로 다른 도커 컨테이너를 사용한다.

쿠버네티스는 여러 머신에서 컨테이너화된 서비스를 관리하며, 다양한 워크로드를 처리할 수 있는 확장성을 제공하며 컨테이너와 노드 수를 동적으로 확장할 수 있다. 이를 통해 환경 모니터링 시스템에서 서로 다른 서비스를 원활하게 관리할 수 있으며, 빠르게 서비스 추가가 가능하다.

3.4. Docker

도커는 환경 모니터링 서비스의 개발 및 배포에 여러 가지 이점을 제공하는 컨테이너화 기술로 사용된다.

도커는 애플리케이션과 해당 종속성을 컨테이너로 패키징하여 서로 다른 환경에서 일관된 실행을 보장한다. 도커 이미지를 사용하여 애플리케이션을 배포하면 여러 환경에서 동일한 구성과 종속성을 유지할 수 있으므로, 문제를 쉽게 식별하고 호환성 문제를 최소화할 수 있다.

도커는 애플리케이션을 컨테이너로 패키징하여 배포 프로세스를 간소화한다. 서비스를 배포하기 위해 도커 호스트에서 인프라를 구성하고 필요한 도커 이미지를 컨테이너로 실행할 수 있다. 또한 수요에 따라 컨테이너 수를 조정하여 서비스를 쉽게 확장할 수 있어서 모니터링 서비스의 디바이스 수 변화에 따라 서버 인프라를 효율적으로 조정할 수 있다.

도커 컨테이너는 격리된 환경에서 실행되므로 애플리케이션이 서로 간섭하지 않고 독립적으로 운영된다. 모니터링 서비스의 경우, LoRaWAN 네트워크 서버와 관련된 특정 기능을 별도의 도커 컨테이너에서 실행할 수 있다. 이를 통해 서비스의 안정성과 보안을 강화하는 동시에 다른 컨테이너나 호스트 시스템에 미치는 영향을 최소화할 수 있다.

도커 컨테이너의 중요한 이점 중 하나는 전체 서비스를 중단하지 않고 개별 컨테이너를 업데이트하거나 수정할 수 있다는 점이다. 이 기능은 다운타임을 줄이고 사용자의 불편을 최소화한다. 또한 도커를 사용하면 이전 버전으로 빠르고 간편하게 롤백할 수 있어 문제 발생 시 신속하게 복구할 수 있다.

모니터링 서비스에서 사용자 또는 디바이스 수가 증가할 때 도커는 수평적 확장을 용이하게 한다. 컨테이너를 여러 호스트에서 실행할 수 있으며 로드 밸런싱을 통해 트래픽을 분산할 수 있다. 이를 통해 서비스 성능이 향상되고 많은 사용자를 효과적으로 처리할 수 있다.

도커는 개발 환경을 컨테이너로 패키징화하여 개발자 간 일관성을 쉽게 유지할 수 있다. 프로젝트 소스 코드와 함께 도커 파일을 공유하면 각 개발자가 동일한 개발 환경에서 작업할 수 있다. 따라서 버그가 최소화되고 원활한 협업이 가능하다.

3.5. React

React는 사용자 인터페이스를 구축하기 위해 사용한다. 가상 DOM을 사용하여 사용자 인터페이스를 효율적으로 업데이트하므로 반응성이 뛰어난 UI를 구현할 수 있어 부드럽고 즐거운 사용자 경험을 제공한다. 모니터링 서비스에서는 데스크탑과 모바일 등 다양한 기기에서 사용할 수 있어야 하므로 반응형이 뛰어난 React는 이 조건을 충족시킬 수 있다.

React는 컴포넌트 기반 아키텍처를 따르기 때문에 개발자는 UI를 독립적이고 재사용할 수 있는 컴포넌트로 만들 수 있다. 쓰레기통 모니터링 서비스에서는 각각의 쓰레기통 UI에 대한 반복적 컴포넌트를 사용할 수 있고, 흡연부스 모니터링 서비스에서도 각 흡연부스에 대한 UI를 나타내는 데 공통적인 부분을 컴포넌트로 편리하게 사용할 수 있다.

