

# 포트폴리오

## 팀 프로젝트 -- SmaTory

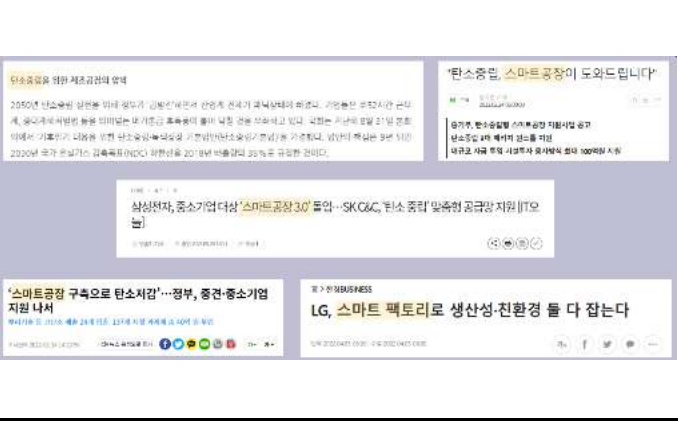
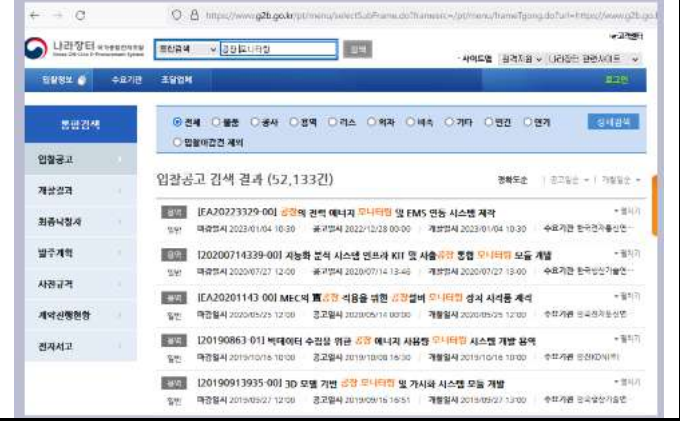
프로젝트 기간	2023.06.15~2023.08.08
개발 인원	4명
요약	스마트 팩토리 운영을 위한 모니터링 및 알람 시스템 SaaS
담당 기능	1) DB 실시간 동기화 2) 데이터 시각화 및 인사이트 제공 3) DB 에러 로그 전송 4) 탄소 배출량 알림
링크	<a href="https://kyounggu.tistory.com/92">https://kyounggu.tistory.com/92</a>

## 1. 프로젝트 개요

### 1.1 개발 기획

산업시대에서 공장은 빠질 수 없는 중요한 요소이다. 공장들은 생산성과 효율성을 높이기 위해 노력을 기울이고 있다. 또한, 4차 산업혁명에 대비하여 클라우드 연결을 통해 언제 어디서든 공장의 데이터에 접근하고 사용하길 원하고 있다. 마지막으로, 환경 문제를 해결하기 위해 친환경적 운영에도 큰 관심을 가지고 있다. 이러한 점들을 모두 해결할 수 있는 것이 '스마트 팩토리'이다.

실제로, 정부에서 탄소 중립을 위해 탄소 배출량 규제를 강화하고 있으며, 기업에서도 스마트 팩토리에 대한 지원을 하고 있으며, 직접 스마트 팩토리를 구축하고 있다. 그리고, 여러 기업에서도 공장 모니터링 서비스에 대한 수요가 있음을 확인할 수 있다. 이를 토대로, 스마트 팩토리 운영을 위한 모니터링 및 알람 시스템 SaaS를 기획하게 됐다.

	
정부의 탄소 중립과 기업의 스마트 팩토리 지원	나라 장터에서 확인한 '공장 모니터링' 수요

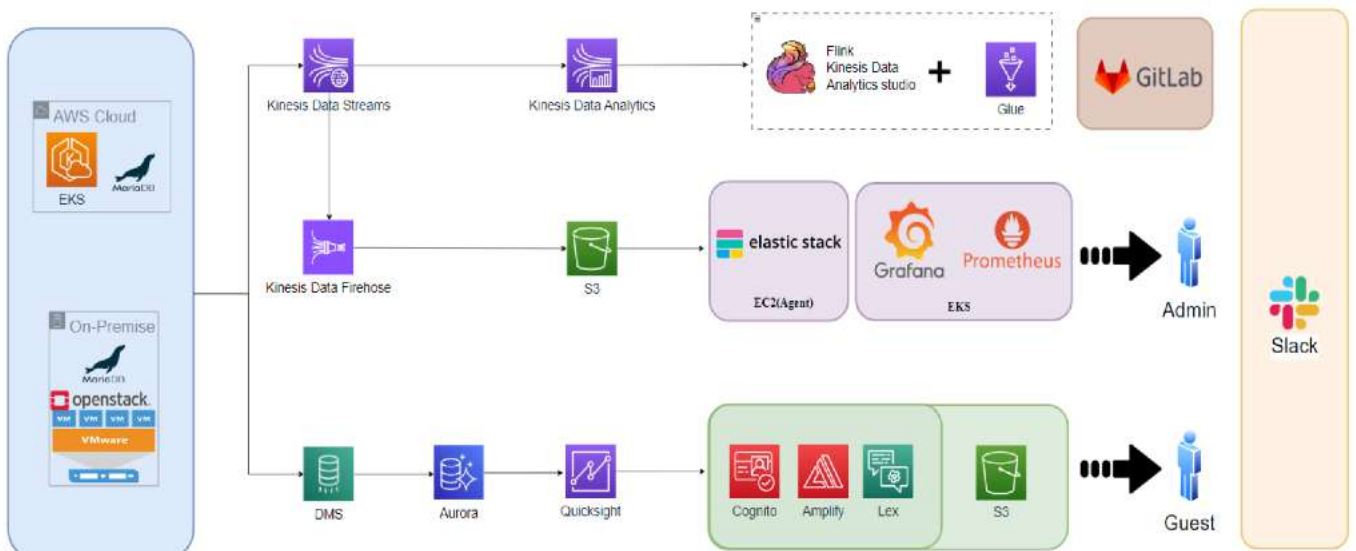
## 1.2 WBS :

단계	세부항목	JUN											JULY											AUG																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
		6월					6월					6월					7월					7월					7월					8월					8월																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																														
	프로젝트 주:	10	16	19	20	21	22	23	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
1	프로젝트 요구사항 분석	요구조사																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																</	

## 1.3 요구사항 :

- 공장)
  - 모든 공장은 자동차 생산을 목적으로 한다.
  - 기존의 데이터베이스를 사용하는 공장과 새로운 데이터베이스 구축이 필요한 공장 모두 고려한다.
- B. Smatory 사용자)
  - 실시간 공장 탄소 배출량, 온도, 자재 수량, 에너지 사용량 등의 분석 자료를 볼 수 있어야 한다.
  - 탄소 배출량이 임계값이 넘을 시 Email과 Slack을 통해 알림을 받아야한다.
- C. Smatory 관리자)
  - 클라우드 환경 DB 구축 및 관리가 가능해야한다.
  - 서비스의 보안 강화, 규정 준수, 문제 진단 및 해결을 위한 서비스 모니터링이 가능해야 한다.
  - 고객 자체 테스트를 위한 개발 서버 제공을 받아야 한다.

## 1.4 전체 Architecture :

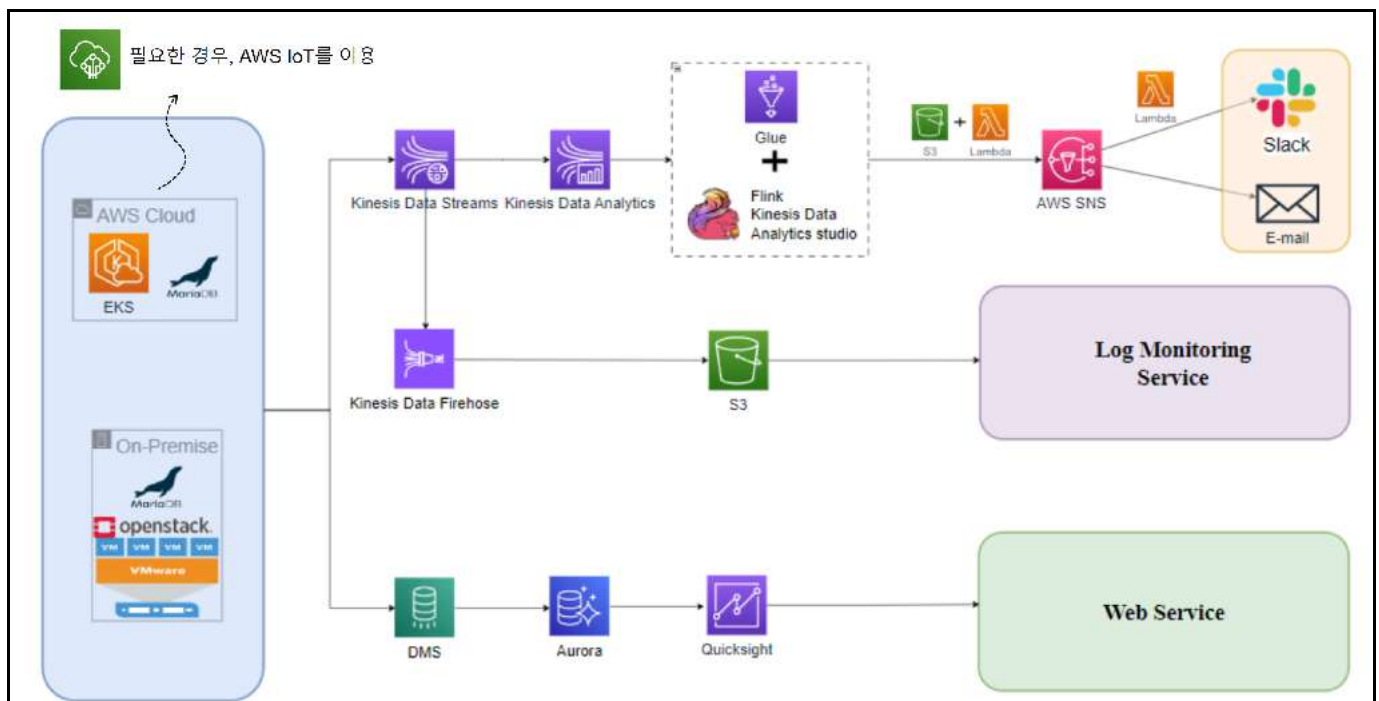


## 2. Data Infra (담당 파트)

### 2.1 요구사항 :



### 2.2 Architecture :



▲ 담당 서비스인 Data Infra Architecture

## Architecture



- Aurora DB

- DMS

[illegible]

## Qucisight로 구현한 '탄소 배출량' 대시보드



▲ SmaToryDB의 데이터를 그래프와 여러 분석 자료로 구성. 우측 하단에선 정보를 종합하여 앞으로의 탄소 배출량을 예측하는 인사이트 제공

## Qucisight로 구현한 '온습도' 대시보드



▲ SmaToryDB의 데이터를 그래프와 여러 분석 자료를 제공.

## Qucisight로 구현한 '자재 수량' 대시보드



▲ SmaToryDB의 데이터를 그래프와 워드 클라우드를 제공. 분석을 통해 보충이 필요한 자재를 알려줌.



## Qucisight로 구현한 '에너지 사용량' 대시보드



▲ SmaToryDB의 데이터를 그래프와 여러 분석 자료를 제공.

## 2.4 DB 에러 로그 전송 기능

### Architecture



### 왜 Kinesis Data Firehose인가?

#### ● 신속한 데이터 전달

데이터를 실시간으로 처리하여 지연시간을 최소화하여 스마트 팩토리의 상황을 빠르게 AWS Cloud에 전달할 수 있다.

#### ● 안정성과 내구성

데이터 전달의 무손실을 보장하고, 장애 자동 복구, 데이터 중복 제거와 오류처리를 지원해주므로 스마트 팩토리 DB의 정보를 안정적으로 전송할 수 있다.

#### ● 완전 관리형 서비스

AWS에서 완전히 관리하는 서비스이므로 설정, 확장, 운영에 대한 부담이 적다.

## - Kinesis Data Stream

데이터 스트림 (6) 정보							
		실시간으로 데이터 처리		Firehose 전송 스트림 생성		작업 ▼	
데이터 스트림 찾기							
<input type="checkbox"/>	이름	상태	출력 모드	프로비저닝된 샤드	데이터 보존 기간	암호화	광상원 라이선스 소유자
<input type="checkbox"/>	factory1-analytics-stream	🟢 활성	온디맨드	-	1일	비활성화됨	0
<input type="checkbox"/>	factory1-firehose-stream	🟢 활성	온디맨드	-	1일	비활성화됨	0
<input type="checkbox"/>	factory2-analytics-stream	🟢 활성	온디맨드	-	1일	비활성화됨	0
<input type="checkbox"/>	factory2-firehose-stream	🟢 활성	온디맨드	-	1일	비활성화됨	0
<input type="checkbox"/>	factory3-analytics-stream	🟢 활성	온디맨드	-	1일	비활성화됨	0
<input type="checkbox"/>	factory3-firehose-stream	🟢 활성	온디맨드	-	1일	비활성화됨	0

## - Kinesis Data Firehose

Factory1-ErrorLog [전송 스트림 상세](#)

전송 스트림 세부 정보

상태 🟢 <b>활성</b>	대상 Amazon S3	이벤트 변환 <b>활성화되지 않음</b>	분당 시간 2023년 7월 18일 15:55 GMT+9
소스 Amazon Kinesis Data Streams	ARN arn:aws:kinesis:ap-northeast-2:205338057397:codepipeline/factory1-error-log	출력 파이프라인 <b>활성화되지 않음</b>	오류 로그 상태 🟢 <b>오류 로그 프로그램</b>

▶ **테스트 데이터로 테스트** [정보](#)  
시뮬레이션된 데이터로 테스트를 실행할 수 있습니다. Amazon Kinesis Data Firehose 콘솔의 전송 스트림을 클릭하십시오.

[보내려 함](#) | [구성](#) | [데이터 오류 로그](#)

전송 스트림 메트릭 [보기](#)

1시간 | 5시간 | 12시간 | 1일 | 3일 | 1주 | 사용자 지정 [📅](#) | [🔄](#) | [📄](#) | [다시보기에 추가](#)

Kinesis Data Streams에서 읽...

ReadRate (Records/Second)

Kinesis Data Streams에서 읽...

ReadRate (Bytes)

DescribeStream 작업이 제한...

DescribeStream 작업이 제한된 횟수 (Number of DescribeStream operations limited)

## - S3

Amazon S3 > [보통](#) > [factory-kinesis-log](#) > 2023/ > 06/ > 30/ > 06/

06/ [🔍 S3 내의 문서](#)

[객체](#) | [속성](#)

객체 (6)  
이 객체는 Amazon S3에 저장되어 있는 기본 정보입니다. Amazon S3 [연락처](#)를 사용하여 S3에 있는 모든 객체의 목록을 얻을 수 있습니다. 다른 사용자에게 객체 이름을 공유할 수 있는 권한을 부여할 수 있습니다. [자세히 알아보기](#)

[🔍 S3 내의 문서](#) | [📄 XML 문서](#) | [📄 다운로드](#) | [📄 열기](#) | [📄 삭제](#) | [📄 작업](#) | [📄 공유 만들기](#) | [📄 업로드](#)

[🔍](#) [백분 표시](#)

<input type="checkbox"/>	이름	유형	마지막 수정	크기	스토리지 클래스
<input type="checkbox"/>	KDS-S3-A1Db4-1-2023-06-30-06-47-54-612059f3-2650-4521-9b5f-4a7be0ba5335	-	2023. 6. 30. pm 3:52:57 PM KST	752 KB	Standard
<input type="checkbox"/>	KDS-S3-A1Db4-1-2023-06-30-06-47-54-612059f3-2650-4521-9b5f-4a7be0ba5335	-	2023. 6. 30. pm 3:52:57 PM KST	1.8 KB	Standard
<input type="checkbox"/>	KDS-S3-A1Db4-1-2023-06-30-06-47-54-612059f3-2650-4521-9b5f-4a7be0ba5335	-	2023. 6. 30. pm 3:52:56 PM KST	1.5 KB	Standard

Kinesis Firehose로 전송된 로그가 S3에 날짜별로 정리된 폴더에 적재되는 모습

KDS-S3-A1Db4-1-2023-06-30-06-47-54-612059f3-2650-4521-9b5f-4a7be0ba5335 - Windows 메...

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```

230630 15:46:23 InnoDB: Shutdown completed; log sequence number 1605458
230630 15:46:35 InnoDB: Compressed tables use zlib 1.2.7
230630 15:46:35 InnoDB: Using Linux native AIO
230630 15:46:37 Percona XtraDB (http://www.percona.com) 5.5.61-MariaDB-38.13 started; log sequence number 1605458
230630 15:49:17 InnoDB: Starting shutdown...
230630 15:51:07 [Note] /usr/libexec/mysqld: Shutdown complete
230630 15:51:28 mysqld_safe Starting mysqld daemon with databases from /var/lib/mysql
230630 15:51:31 InnoDB: Mutexes and rw_locks use GCC atomic builtins
230630 15:51:31 InnoDB: Initializing buffer pool, size = 128.0M
230630 15:51:33 Percona XtraDB (http://www.percona.com) 5.5.61-MariaDB-38.13 started; log sequence number 1605458
  
```

적재된 로그 파일의 내용

## 2.5 탄소 배출량 알림 기능

### Architecture



### 왜 Kinesis Data Analytics인가?

#### ● 실시간 데이터 SQL 기반 분석

실시간으로 스트리밍 데이터를 처리하고 의사결정을 내릴 수 있어 스마트 팩토리의 데이터를 빠르게 처리할 수 있다. 또한 SQL을 지원하므로 전송된 스마트 팩토리의 데이터를 쉽게 분석할 수 있다.

#### ● 데이터 변환 및 필터링

스트리밍 데이터에서 필터링을 통해 필요한 정보만 추출하여 가공할 수 있다. 즉, 수집한 스마트 팩토리 데이터 중 필요한 정보만 필터링 하여 사용할 수 있다.

#### ● 확장성과 성능

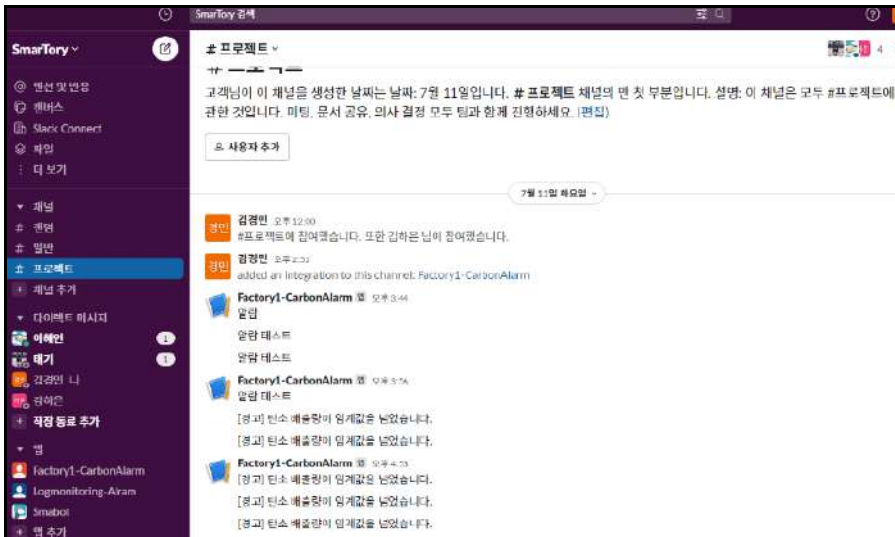
자동으로 확장되기 때문에 대용량 스트리밍 데이터 처리에도 문제가 없으며, 내구성과 안정성을 보장하여 스마트 팩토리의 데이터 손실 없이 분석할 수 있다.

## - Kinesis Data Analytics

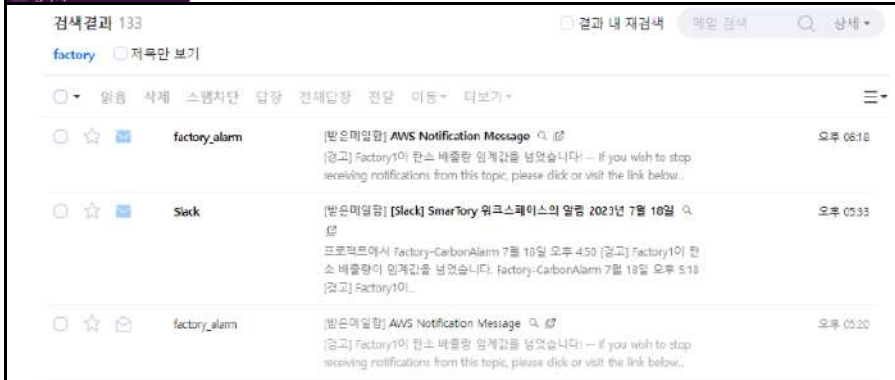
	<p>Factory1의 정보를 분석할 Kinesis Data Analytics.</p>
<pre> -- Flink SQL CREATE TABLE IF NOT EXISTS input_stream; CREATE TABLE IF NOT EXISTS output_streams (   days VARCHAR,   carbon_emission float ); INSERT INTO output_streams SELECT * FROM input_stream WHERE carbon_emission &gt;= 20.0; </pre>	<p>Flink Studio에서 SQL 기반으로 데이터를 실시간으로 분석.</p>



## - AWS SNS



Lambda를 통해 Slack에 알람 발생.



Email에 알람 발생.

### 맡은 역할

1. 공장 데이터베이스와 모니터링 데이터베이스를 구축.
2. 데이터를 시각화하고 인사이트를 제공하는 대시보드를 웹 서비스에 제공
3. 데이터를 전송하고 분석을 통해 경보를 발생하는 인프라 구축.

### 프로젝트를 진행하면서 어려웠던 점

1. DMS, QuickSight, Kinesis에 새로 도전하는데, 자료가 적어서 구현하는데 어려움을 겪음. 하지만, AWS 공식 문서와 WorkShop의 예제를 토대로 직접 여러 테스트를 하여 해결.
2. 각 팀원들이 각자 사용하고 싶은 기능들이 있었지만, 불필요한 경우들이

### 깨달은 점

1. 팀장 역할 수행하면서, 팀원 간의 역할 분배와 빠르고 세밀한 의사소통이 프로젝트 진행에 큰 영향을 끼치는 것을 느낌
2. AWS에는 많은 서비스들이 존재하며, 구성하고 싶은 대부분의 인프라를 서비스를 통해 구성할 수 있음을 알게 되어, AWS 서비스 사용 숙련도의 중요성을 느낌
3. AWS Aurora를 사용하면서 가용성, 보안, 성능 등의 많은 장점을 경험할 수 있었으며, DMS를 통해 CDC를 구현하며, 마이그레이션을 경험할 수 있었음.
4. Kinesis와 DMS의 경우, 정보가 적었기 때문에 AWS 공식 문서와 WorkShop의 예시를 참고하며, 많은 시행 착오 끝에 구현에 성공. 이를 통해, 새로운 영역이라도 끈기와 도전 정신을 가지면 해결할 수 있음을 느낌. 이렇게 얻은 성공이 다른 성공보다 가치 있음을 깨달았다. 설령 실패한다 하더라도 이 과정에서 배울 것들이 존재하며, 이런 습관이 또 다른 난관이 닥칠 때, 강하게 부딪힐 수 있는 용기를 주기 때문에 더욱더 중요함을 깨달았다.

## 팀 프로젝트 -- 강의실 커뮤니티 서버

프로젝트 기간	2023.03.09~2023.03.15
개발 인원	5명
요 약	On-premise 환경의 인프라 구축
담당 기능	1) DNS 2) DB (MariaDB)
링크	<a href="https://kyounggu.tistory.com/25">https://kyounggu.tistory.com/25</a>

# 1. 프로젝트 개요

## 1.1 개발 기획

현재 대다수의 기업에서 자사에 하나의 온-프레미스 인프라를 구성하기보다는 회사 외부에 오프-프레미스 인프라를 가지는 경우가 많다.

그렇기에 해당 프로젝트에서 협업을 통해 각자의 PC에서 온-프레미스 인프라를 구성하고 각자 담당한 기능을 구현한 뒤, 서로의 인프라를 마이그레이션하여 온-프레미스 인프라를 구현함과 동시에 전체적으로 오프-프레미스 인프라와 같은 구조를 구현할 수 있는 능력을 갖춰야한다.

## 1.2 목표 :

실습에서 배운 내용들을 토대로 각자의 온프레미스 인프라를 구성하고 담당한 기능을 구현하는 것이 1차적인 목표이며, 이후에 각 온프레미스 인프라를 연결하여 서로의 내부망에 접속할 수 있어야 한다.

실습에서 더 나아가, 각자가 담당한 서버에 추가적으로 활용할 수 있는 기능을 조사 및 구현하여 실습에서 구성한 인프라보다 더 발전한 인프라를 구축하는 것이 최종적인 목표다.

## 1.3 담당 요구사항 :

### ● DNS

1. DNS를 외부에서 사용할 수 있어야한다.

⇒ DNS의 Zone을 내부망(10.31.0.x)이 아닌 외부망(192.168.0.x)설정,

윈도우 실제 랜카드의 Ipv4의 DNS 서버를 DNS 컴퓨터의 NAT 주소로 설정,

NAT에 DNS 포트(53/TCP, 53/UDP)를 DNS서버로 포트 포워딩

### ● DB

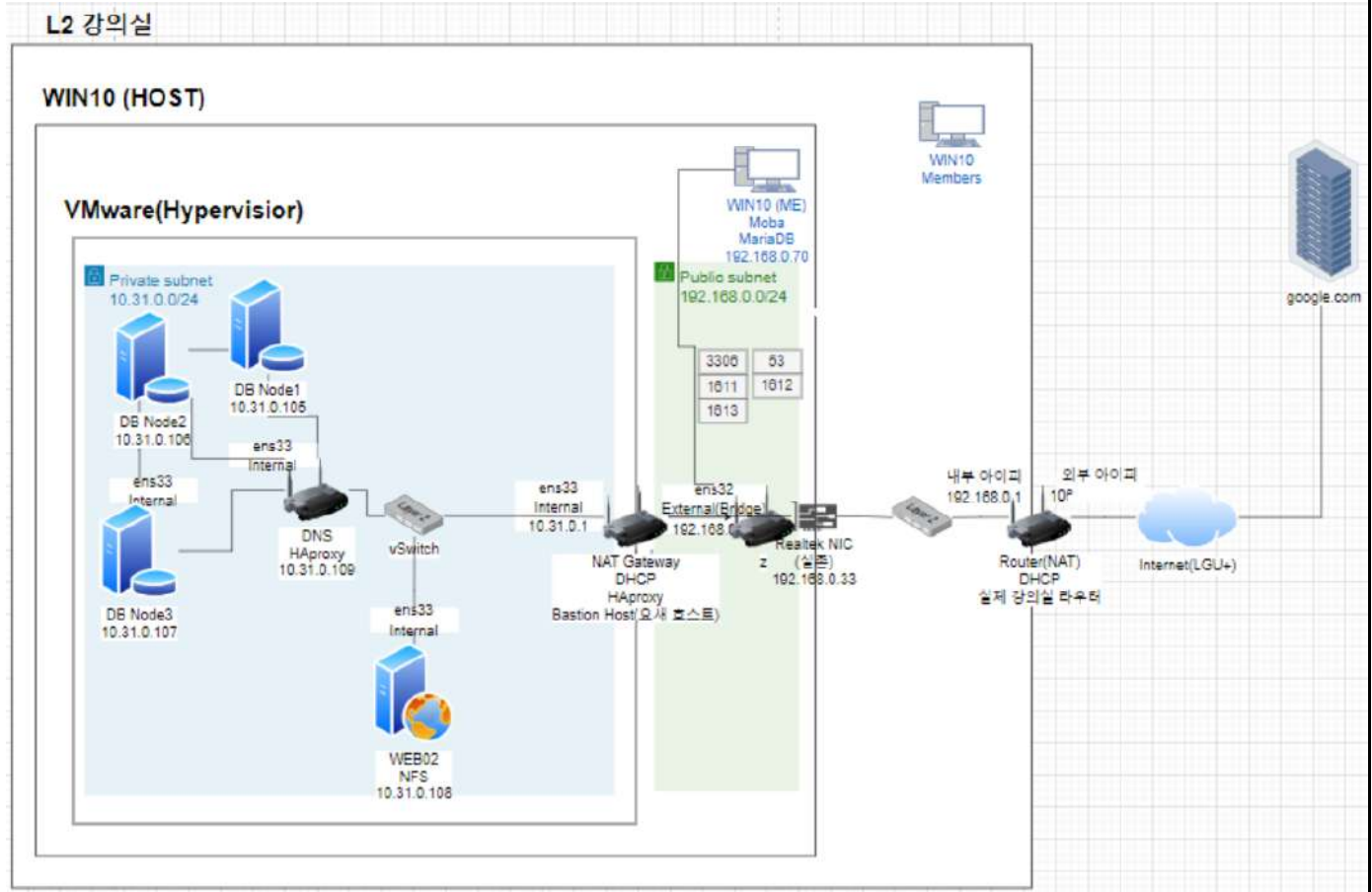
1. DB 서버는 외부에서도 접근하여 원하는 데이터를 전송/전달 받을 수 있어야 한다.

⇒ NAT를 통해 외부망과 내부망을 연결, NAT에 3306포트를 포워딩하여 DB접근 시, 별도의 과정 없이 DNS서버에 도착하여 HAProxy에 의해 세개의 DB 중 한 곳으로 접근

2. DB 서버는 다중화를 통해 하나의 DB서버가 다운되어도 다른 DB서버를 통해 서비스를 제공 해야한다.

⇒ 갈레라 클러스터(Galera Cluster)를 이용하여 DB를 다중화하여 서로 실시간으로 같은 DB서버를 사용. 셋 중 하나가 다운되어도 다른 두 서버가 DB를 운용. 로드밸런스의 효과 또한 볼 수 있다.

## 1.4 전체 Architecture :



## 2. 구축 과정

### 2.1 NAT :

#### NAT