익숙한 기술을 활용함으로써 기존의 지식과 기술을 활용하여 환경 모니터링 서비스를 효과적으로 구축할 수 있으므로 개발 시간을 절약할 수 있다.

3.6. Java(Spring)

백엔드 서버는 Spring Framework를 사용한다.

Spring Framework는 Java로 개발된 Application Framework이며, 대부분 웹 개발에서 사용되고 있다. 또한 자바 진영에서 사용되는 ORM(Object Relational Mapping) 표준

기술인 JPA(Java Persistence API)를 사용하여 자바 코드를 DB와 매핑하여 사용할 수 있다.

정부 및 공공기관의 웹사이트를 개발할 때 사용하는 전자정부 표준프레임워크는 Spring Framework를 기반으로 만들었으며, 많은 대기업에서 Spring Framework를 사용하고 있기 때문에 그만큼 검증된 Framework임을 알 수 있다.

3.7. MongoDB

데이터베이스 기술로는 MongoDB를 사용한다.

MongoDB는 기존 관계형 데이터베이스에 비해 더욱 유연한 스키마를 지원한다.

쓰레기통 상태, 흡연부스 정보 등 데이터의 필드가 다양할 수 있는 환경 모니터링 서비스의 경우, 관계형 스키마로 이러한 데이터를 관리하기가 어려울 수 있으며 다른 데이터에 영향을 미치는 스키마 수정이 필요할 수 있다. MongoDB의 문서 기반 접근 방식은 개별 필드를 추가할 수 있고 각 문서에 서로 다른 필드를 수용할 수 있어 이러한 유연한 데이터 요구 사항을 처리할 수 있다.

MongoDB는 저장된 데이터에 대한 뛰어난 가시성을 제공한다. 이는 각 쓰레기통의 상태와 같은 간단한 데이터만 저장하면 되는 쓰레기통 모니터링 서비스에 특히 유리하다. MongoDB의 문서 구조는 이러한 유형의 데이터를 직관적으로 표현하고 쉽게 검색할 수 있게 한다.

MongoDB는 수평적 확장성을 지원하여 대량의 데이터를 효율적으로 처리하고 성능을 개선할 수 있다. 모니터링 서비스에서 데이터양과 디바이스 수가 증가함에 따라 데이터 저장 및 처리를 분산하기 위해 MongoDB를 클러스터로 구성하여 확장성을 확보할 수 있다. 이에 대용량 데이터를 효율적으로 처리하여 환경 모니터링 서비스에 적합한 데이터 수집 및 분석이 가능하다.

4. 개발 일정 및 역할 분담

4.1. 개발 일정

5월		6월				7월				8월					9월			
3주	4주	1주	2주	3주	4주	1주	2주	3주	4주	1주	2주	3주	4주	5주	1주	2주	3주	4주
서비스 UML 작성, 프로토타입 개발																		
쿠버네티스 스터디																		
		컨테이너 기반 애플리케이션구현																
						쿠버네티스 적용 및 환경 완성												
								중간 보고서 준비										
										LoRa 적용하여 센서 데이터 받아오기								
															안정화 (테스트 및 디버깅)			
																	최종 보고서 준비	

4.2. 역할 분담

이름	역할
이현	1. 스마트 쓰레기통 서비스 (백엔드) 2. 흡연부스 모니터링 서비스 (백엔드) 3. 쿠버네티스 구축
김은지	1. 흡연부스 모니터링 서비스 (프론트엔드) 2. LoRaWAN 네트워크 구축 3. 쿠버네티스 구축
장서윤	1. 스마트 쓰레기통 서비스 (프론트엔드) 2. LoRaWAN 네트워크 구축 3. 쿠버네티스 구축