```
ddns-update-style interactive;
subnet 10.31.0.0 netmask 255.255.255.0 {
    # = 10.31.0.0/24
    option routers 10.31.0.1;
    option subnet-mask 255.255.255.0;
    range dynamic-bootp 10.31.0.100 10.31.0.110;
    option domain-name-servers 10.31.0.109, 8.8.8.8, 8.8.4.4;
    default-lease-time 7200;
    max-lease-time 86400;
}
```

- DHCP를 이용하여 내부 네트워크(10.31.0.x)에서 100~110으로 아이피를 부여
- DNS 서버를 10.31.0.109로 설정

```
firewall-cmd --permanent --zone=external --add-forward-port=port=53:proto=tcp:toport=53:toaddr=10.31.0.109
firewall-cmd --permanent --zone=external --add-forward-port=port=53:proto=udp:toport=53:toaddr=10.31.0.109
firewall-cmd --reload
```

- DNS 포트를 (53/TCP, 53/UDP) DNS서버 (10.31.0.109)로 포트포워딩

```
199 firewall-cmd --permanent --zone=external --add-forward-port=port=1611:proto=udp:toport=161:toaddr=10.31.0.10
5
200 firewall-cmd --reload
201 firewall-cmd --permanent --zone=external --add-forward-port=port=1612:proto=udp:toport=161:toaddr=10.31.0.10
6
202 firewall-cmd --permanent --zone=external --add-forward-port=port=1613:proto=udp:toport=161:toaddr=10.31.0.10
7
203 firewall-cmd --reload
```

- cacti서버의 포트를 (1611, 1612, 1613/UDP) 각각 DB{1,2,3} 서버로 포트포워딩 (161포트로)

## 2.2 CentOS7 운영체제에 DB 설치 :

### DB {1,2,3} 서버 설치

```
curl -sS https://downloads.mariadb.com/MariaDB/mariadb_repo_setup | bash
vi /etc/yum.repos.d/mariadb.repo
yum makecache fast
yum install -y mariadb-server galera
```

- MariaDB yum repo를 설치하고, repo를 원하는 버전의 mariadb으로 수정하고 캐시를 빌드하고 설치 (갈레라도 함께)

```
firewall-cmd --permanent --add-service=mysql
firewall-cmd --permanent --add-port={4567,4568,4444}/tcp
firewall-cmd --reload
```

- Mysql 서비스와 갈레라 클러스터에 필요한 포트 ( 4444:SST / 4568:IST / 4567:Galera CLutser)를 방화벽 설정

```
vi /etc/my.cnf.d/server.cnf
```

```
[galera]
# Mandatory settings
wsrep_on=ON
wsrep_provider=/usr/lib64/galera-4/libgalera_smm.so
wsrep_cluster_address=gcomm://{10.31.0.105,10.31.0.106,10.31.0.107}
binlog_format=row
default_storage_engine=InnoDB
innodb_autoinc_lock_mode=2
```

- 갈레라 클러스터의 경로와 갈레라 클러스터에 쓰일 노드(DB)들의 주소를 입력

```
galera_new_cluster
```

```
systemctl start mariadb.service
```

- 최초(최신)노드에서 갈레라 클러스터를 생성하고나서 나머지 노드에서 mariadb를 시작해야한다.

## 2.3 DNS 서버:

### DNS 서버

```
yum install -y haproxy
```

- DNS서버의 HAProxy를 이용하여 3306포트는 DB {1,2,3}에 최소 접속 알고리즘으로 로드 밸런스

```
vi /etc/haproxy/haproxy.cfg
```

```
# Galera Cluster Frontend configuration
frontend galera_cluster_frontend
  bind *:3306
  mode tcp
  option tcplog
  default_backend galera_cluster_backend

# Galera Cluster Backend configuration
backend galera_cluster_backend
  mode tcp
  option tcpka
  balance leastconn
  server db1 10.31.0.105:3306 check weight 1
  server db2 10.31.0.106:3306 check weight 1
  server db3 10.31.0.107:3306 check weight 1
```

```
systemctl restart haproxy
```



```
firewall-cmd --permanent --add-port=3306/tcp
firewall-cmd --reload
```

- 3306/TCP 방화벽 설정

```
yum -y install bind bind-chroot bind-utils
```

- '/var/named/kyoung1.shop.zones'를 참조하도록 변경

```
vi /etc/named.conf
```

```
view "internal" {
    zone "." IN {
        type hint;
        file "named.ca";
    };

    include "/etc/named.rfc1912.zones";
    include "/var/named/kyoung1.shop.zones"; # 포스팅 업로드 생성
};
```

```
vi /var/named/kyoung1.shop.zones
```

```
zone "kyoung1.shop" IN {
    type master;
    file "kyoung1.shop.db";
    allow-update { none; };
};

zone "0.168.192.in-addr.arpa" IN {
    type master;
    file "0.168.192.in-addr.arpa.db";
    allow-update { none; };
};
```

- 순방향 존 (kyoung1.shop)이 파일 'kyoung1.shop.db'를 참조 하도록 설정
- 역방향 존 (0.168.192)이 파일 '0.168.192.in-addr.arpa.db'를 참조하도록 설정

```
vi /var/named/kyoung1.shop.db
```

```

@ITL      86400
@          IN      SOA      kyoung1.shop.  root.kyoung1.shop. (
                                2022041401 ; Serial
                                3h         ; Refresh
                                1h         ; Retry
                                1w         ; Expire
                                1h )      ; Minimum

                                IN      NS      ns.kyoung1.shop.
                                IN      MX      10 ns.kyoung1.shop.

ns         IN      A        192.168.0.109
samba     IN      A        192.168.0.189
dns        IN      A        192.168.0.183
nfs        IN      A        192.168.0.192
nat        IN      A        192.168.0.209
cacti     IN      A        192.168.0.203

```

```
vi /var/named/0.168.192.in-addr.arpa.db
```

```
$TTL      86400
@         IN      SOA      kyoung1.shop.  root.kyoung1.shop. (
                                2022041401 ; Serial
                                3h         ; Refresh
                                1h         ; Retry
                                1w         ; Expire
                                1h )       ; Minimum

          IN      NS       ns.kyoung1.shop.
183      IN      PTR      dns.kyoung1.shop.
189      IN      PTR      samba.kyoung1.shop.
192      IN      PTR      nfs.kyoung1.shop
209      IN      PTR      cacti.kyoung1.shop
203      IN      PTR      nat.kyoung1.shop
```

```
systemctl enable --now named
```

- 도메인 네임 samba, dns 의 아이피 주소 입력

- 도메인의 아이피 역주소 입력

인터넷 프로토콜 버전 4(TCP/IPV4) 속성

기본 구성

네트워크가 IP 자동 설정 기능을 지원하면 IP 설정이 자동으로 할당되도록 할 수 있습니다. 지원하지 않으면, 네트워크 관리자에게 적절한 IP 설정값을 문의해야 합니다.

☒ 자동으로 IP 주소 받기(O)

☐ 다음 IP 주소 사용(S):

IP 주소(I): . . .

서브넷 마스크(U): . . .

기본 게이트웨이(G): . . .

☐ 자동으로 DNS 서버 주소 받기(B)

☒ 다음 DNS 서버 주소 사용(E):

기본 설정 DNS 서버(P): 192. 168. 0. 183

보조 DNS 서버(A): 8. 8. 4. 4

☐ 공인 때 설정 유효성 검사(L)

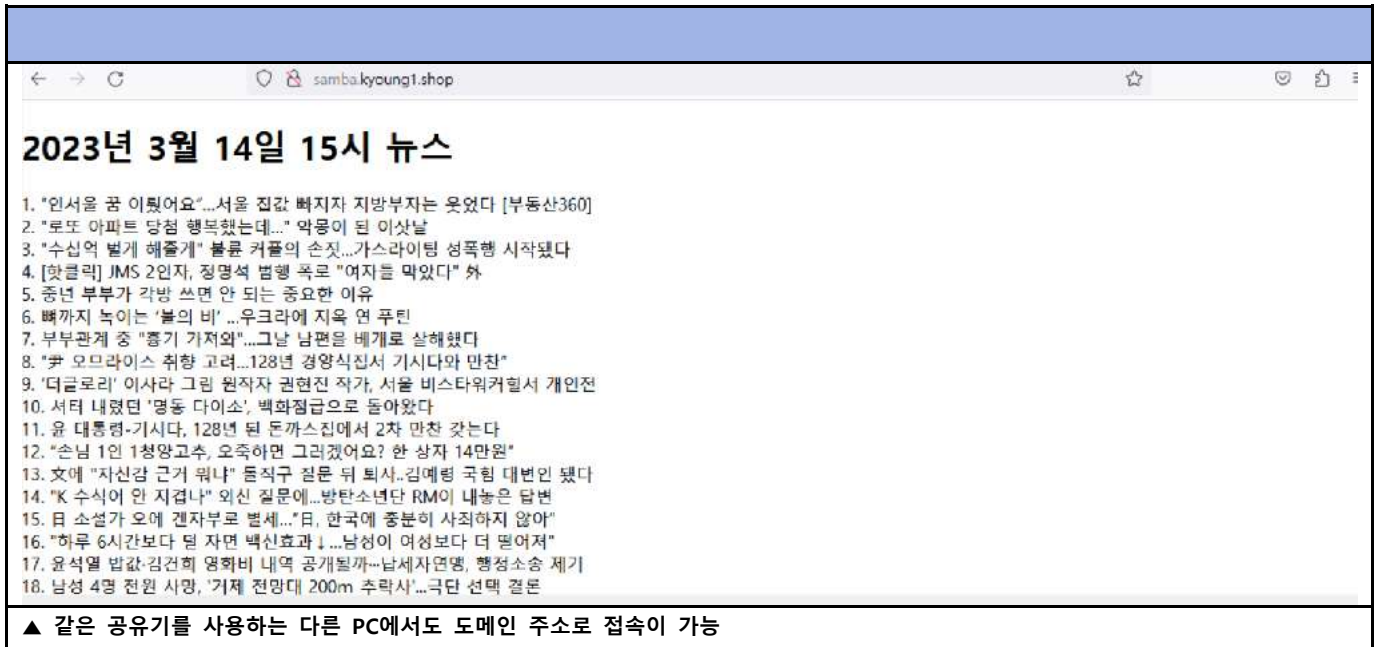
다음(N) >

- 윈도우(외부)가 DNS서버의 도메인을 이용하도록 설정



## 2. 구축 결과

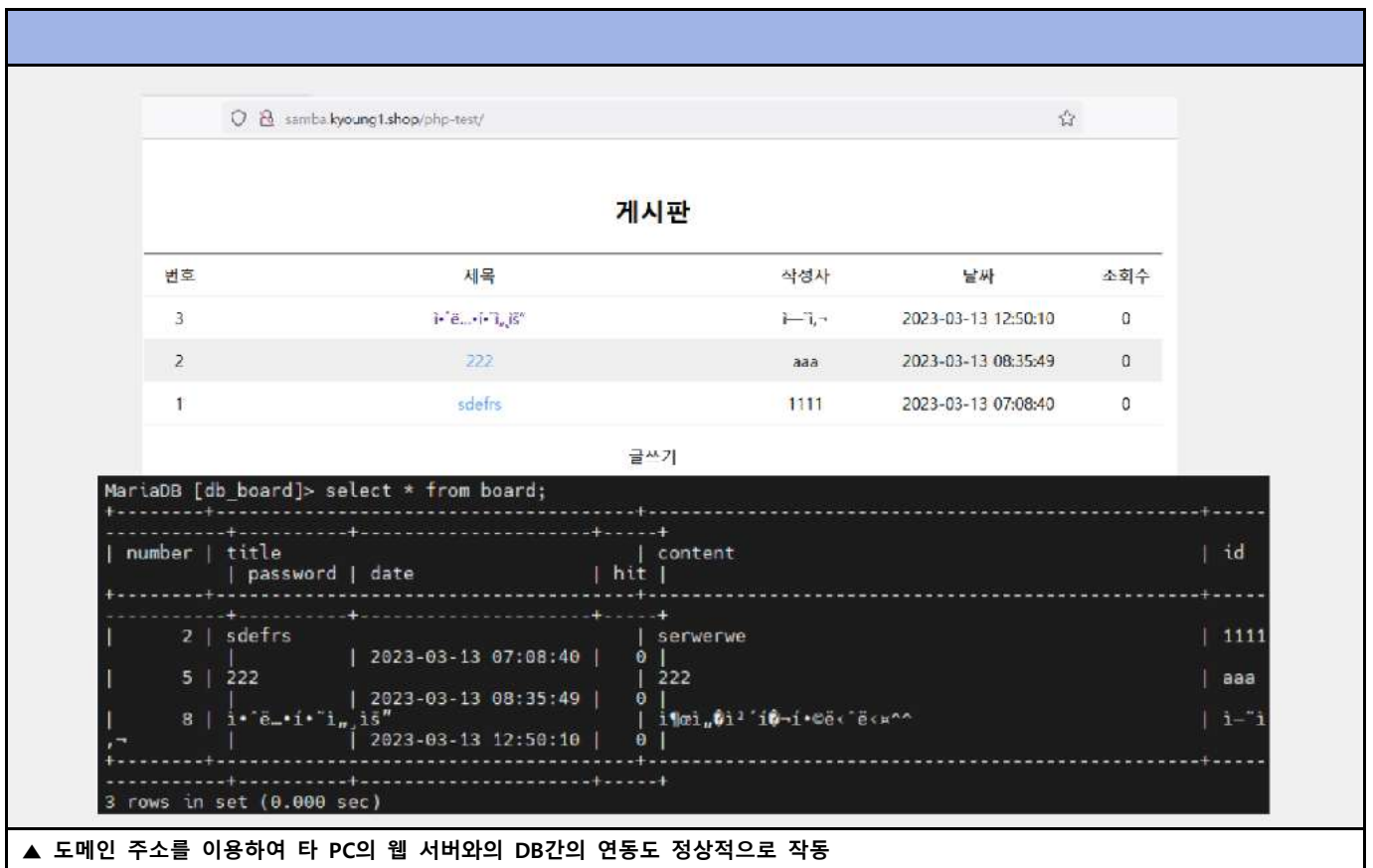
### - DNS 외부망 사용



2023년 3월 14일 15시 뉴스

1. "인서울 꿈 이뤄어요"...서울 집값 빠지자 지방부자는 웃었다 [부동산360]
2. "로또 아파트 당첨 행복했는데..." 악몽이 된 이삿날
3. "수십억 벌게 해줄게" 불륜 커플의 손짓...가스라이팅 성폭행 시작됐다
4. [핫클릭] JMS 2인자, 정명석 범행 폭로 "여자를 먹었다" 외
5. 중년 부부가 각방 쓰면 안 되는 중요한 이유
6. 뼈까지 녹이는 '불의 비'...우크라이나 지옥 연 푸틴
7. 부부관계 중 "흥기 가져와"...그날 남편을 베개로 살해했다
8. "췌 오므라이스 취향 고려...128년 경양식집서 기시다와 만찬"
9. '더글로리' 이사라 그림 원작자 권현진 작가, 서울 비스타워커힐서 개인전
10. 서덕 내렸던 '명동 다이소', 백화점급으로 돌아왔다
11. 윤 대통령-기시다, 128년 된 돈까스집에서 2차 만찬 갖는다
12. "손님 1인 1청양고추, 오죽하면 그러겠어요? 한 상자 14만원"
13. 문에 "자신감 근거 뭐냐" 물적구 질문 뒤 퇴사...김예령 국힘 대변인 됐다
14. "K 수식어 안 자갑나" 외신 질문에...방탄소년단 RM이 내놓은 답변
15. 日 소설가 오에 겐자부로 별세..."日, 한국에 충분히 사죄하지 않아"
16. "하루 6시간보다 덜 자면 백신효과↓...남성이 여성보다 더 멀어져"
17. 윤석열 발값-김건희 영하비 내역 공개되까...남세자연맹, 행정소송 제기
18. 남성 4명 전원 사망, '기재 전망대 200m 추락사'...국단 선택 결론

▲ 같은 공유기를 사용하는 다른 PC에서도 도메인 주소로 접속이 가능



게시판

번호	제목	작성자	날짜	조회수
3	i*ë...i*i,i5"	i-i,i,-	2023-03-13 12:50:10	0
2	222	aaa	2023-03-13 08:35:49	0
1	sdefrs	1111	2023-03-13 07:08:40	0

글쓰기

```
MariaDB [db_board]> select * from board;
+-----+-----+-----+-----+-----+
| number | title | password | date | hit | content | id |
+-----+-----+-----+-----+-----+
| 2 | sdefrs | | 2023-03-13 07:08:40 | 0 | | 1111 |
| 5 | 222 | | 2023-03-13 08:35:49 | 0 | | aaa |
| 8 | i*ë...i*i,i5" | | 2023-03-13 12:50:10 | 0 | i%mi,,0i^'i0-i*ë<^ë<u^^ | i-i,i,- |
+-----+-----+-----+-----+-----+
3 rows in set (0.000 sec)
```

▲ 도메인 주소를 이용하여 타 PC의 웹 서버와의 DB간의 연동도 정상적으로 작동

## - DB - 갈레라 클러스터

```
MariaDB [(none)]> show global status like 'wsrep_cluster_size';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| wsrep_cluster_size | 3 |
+-----+-----+
1 row in set (0.001 sec)
```

```
[root@db1 ~]# mysql
Welcome to the MariaDB monitor.  Commands
Your MariaDB connection id is 509
Server version: 10.4.28-MariaDB MariaD
Copyright (c) 2000, 2018, Oracle, Mari
Type 'help;' or '\h' for help. Type '\
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| abc      |
| cacti    |
| db_board |
| information_schema |
| mysql    |
| performance_schema |
| test     |
| test2    |
| test3    |
| wordpress |
+-----+
10 rows in set (0.001 sec)
```

```
[root@db2 ~]# mysql
Welcome to the MariaDB monitor.  Commands
Your MariaDB connection id is 431
Server version: 10.4.28-MariaDB MariaDB Se
Copyright (c) 2000, 2018, Oracle, MariaDB
Type 'help;' or '\h' for help. Type '\c' t
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| abc      |
| cacti    |
| db_board |
| information_schema |
| mysql    |
| performance_schema |
| test     |
| test2    |
| test3    |
| wordpress |
+-----+
10 rows in set (0.001 sec)
```

```
[root@db3 ~]# mysql
Welcome to the MariaDB monitor.
Your MariaDB connection id is
Server version: 10.4.28-Maria
Copyright (c) 2000, 2018, Ora
Type 'help;' or '\h' for help.
MariaDB [(none)]> show databa
+-----+
| Database |
+-----+
| abc      |
| cacti    |
| db_board |
| information_schema |
| mysql    |
| performance_schema |
| test     |
| test2    |
| test3    |
| wordpress |
+-----+
10 rows in set (0.000 sec)
```

▲ 갈레라 클러스터를 사용하여 총 3개의 클러스터를 사용중. 각 클러스터들은 실시간으로 동기화 중.

## 팀 프로젝트 -- 방송 플랫폼 클라우드 확장

프로젝트 기간	2023.05.04~2023.05.11
개발 인원	5명
요 약	클라우드 환경(AWS, GCP)의 인프라 구축
담당 기능	1) RDS 2) GSLB 3) DMS
링크	<a href="https://kyounggu.tistory.com/70">https://kyounggu.tistory.com/70</a>

### 1. 프로젝트 개요

#### 1.1 개발 기획

현재 많은 회사들이 자신들의 데이터들을 온-프레미스가 아닌 클라우드에 저장할 하여 보다 간편하게 유지보수를 하며, 리소스에 제약을 덜 받고, 재해 복구나 장애 처리, 백업 등을 편리하게 사용하는 추세이다. 그 중에서도 아마존의 AWS를 가장 많이 사용하고 있기 때문에, 실제 상황을 가정하여 온-프레미스 서버를 사용하던 고객의 데이터를 클라우드로 옮기고 고객이 클라우드에 데이터를 저장하고 사용할 수 있도록 프로젝트를 구성했다.

#### 1.2 목표 :

실습에서 배운 내용들을 토대로 IAM으로 각자 역할을 부여받고 오프-프레미스인 AWS에 온-프레미스인 OpenStack과 GCP를 VPN으로 연결하여 서로 통신이 되는 것이 1차적 목표이다. 그 후, 각각의 온-프레미스에서 AWS에 있는 S3와 AuroraDB를 공유하는 것과 GSLB로 재해 복구를 하는 것이 2차 목표이다. 여기서 각자 실습 외 내용을 추가하여 S3 Glacier와 DMS를 사용하여 데이터를 보관하고, CloudWatch를 AWS가 아닌 온-프레미스까지 모니터링 하도록 확장 시키는 것이 최종적인 목표이다.

#### 1.3 담당 요구사항 :

- RDS

1. DB는 용량의 제약을 받으면 안된다.

⇒ Amazon Aurora를 사용하면 DB는 용량 제한 없이 사용 가능하다.

2. DB는 장애가 발생해도 자동으로 장애 복구(Fail over)를 할 수 있어야 한다.

⇒ Amazon Aurora를 사용하면 다중 AZ를 통해 서로 다른 가용 영역에 동기식 예비 복제본을 프로비저닝하고 유지하여 Master DB에 장애 발생 시, Standby 인스턴스를 Master DB로 승격시켜 장애 조치를 자동으로 수행한다.

3. DB의 성능을 효율적으로 높여야한다.

⇒ Amazon Aurora는 라이터(Writer) Instance와 리더(Reader) Instance로 인스턴스를 나눠, CRUD는 라이터 인스턴스로, 비교적 트래픽이 많은 읽기는 리더 인스턴스로 진행하여 트래픽을 분산하여 효율성을 올릴 수 있다. 또한, 오토 스케일링을 통해 CPU 부하도 분산 시킬 수 있다.

- **GSLB**

1. 웹 서버가 리전 단위로 다운돼도 다른 리전을 통해 고객들이 계속 웹 서버를 사용할 수 있어야 한다.

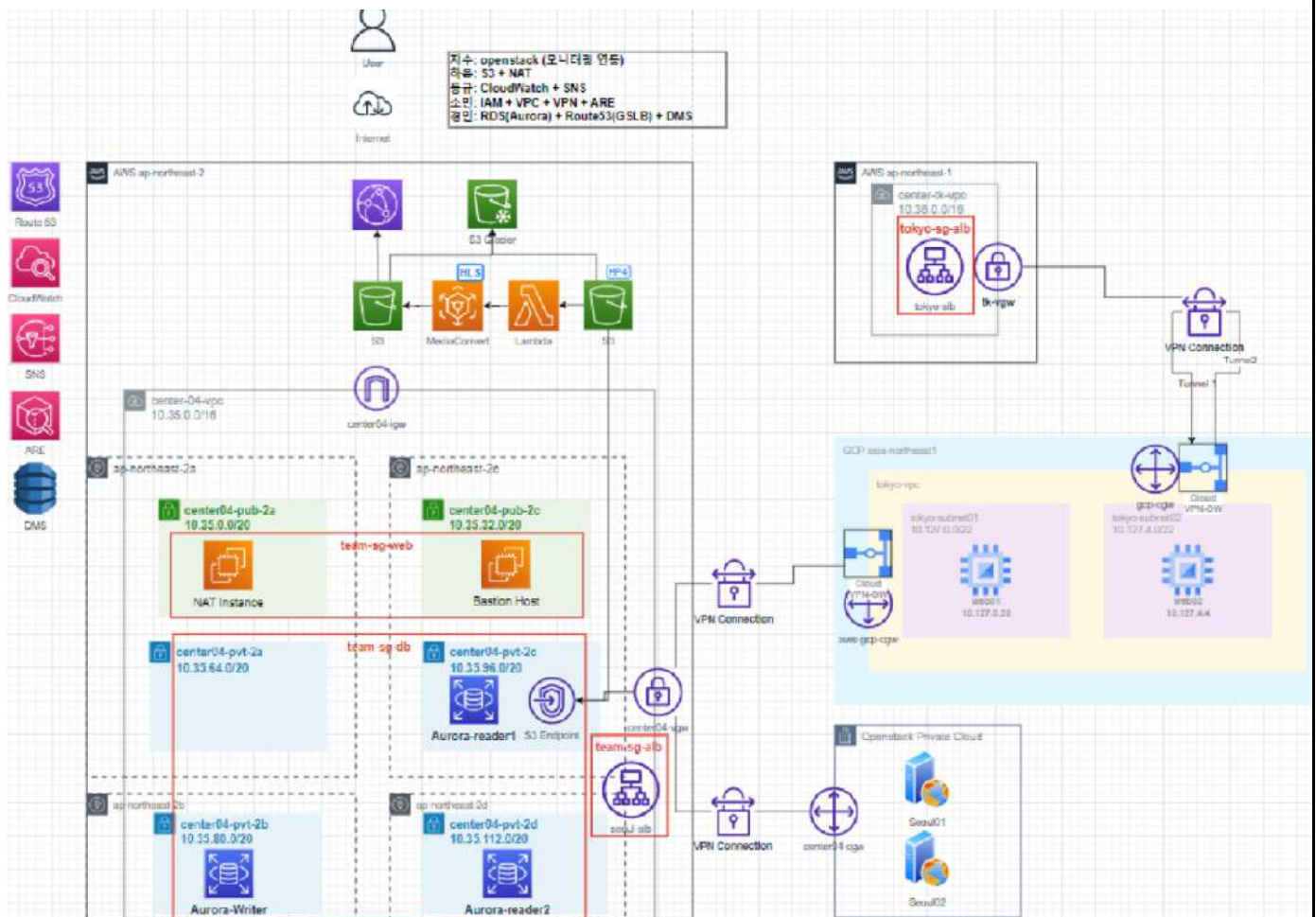
⇒ GSLB를 이용하여 리전이 다른 온-프레미스들한테 로드 밸런싱을 하여, 서울 리전이 다운돼도 고객들은 접속 방법의 차이 없이 자동으로 도쿄 리전을 통해 웹 서버를 이용할 수 있다.

- **DMS**

1. 간편하게 백업 및 이식이 가능해야 한다.

⇒ DMS를 이용하여 마이그레이션 태스크를 생성하면, 설정해둔 소스 DB에서 타겟 DB로 간편하게 버튼 하나로 백업 및 이식을 할 수 있다.

#### 1.4 전체 Architecture :



## 2. 구축 결과

### 2.1 Amazon Aurora:

### Amazon Aurora

관리 콘솔 화면 캡처

관련

DB 인스턴스

DB 인스턴스	인스턴스 유형	엔진	리더 AZ	크기	상태	작업	CPU	읽기 활동	유지 관리	VPC	다중 AZ	최대 백업 시간
aurora	리더 인스턴스	Aurora MySQL	ap-northeast-2	2 인스턴스	사용 가능	2 작업	-	-	-	-	-	-
aurora-instance-1	리더 인스턴스	Aurora MySQL	ap-northeast-2b	db.t3.small	사용 가능	-	14.23%	0.00/0.00	ap-northeast-2b	ap-northeast-2b	3 영역	-
aurora-read1	리더 인스턴스	Aurora MySQL	ap-northeast-2a	db.t3.small	사용 가능	-	12.94%	0.00/0.00	ap-northeast-2a	ap-northeast-2a	3 영역	24.071 일리온
aurora-read2	리더 인스턴스	Aurora MySQL	ap-northeast-2a	db.t3.small	사용 가능	-	18.37%	0.00/0.00	ap-northeast-2a	ap-northeast-2a	3 영역	43.056 일리온

- 서울 리전에 Aurora DB를 생성.
- 다중 AZ를 활성화 하여 장애 복구 성능을 증가.
- 읽기 복제본을 추가하여 리더 인스턴스 1개 / 리더 인스턴스 2개를 최소 용량으로 지정
- 오토 스케일링을 통해 규칙에 따라 인스턴스를 줄이거나 늘릴 수 있다.

```

MySQL [(none)]> create database dongkyu;
Query OK, 1 row affected (0.05 sec)

MySQL [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| dongkyu |
| mysql |
| performance_schema |
| sys |
| test |
| wordpress |
+-----+
7 rows in set (0.04 sec)

```

GCP에서 'dongkyu' 데이터베이스 생성

```

MySQL [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| dongkyu |
| mysql |
| performance_schema |
| sys |
| test |
| wordpress |
+-----+
7 rows in set (0.01 sec)

```

OpenStack에서 정상적으로 확인 가능

### 클러스터 용량 세부 정보

최소 용량  
유지할 Aurora 복제본의 최소 수를 지정합니다.

3 Aurora 복제본

최대 용량  
유지할 Aurora 복제본의 최대 수를 지정합니다. 최대 15개 Aurora 복제본이 지원됩니다.

15 Aurora 복제본

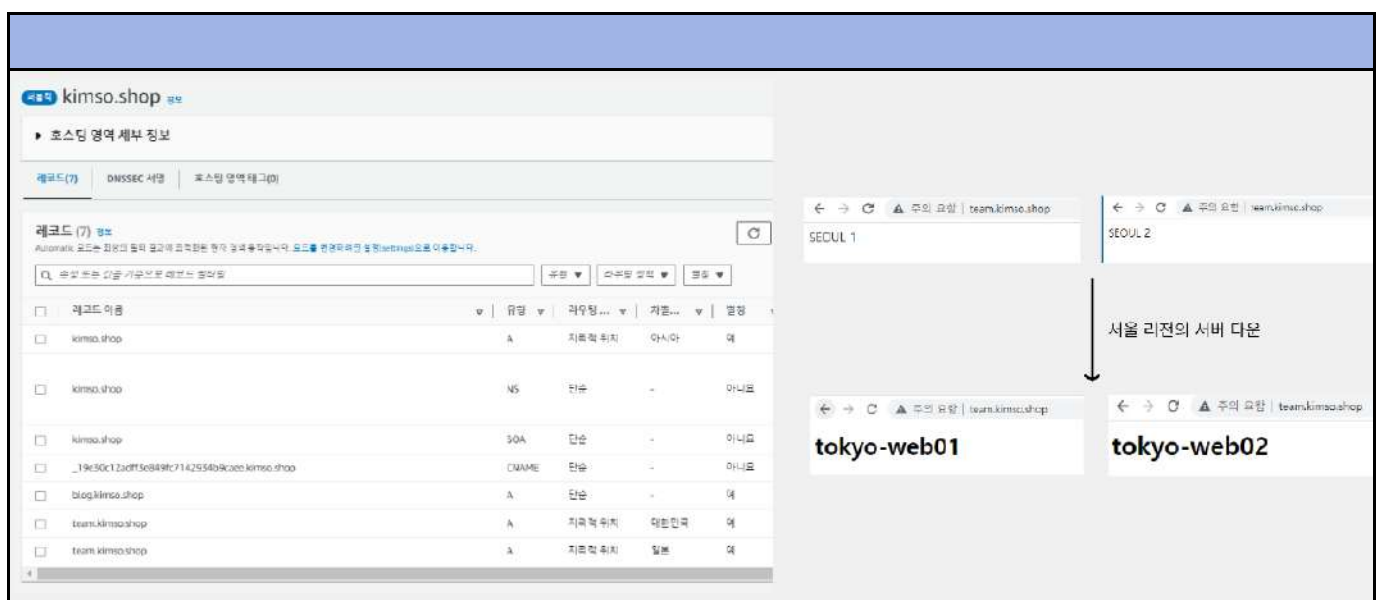
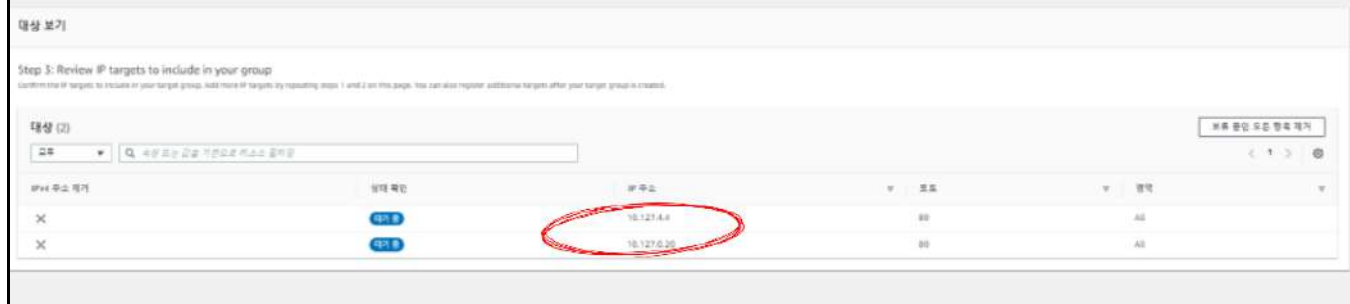
오토 스케일링을 통해 지정된 클러스터 용량에 맞춰 인스턴스 생성





## 2.3 GSLB:

- 서울 리전의 ALB와 도쿄 리전의 ALB 생성
  - 대상 그룹 선택 시, 온-프레미스의 웹 서버 아이피를 등록
- Route 53으로 각 ALB를 지리적 라우팅



▲ 가장 가까운 서울 리전의 서버가 다운돼도, 다음으로 가장 가까운 리전인 도쿄 리전의 서버로 로드 밸런스된다.

## 개인 프로젝트 – 온라인 RPG 게임 구조를 기반으로 한 데이터베이스 구현

개발 기간	2019.11.04 – 2019.12.09
개발 환경	Oracle
개발 인원	1명
요 약	온라인 게임의 데이터 구조를 이해하고 이를 E-R 다이어그램으로 명세화를 하여 이를 기반으로 오라클에 구현하여 실제 데이터 조작 명령어를 통해 게임에서의 데이터 상호작용을 구현함.
기능	1) 캐릭터, 몬스터, 스킬, 맵 등등의 개체 테이블들의 튜플 생성, 삭제, 변경 2) 데이터 변동에 의한 연관된 데이터 변동 구현 3) 파티, 길드, NPC위치 등등을 위한 관계 구현

### 1. 개발 기획

	~19.11.11							~19.11.18							~19.11.25							~19.12.02							~19.12.09						
분석																																			
기획 – 요구사항 분석																																			
기획 – 개체 분석																																			
기획 – 관계 분석																																			
설계																																			
E-R 다이어그램 설계																																			
구현																																			
DB 구현																																			
테스트																																			

**1.1 요구사항 분석 :** RPG게임에서는 대부분의 행동의 주체가 되는 플레이어와 플레이어와 관련되어있는 여러 요소들(서버, 아이템, 스킬, 친구, 계정 몬스터, 직업과 같은 플레이어정보외 등등)들을 포함하고 있어야 한다. 정상적인 사용이 가능하도록 무결성을 지켜야한다.

### 1.2 개체 분석 :

▣'서버'는 하나의 개체이며, 기본키로는 서버이름을 가져야한다. 그리고 이 서버는 속성으로 서버 생성일 등등을 가져야 한다.

▣'채널'은 하나의 개체이며, 서버라는 개체에 종속되는 약한 개체다. 채널은 부분키로 채널 번호를 가진다(ex. channel 1, channel 2 등)

채널의 경우, 속성으로 접속중인 인원수를 가지며, 이를 토대로 서버상태(ex. 패적, 보통, 혼잡)를 표시하는 속성을 가져서 플레이어에게 보다 쉽게 채널 상태를 보여줘야한다.(유도 속성) 단, 유도속성은 아직 스키마로 표현 불가.

▣'계정'은 하나의 개체이며, 기본키로 계정아이디를 가진다. 그 외에는 속성으로 비밀번호, 주민등록번호, 이름 등을 가진다.

▣'캐릭터'는 하나의 개체이며, 닉네임을 기본키로 가진다. 그 외에도 속성으로 레벨, 캐릭터생성일, 소지금, 경험치 등을 가져야한다.

▣'직업'은 하나의 개체이다. 직업코드를 기본키로 가진다. 속성으로 직업이름과 직업계열을 가진다.

▣'스킬'은 하나의 개체이며, 스킬코드를 기본키로 가진다. 속성으로 스킬이름, 스킬최대레벨, 스킬타입(ex. 패시브, 액티브, 버프) 복합속성인 스킬내용(데미지 , 타겟 수, 강화능력, 쿨타임)을 가진다.

▣'아이템'은 하나의 개체이며, 아이템코드를 기본키로 가진다. 속성으로 아이템 이름, 아이템의 타입(장비, 소비, 기타), 교환가능 여부(거래가능, 거래불가, 계정내의 거래 가능), 아이템 효과(복합속성으로, 회복, 강화를 포함)를 가진다.

▣'파티'개체. 기본키로는 파티코드를 가진다. 그리고 속성은 이름, 인원수를 갖는다.

▣'NPC' 개체. 기본키는 NPC코드를 가지며, 속성으로 이름, 대화 내용, 유형을 갖는다.

▣'맵' 개체. 기본키로는 맵코드를 가지며, 속성으로 맵이름을 갖는다.

▣'몬스터' 개체. 기본키로는 몬스터코드를 가지며, 속성으로 이름, 경험치, 레벨, 유형, 돈 를 갖는다.

▣'퀘스트' 개체. 기본키로는 퀘스트코드를 가지며, 속성으로 이름, 내용, 레벨제한, 경험치를 갖는다.

▣'길드' 개체 추가. 기본키로는 길드이름을 가지며, 속성으로 인원수, 길드레벨을 갖는다.

### 1.3 관계 분석

▣캐릭터와 계정은 '생성'이라는 관계를 갖는다. 한계정은 여러 캐릭터를 생성하고, 한 캐릭터는 하나의 계정에 의해서 만 생성된다(1:다 관계). 그리고 캐릭터가 필수적 참여이다.

▣캐릭터와 서버는 '서버소속'이라는 관계를 갖는다. 한 서버는 여러 캐릭터를 소속시키고, 한 캐릭터는 하나의 서버에만 소속된다(1:다 관계). 그리고 캐릭터가 필수적 참여이다.

▣서버와 채널은 '포함'이라는 관계를 갖는다. 한 서버는 여러 채널을 포함하지만, 한 채널은 하나의 서버에만 포함된다(1:다 관계). 그리고 채널이 필수적 참여이다.

■캐릭터는 스스로와 '친구'라는 관계를 갖는다. 다:다 관계(친구신청, 친구수락)이다.

■캐릭터와 직업은 '직업소속'이라는 관계를 갖는다. 한 직업은 여러 캐릭터를 소속시키고, 한 캐릭터는 하나의 직업에만 소속된다. (1:다 관계). 그리고 캐릭터는 필수적 참여이다.

■직업은 스킬과 '스킬소유'라는 관계를 갖는다. 다:다 관계이며(why? 공통스킬), 스킬은 필수적 참여이다.

■캐릭터와 아이템은 '소유'라는 관계를 갖는다. (이중관계) 다:다 관계이며, 속성으로 개수를 갖는다.

■캐릭터와 아이템은 '거래'라는 관계를 갖는다. (이중 관계) 다:다 관계이며, 속성으로 상대이름, 개수를 가진다.

■캐릭터와 파티는 '파티소속'이라는 관계를 갖는다. 1:다 관계(캐릭터가 다)이다.

■캐릭터와 길드는 '길드소속'이라는 관계를 갖는다. 1:다 관계(캐릭터가 다)이며, 속성으로 직위를 갖는다.

■캐릭터와 NPC는 '상호작용'이라는 관계를 갖는다. 다:다 관계이다.

■캐릭터와 맵은 '캐릭터위치'라는 관계를 갖는다. 1:다 관계(캐릭터가 다)이며, 캐릭터는 필수적 참여를 한다.

■NPC와 맵은 'NPC위치'라는 관계를 갖는다. 1:다 관계(NPC가 다)이며, 속성으로 좌표를 갖는다.

■몬스터와 맵은 '몬스터위치'라는 관계를 갖는다. 다:다 관계이며, 속성으로 좌표를 갖는다.

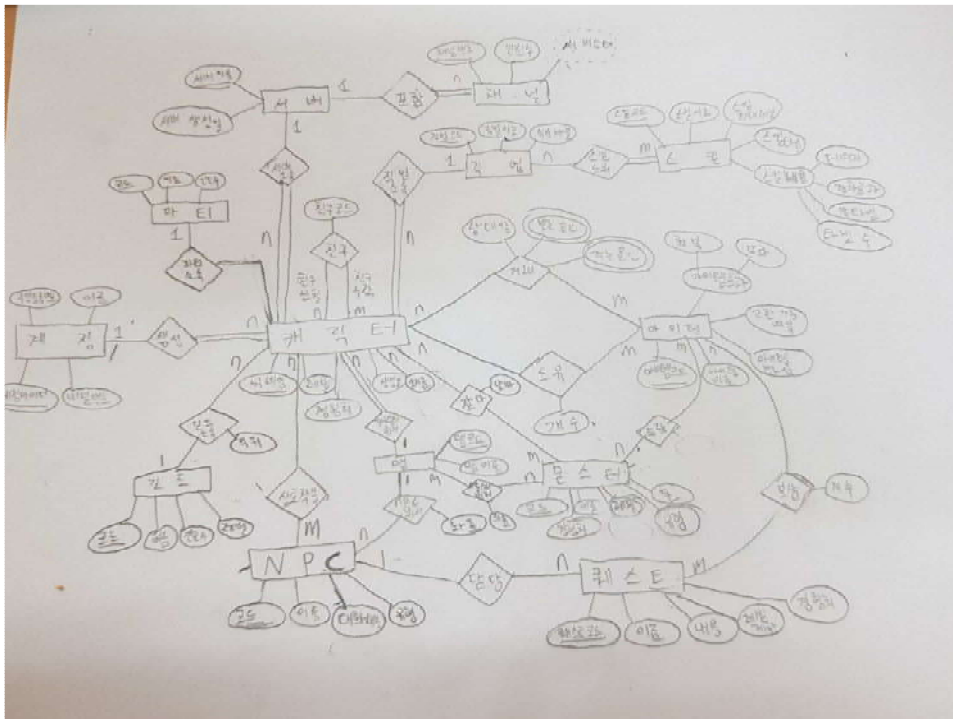
■NPC와 퀘스트는 '담당'이라는 관계를 갖는다. 1:다 관계(퀘스트가 다)이다.

■캐릭터와 몬스터는 '잡다'라는 관계를 갖는다. 다:다 관계이며, 날짜를 속성으로 갖는다.

■몬스터와 아이템은 '드랍'이라는 관계를 갖는다. 다:다 관계이다.

■퀘스트와 아이템은 '보상'이라는 관계를 갖는다. 다:다 관계이며, 속성으로 개수를 갖는다.

## 2. E-R 다이어그램 설계





### 3. DB 구현

#### 3.1 ----- 개체

Code	해설
<pre> create table 계정 (     계정아이디 varchar2(40) primary key,     비밀번호 varchar2(40) not null,     이름 varchar2(20) not null,     주민등록번호 char(14) not null     check (주민등록번호 like '9-99-999999999-9999999999') );  alter table 계정 add constraint 주민등록번호유일 unique(주민등록번호); </pre>	<p>주민등록번호는 일정한 형태 이므로 제약 형식상의 제약조건 을 줌. 그리고 주민등록번호는 유일하므로 유일성을 부여함.</p>
<pre> create table 파티 (     파티코드 number(6) primary key,     이름 varchar2(30) not null,     인원수 number(1) not null );  alter table 파티 add constraint 파티인원수 check (인원수 between 1 and 6); </pre>	<p>나중에 인원수에 제약을 두었 습니다. (최대 6명)</p>
<pre> create table 캐릭터 (     닉네임 varchar2(10) primary key,     레벨 number(3)     check (레벨 between 1 and 100),     생년월일 date,     소지금 number(10) not null     check (소지금 between 1 and 999999999999),     계정아이디 varchar2(40) not null,     파티코드 number(6) null,     서버이름 varchar2(10) not null,     길드이름 varchar2(20) null,     길드직위 varchar2(10) null,     직업코드 number(4) not null,     맵코드 number(5) not null,     constraint 캐릭소속계정 foreign key (계정아이디) references 계정 (계정아이디) on delete cascade,     constraint 캐릭소속파티 foreign key (파티코드) references 파티 (파티코드),     constraint 캐릭소속서버 foreign key (서버이름) references 서버 (서버이름),     constraint 캐릭소속길드 foreign key (길드이름) references 길드 (길드이름),     constraint 캐릭직업 foreign key (직업코드) references 직업 (직업코드),     constraint 캐릭위치맵 foreign key (맵코드) references 맵 (맵코드) );  alter table 캐릭터 add 경험치 number(20); </pre>	<p>게임의 주체인 만큼 외래키가 많아서 가장 많은 제약조건을 걸었습니다. 레벨은 1~100으로 제약을 걸어 놔고, 소지금도 최대 100억 미 만을 가지도록 제약을 걸었습니 다. 그리고 소속되는 계정이 삭제되 면 캐릭터도 따라 삭제되도록 했습니다. 경험치는 추후 따로 추가하였습 니다.</p>
<pre> create table 길드 (     길드이름 varchar2(20) primary key,     인원수 number(3) not null,     레벨 number(2) not null     check (레벨 between 1 and 20) ); </pre>	

```
create table 서버(
  서버이름 varchar2(10) primary key,
  서버생성일 date not null
);
```

```
create table 채널(
  채널번호 number(2),
  서버이름 varchar2(10),
  인원수 number(20),
  constraint 채널의서버
  foreign key (서버이름)
  references 서버(서버이름)
  on delete cascade,
  constraint 채널pk primary key(채널번호, 서버이름)
);
```

소속되는 서버가 삭제되면 따라 삭제되도록 했습니다.

```
create table 직업(
  직업코드 number(4) primary key,
  직업이름 varchar2(20),
  직업계열 varchar2(20)
);
```

```
create table 스킬(
  스킬코드 number(5) primary key,
  스킬이름 varchar2(20) not null,
  스킬최대레벨 number(2) not null,
  스킬타입 varchar2(15) not null,
  데미지 number(4) null,
  강화효과 varchar2(10) null,
  쿨타임 number(4) null,
  타겟수 number(2) null
);
alter table 스킬
modify 강화효과 varchar2(30);
```

스킬은 유형에 따라 데미지, 강화효과가 없을 수도 있기 때문에, NULL값을 허용했습니다. 그리고 강화효과 크기 확대를 했습니다.

```
create table NPC(
  NPC코드 number(5) primary key,
  NPC이름 varchar2(20),
  대화내용 varchar2(3000) null,
  유형 varchar2(20),
  맵코드 number(5) null,
  좌표 varchar2(20) null
  check (좌표 like 'x-y'),
  constraint NPC맵
  foreign key (맵코드)
  references 맵(맵코드)
);
```

NPC를 우선 만들어놓고 맵에 두지 않는 개발단계를 고려해서 맵, 좌표는 NULL값을 허용했습니다. 그리고, 좌표의 경우 X, Y 축의 형태로 CHECK를 써서 제약을 두었습니다. 그리고 npc 이름에 중복을 허용한 이유는, 같은 npc가 다른 역할을 할 경우를 고려한 것입니다. (ex. 진행에 따른 대화내용 변경)

```
create table 맵(
  맵코드 number(5) primary key,
  맵이름 varchar2(20) null
);
```

```
create table 몬스터(
  몬스터코드 number(5) primary key,
  몬스터이름 varchar2(20),
  몬스터레벨 number(3) not null,
  돈 number(5) null,
  유형 varchar2(20) not null,
  경험치 number(20) null
);
```

<pre> create table 퀘스트(   퀘스트코드 number(5) primary key,   퀘스트이름 varchar2(20) not null,   퀘스트내용 varchar2(3000) null,   레벨제한 number(3) not null     check (레벨제한 between 1 and 100),   경험치 number(20)  );  alter table 퀘스트 add constraint 담당npc foreign key (npc코드) references npc(npc코드); </pre>	<p>(나중에 npc코드 외래키를 추가했습니다.)</p> <p>도메인 무결성을 생각해서 CHECK로 레벨제한에 제약을 두었습니다.</p>
<pre> create table 아이템(   아이템코드 number(5) primary key,   아이템이름 varchar2(20) not null,   아이템타입 varchar2(20) not null,   교환가능여부 varchar2(10) not null,   회복 number(6) null,   강화 varchar2(20) null ) </pre>	<p>아이템의 경우, 스킬과 비슷하게 회복이나 강화 효과가 없는 경우를 생각해서 NULL값을 허용.</p>

### 3.2 ----- 관계

Code	해설
<pre> create table 친구(   신청캐릭터 varchar2(10),   수락캐릭터 varchar2(10),   constraint 친구신청   foreign key (신청캐릭터)   references 캐릭터 (닉네임),   constraint 친구수락   foreign key (수락캐릭터)   references 캐릭터 (닉네임) ); </pre>	
<pre> create table 스킬소유(   직업코드 number(4),   스킬코드 number(5),   constraint 소유직업   foreign key (직업코드)   references 직업(직업코드),   constraint 직업스킬   foreign key (스킬코드)   references 스킬(스킬코드) )  alter table 스킬소유 add constraint 스킬소유키 primary key (스킬코드, 직업코드) </pre>	<p>-추가로 나중에 ALTER를 써서 기본키를 설정했습니다.</p>
<pre> create table 잡다(   닉네임 varchar2(10),   몬스터코드 number(5),   날짜 date,   constraint 잡은캐릭터   foreign key (닉네임)   references 캐릭터 (닉네임),   constraint 잡힌몬스터   foreign key (몬스터코드)   references 몬스터(몬스터코드) ); </pre>	<p>오류나 비정상적인 행동을 잡기 위해 로그를 남기도록 하는 테이블입니다.</p>

```

create table 몬스터위치(
  맵코드 number(5),
  몬스터코드 number(5),
  좌표 varchar2(20) not null
    check (좌표 like '%,%'),
  constraint 몬스터맵
  foreign key (맵코드)
  references 맵 (맵코드),
  constraint 맵몬스터코드
  foreign key (몬스터코드)
  references 몬스터 (몬스터코드),
  constraint 몬스터위치키
  primary key (맵코드, 몬스터코드)
);

```

-NPC와 같은 이유로 좌표값에  
형식을 넣었습니다.

```

create table 거래(
  닉네임 varchar2(10) not null,
  아이템코드 number(5) null,
  상대이름 varchar2(10) not null,
  개수 number(3) null,
  constraint 거래하는캐릭터
  foreign key (닉네임)
  references 캐릭터 (닉네임),
  constraint 거래아이템
  foreign key (아이템코드)
  references 아이템 (아이템코드),
  constraint 거래상대이름
  foreign key (상대이름)
  references 캐릭터 (닉네임)
);

```

```

alter table 거래
add constraint 거래키 primary key (닉네임,아이템코드);

```

기본키를 alter로 추가했습니다.

```

create table 소유(
  닉네임 varchar2(10) not null,
  아이템코드 number(5) not null,
  개수 number(3) not null,
  constraint 소유하는캐릭터
  foreign key (닉네임)
  references 캐릭터 (닉네임),
  constraint 소유아이템
  foreign key (아이템코드)
  references 아이템 (아이템코드)
);

```

```

alter table 소유
add constraint 소유키 primary key (닉네임,아이템코드);

```

기본키를 alter로 추가했습니다

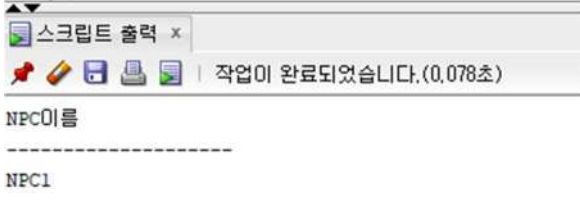
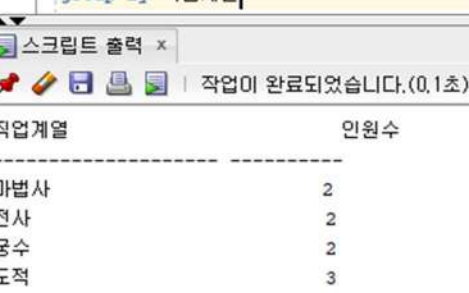
```

create table 보상(
  아이템코드 number(5),
  퀘스트코드 number(5),
  개수 number(3),
  constraint 보상아이템
  foreign key (아이템코드)
  references 아이템 (아이템코드),
  constraint 보상있는퀘스트
  foreign key (퀘스트코드)
  references 퀘스트 (퀘스트코드),
  constraint 보상기본키
  primary key (아이템코드, 퀘스트코드)
);

```



#### 4 ----- 테스트

Code 및 결과	해설																		
<pre> insert into 거래 values ('캐릭터3',10001,'캐릭터6',1); insert into 거래 values ('캐릭터6',20001,'캐릭터3',1);  insert into 스킬 values (10001, '전사액티브1', 10, '액티브', 100, null, null, 3); insert into 스킬 values (10002, '전사액티브2', 20, '액티브', 300, null, null, 5); insert into 스킬 values (10101, '전사패시브1', 10, '패시브', null, '공격력증가', null, null); insert into 스킬 values (20001, '마법사액티브1', 10, '액티브', 500, null, 5, 10); insert into 스킬 values (20002, '마법사액티브2', 20, '액티브', 200, null, null, 5); insert into 스킬 values (30001, '도적액티브1', 5, '액티브', 200, null, null, 3); insert into 스킬 values (30101, '도적패시브1', 10, '패시브', null, '회피율증가', null, null); insert into 스킬 values (40001, '궁수액티브1', 10, '액티브', 1000, null, 10, 1); insert into 스킬 values (40101, '궁수패시브1', 20, '패시브', null, '속도증가', null, null); insert into 스킬 values (40102, '궁수패시브2', 20, '패시브', null, '공격력증가', null, null); </pre>	여러 테스트 데이터들이 데이터베이스에 입력됨.																		
<pre> select npc이름 from npc where npc코드 = (select npc코드 from npc where npc코드 in (select npc코드 from 퀘스트)) </pre>  <p>스크립트 출력 x</p> <p>작업이 완료되었습니다.(0,078초)</p> <p>NPC이름</p> <p>-----</p> <p>NPC1</p>	퀘스트를 담당하는 npc 출력																		
<pre> select 직업코드, count(닉네임) 인원 from 캐릭터 group by 직업코드 order by 직업코드 </pre>  <p>스크립트 출력 x</p> <p>작업이 완료되었습니다.(0,01초)</p> <table border="1"> <thead> <tr> <th>직업코드</th><th>인원</th></tr> </thead> <tbody> <tr><td>1001</td><td>1</td></tr> <tr><td>1002</td><td>1</td></tr> <tr><td>2001</td><td>1</td></tr> <tr><td>2002</td><td>1</td></tr> <tr><td>3001</td><td>2</td></tr> <tr><td>3002</td><td>1</td></tr> <tr><td>4001</td><td>1</td></tr> <tr><td>4002</td><td>1</td></tr> </tbody> </table>	직업코드	인원	1001	1	1002	1	2001	1	2002	1	3001	2	3002	1	4001	1	4002	1	직업코드에 따른 캐릭터수를 직업코드 오름차순으로 출력
직업코드	인원																		
1001	1																		
1002	1																		
2001	1																		
2002	1																		
3001	2																		
3002	1																		
4001	1																		
4002	1																		
<pre> select 직업계열, count(직업계열) 인원수 from 캐릭터, 직업 where 캐릭터.직업코드 = 직업.직업코드 group by 직업계열 </pre>  <p>스크립트 출력 x</p> <p>작업이 완료되었습니다.(0,1초)</p> <table border="1"> <thead> <tr> <th>직업계열</th><th>인원수</th></tr> </thead> <tbody> <tr><td>마법사</td><td>2</td></tr> <tr><td>전사</td><td>2</td></tr> <tr><td>궁수</td><td>2</td></tr> <tr><td>도적</td><td>3</td></tr> </tbody> </table>	직업계열	인원수	마법사	2	전사	2	궁수	2	도적	3	현재 캐릭터들의 직업계열들과 인원수 출력								
직업계열	인원수																		
마법사	2																		
전사	2																		
궁수	2																		
도적	3																		



```

select 직업계열
from (select 직업계열, count(직업계열) 인원수
from 캐릭터, 직업
where 캐릭터.직업코드 = 직업.직업코드
group by 직업계열)
where 인원수 in (select max(인원수) from (select 직업계열, count(직업계열) 인원수
from 캐릭터, 직업
where 캐릭터.직업코드 = 직업.직업코드
group by 직업계열))

```

가장 인기있는 직업계열 출력

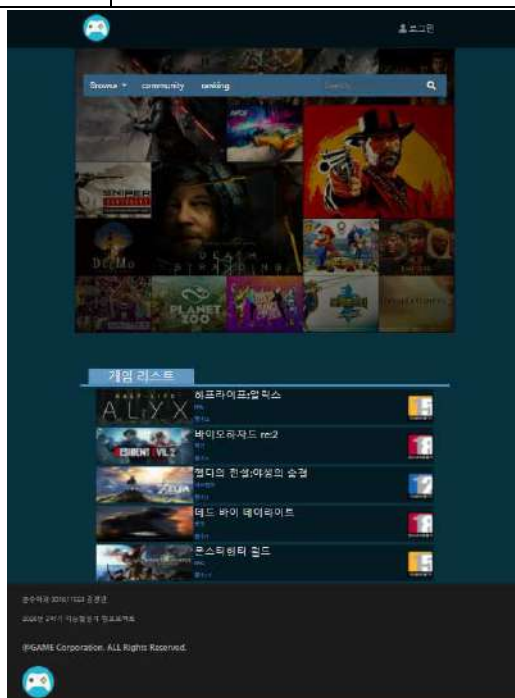
스크립트 출력 x 자동 추적 x  
작업이 완료되었습니다.(0.125초)

직업계열

도적

## 개인 프로젝트 – 데이터베이스를 이용한 게임 리뷰 커뮤니티 사이트

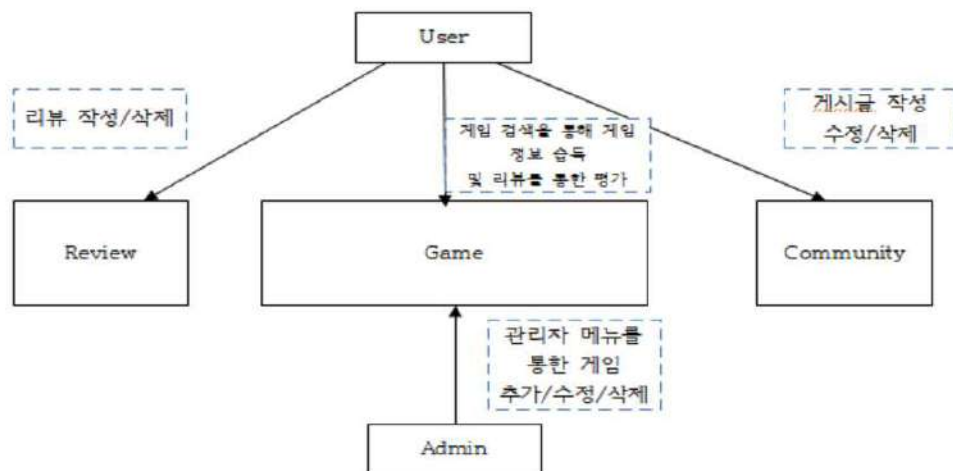
개발 기간	2020.09.28 – 2020.12.06
개발 환경	eclipseIDE / Javascript, JSP
개발 인원	1명
요약	게임과 관련된 여러 기능을 제공하고 데이터베이스와 연동되어 유저들 간의 리뷰와 게시글, 댓글들이 저장되며, 어드민이 관리 가능한 실무적인 웹페이지를 설계
기능	<ol style="list-style-type: none"> <li>1) 데이터베이스 연동</li> <li>2) 게시글/댓글 작성, 삭제, 편집</li> <li>3) 게임 소개 및 유저 평가를 실시간으로 반영하여 평점 출력</li> <li>4) 게임 추천 및 장르별 출력, 평점순 출력</li> <li>5) 운영자는 게임 추가, 삭제, 편집이 가능하고 유저의 게시글/댓글 삭제 가능</li> <li>6) 쿠키를 이용하여 최근에 본 항목 출력</li> </ol>
링크	코드: <a href="https://github.com/kyounggudev/ReviewSite">https://github.com/kyounggudev/ReviewSite</a>



## 1. 요구사항 :

- 게임 정보와 게임 리뷰를 모두 표시
  - 게임 정보와 더불어 그 게임에 대한 사람들의 생각을 볼 수 있음.
  - 게임 점수는 리뷰를 토대로 결정되며 실시간으로 이루어짐.
- 게임 검색, 장르별 게임, 게임 순위에 따른 리스트 출력
  - 사람들이 보다 쉽게 원하는 게임을 찾을 수 있음.
  - 게임 순위를 보고 게임 평판을 전체적으로 알 수 있음.
- 커뮤니티 구현
  - 리뷰 활동과 더불어 커뮤니티 활동도 장려함.
- 관리자 모드 구현
  - 새로 나온 게임들을 추가 가능
  - 기존 게임들 수정 가능
- 예외 페이지를 통한 예외처리


## 2. 설계 :

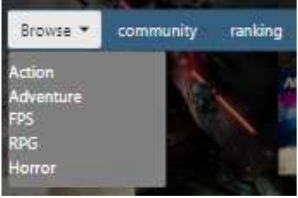

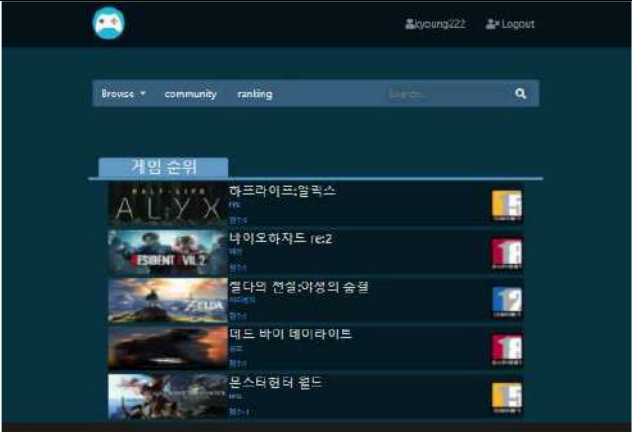
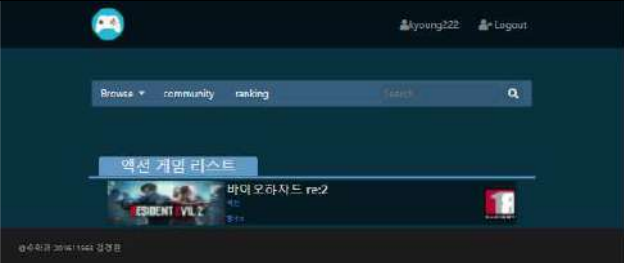


시스템	기능 및 설명
User	<ul style="list-style-type: none"> <li>- 회원 가입을 통해 DB에 추가</li> <li>- 정규 표현식을 통해 회원 가입 시 양식 추가</li> <li>- 쿠키를 통해 최근 본 게임, 최근 리뷰를 남긴 게임 출력</li> </ul>
Admin	<ul style="list-style-type: none"> <li>- 관리자 메뉴를 통해 게임 추가/삭제/수정</li> <li>- 시큐리티로 만든 계정</li> </ul>
Game	<ul style="list-style-type: none"> <li>- 게임 정보들 출력</li> <li>- 해당 게임의 리뷰들도 출력</li> <li>- 리뷰를 기반으로 점수 생성</li> <li>- 없는 게임은 예외처리</li> </ul>

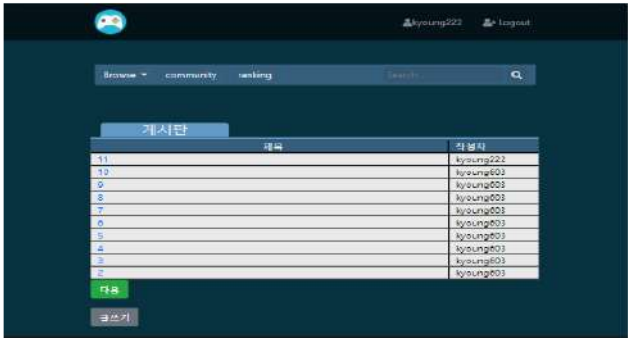
Community	<ul style="list-style-type: none"> <li>- 제목/작성자로 게시판 생성</li> <li>- 페이지 당 10개의 게시물</li> <li>- 다른 유저는 수정/삭제 불가</li> <li>- 항목 미 입력시 추가 불가</li> <li>- 없는 게시글은 예외 처리</li> </ul>
Review	<ul style="list-style-type: none"> <li>- 추천/비추천으로 게임 평가</li> <li>- 항목 미 입력시 추가 불가</li> <li>- 사용자 본인만 삭제 가능</li> <li>- 로그인 상태 + 해당 게임에 대한 리뷰가 없을 경우만 작성 가능</li> <li>- 비로그인 혹은 리뷰가 있을 경우 리뷰 생성창 미출력</li> </ul>
Menu	<ul style="list-style-type: none"> <li>- 장르별 게임 출력</li> <li>- 점수 순위순으로 게임 리스트 출력</li> <li>- 다른 링크 이동</li> <li>- 게임 검색 기능</li> <li>- 세션을 통해 로그인/비로그인 상태 구별</li> </ul>
AdminMenu	<ul style="list-style-type: none"> <li>- 시큐리티를 통해 관리자만 사용</li> <li>- 다국어 처리</li> <li>- 게임 추가/수정/삭제 시 정규식으로 양식 생성</li> </ul>

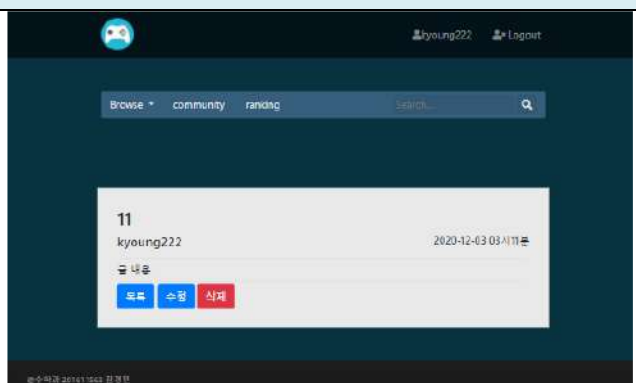
### 3. 구현 및 주요기능

1.1 게임 정보 (game.jsp)	설명
	<ul style="list-style-type: none"> <li>-게임 정보 : post형식으로 가져온 게임ID와 일치하는 게임을 DB에서 가져옴.</li> <li>- 추천 현황 : DB에서 해당 게임ID를 가진 리뷰들의 추천 현황을 가져옴.</li> <li>- 리뷰 작성 : 이미 작성한 적이 있거나, 비로그인 상태시 노출 안됨.</li> <li>- 리뷰 리스트: DB에서 가져온 해당 게임의 리뷰들을 출력 본인의 리뷰만 삭제 가능.</li> <li>- 삭제 버튼을 눌러 삭제 시, DB에 delete</li> </ul>

1.2 메뉴바 (main.jsp)	설명
	<p>전체적인 메뉴들을 포함</p> <ul style="list-style-type: none"> <li>- 'Brose'는 드롭다운 구성.</li> </ul>
1.3 게임 평가 (game.jsp)	설명
	<p>① localhost:8081/Term/game.jsp?id=G001</p> <p>다음과 같이 게임코드에 대응하는 게임 페이지 접속</p> <ul style="list-style-type: none"> <li>- 조건을 만족하면 이렇게 작성란이 노출.</li> </ul> <p>아무런 값을 입력 안할 시 작성 불가.</p> <ul style="list-style-type: none"> <li>- 작성 성고 시 , DB에 추가</li> </ul>
2. 게임 순위 (rank.jsp)	설명
	<p>점수를 기반으로 해당 게임의 순위 출력.</p>
3. 장르별 게임 (rank.jsp)	설명
	<p>- 'Browse'에서 택한 장르의 게임리스트 출력</p>

4. 최근정보(watchdlist.jsp)	설명
	<ul style="list-style-type: none"> <li>- 쿠키값을 통해 최근 본 게임 출력</li> <li>- 쿠키값을 통해 최근 리뷰를 남긴 게임 출력</li> </ul>

5.1 커뮤니티(watchdlist.jsp)	설명
	<ul style="list-style-type: none"> <li>- DB에서 게시글 목록을 가져옴</li> <li>- 게시글은 제목/작성자로 리스트에 출력</li> <li>- 게시글이 10개가 넘어갈 경우, 페이지 생성</li> <li>- 글쓰기 기능</li> </ul> <p><code>localhost:8081/Term/community.jsp?pageNumber=2</code></p> <ul style="list-style-type: none"> <li>- post로 'pageNumber'를 입력받아서 입력된 숫자의 페이지를 출력</li> </ul>

5.2 게시글(view.jsp)	설명
	<ul style="list-style-type: none"> <li>- 제목 / 작성자 / 날짜를 출력</li> <li>- 내용 출력</li> <li>- 작성자 본인만 수정/삭제가 가능</li> <li>- 다른 작성자의 게시글을 수정/삭제 버튼이 노출되지 않도록 설정.</li> </ul>

### 5.3 게시물 작성,수정 (addCommu.jsp)

#### 설명

- 제목/ 내용을 입력받는 폼 형식
- 아무것도 입력 안하면 글쓰기 불가능

localhost:8081 내용:  
입력이 안된 사항이 있습니다

확인

작성 성공시, DB에 추가

- 게시물 수정은 게시물 입력과 같은 형식
- 수정 성공시, DB에 update

### 6.1 로그인(login.jsp)

#### 설명

- 로그인 / 회원가입/ 관리자 로그인을 포함

- 항목 미 입력 시, 경고창 출력

### 6.2 회원가입(join.jsp)

#### 설명

- 아이디(키)/ 비밀번호 / 이름을 입력

- 유효성 검사로 양식을 지킴으로 만들.

- 이미 존재하는 아이디는 생성 불가.
- 가입 성공 시, DB에 추가



### 6.3 관리자 로그인(login\_admin.jsp)

#### 설명



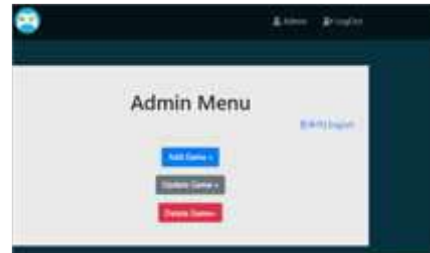
-관리자 메뉴 진입시, 설정한 시큐리티로 인해  
불러옴. 회원과는 다르게 시큐리티에 등록된  
'Admin'만 로그인 가능

### 7.1 관리자 메뉴(main\_admin.jsp)

#### 설명



-다국어 처리



-게임 추가/수정/삭제 메뉴가 출력

### 7.2 게임 추가(addGame.jsp)

### 게임 수정(updateGame.jsp)

#### 설명



- 전과 마찬가지로 다국어 처리

- 이미지를 제외한 항목들은 필수 입력하며,  
게임코드와 출시일은 유효성 검사를 한다.

localhost:8081 내용:

[게임 코드]

게임 코드는 G로 시작하고 숫자만을 더하여 4자리를 입력해야 합니다.

확인

localhost:8081 내용:

[출시일]

YYYY/MM/dd 형식으로 입력하세요.

확인

- 이미 존재하는 게임코드 입력 시.

localhost:8081 내용:

이미 존재하는 키입니다

확인

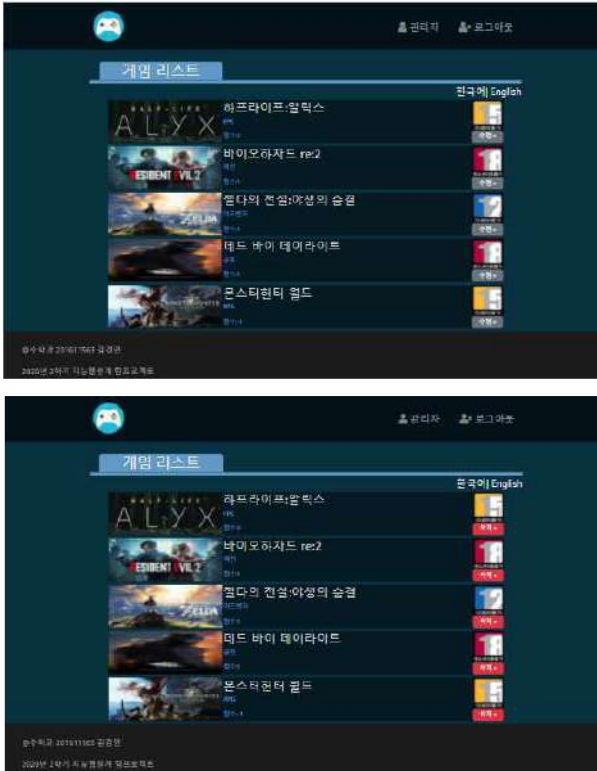
추가에 실패한다.

- 추가 성공 시, DB에 추가

-게임 수정 시, 코드는 수정 불가

### 7.3 게임 수정/삭제 목록(editGame.jsp)

### 설명



- 'post'로 'edit'값을 받아옴

- 다국어 처리

- 'edit'값이 'update'이면 게임 목록들에 수정 버튼이 생김

(editGame.jsp?edit=delete)

- 'edit'값이 'delete'인 경우 게임 목록들에 삭제 버튼이 생김

- 삭제 버튼 누를 시, 알림창이 뜨고, '예'를 누르면 삭제

### 8.예외처리 페이지

(exceptionNoGameId.jsp)/  
(exceptionNoCommId.jsp)

### 설명



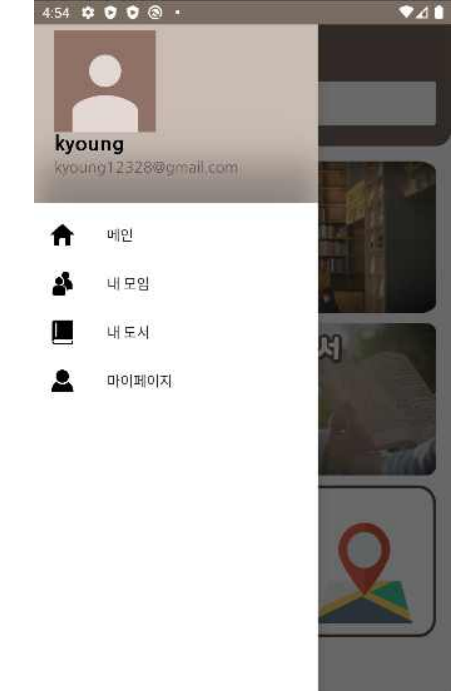


- 존재하지 않는 게임ID/게시글ID의 정보를 불러오면 해당 화면이 출력

# 기업 연계 팀 프로젝트 — Android 기반 도서 공유 플랫폼 애플리케이션

개발 기간	2022.03.09.-2022.06.15
개발 환경	Android Studio / Java, Xml,
개발 인원	6명
요 약	지역별 그룹을 형성하여 사람들과 책을 공유할 수 있는 애플리케이션 개발
기능	<ol style="list-style-type: none"> <li><b>1. 사용자 로그인</b> <ul style="list-style-type: none"> <li>- 사용자 인증(Google OAuth)</li> </ul> </li> <li><b>2. 도서 관리 서비스</b> <ul style="list-style-type: none"> <li>- 개인 또는 그룹의 도서 등록 및 관리 지원</li> <li>- 도서의 대여, 반납 관리 및 그에 따른 이력 조회 기능</li> <li>- 도서 공유 정책 설정 기능</li> </ul> </li> <li><b>3. 도서 공유 서비스</b> <ul style="list-style-type: none"> <li>- 사용자의 위치를 기반으로 주변에서 공유 중인 도서를 확인할 수 있는 기능(Google Map)</li> <li>- 그룹 생성 기능 및 해당 그룹원 간 도서 공유 서비스 지원</li> <li>- 도서의 대여 및 반납</li> <li>- 정해진 기준에 따른 도서 정보 조회</li> </ul> </li> <li><b>4. 그룹 관리 서비스</b> <ul style="list-style-type: none"> <li>- 그룹 관리자의 그룹원 관리자 임명 혹은 강퇴 기능</li> </ul> </li> </ol>
링크	<b>Front-end:</b> <a href="https://github.com/cozym/Capstone_FrontEnd/tree/integration">https://github.com/cozym/Capstone_FrontEnd/tree/integration</a> <b>Back-end:</b> <a href="https://github.com/CptBluebear/CapstoneLibrary">https://github.com/CptBluebear/CapstoneLibrary</a> <b>Notion:</b> <a href="https://www.notion.so/46cd51a662ac40c894db3410893a3ea9?v=b7e17a0abef14b07be3c775777c608b0">https://www.notion.so/46cd51a662ac40c894db3410893a3ea9?v=b7e17a0abef14b07be3c775777c608b0</a>

## 실행화면

로그인 화면	메인 화면	슬라이드 메뉴
 <p>21세기 중고도서 공유 어플</p> <p>공 책</p> <p>Sign in</p> <p>구글 계정으로 바로 로그인하세요.</p>	 <p>공 책</p> <p>검색어를 입력해주세요.</p> <p>전체 그룹 &amp; 도서</p> <p>내 그룹</p> <p>내 도서</p> <p>GPS</p> <p>현재 위치를 표시합니다.</p>	 <p>kyoung</p> <p>kyoung12328@gmail.com</p> <p>메인</p> <p>내 모임</p> <p>내 도서</p> <p>마이페이지</p>

## 메인 리스트 - 그룹

그룹	도서
 경기대 회원 수: 1 도서 수: 1	
 졸업만 모여요 회원 수: 4 도서 수: 1	
 서울 책방 회원 수: 3 도서 수: 3	
 영통구 책방 회원 수: 3 도서 수: 2	
 공대 전공도서 회원 수: 5 도서 수: 1	
 대학생 회원 수: 5	

## 메인 리스트 - 그룹(우측 하단 플로팅 버튼)

그룹	도서
 경기대 회원 수: 1 도서 수: 1	
 졸업만 모여요 회원 수: 4 도서 수: 1	
 서울 책방 회원 수: 3 도서 수: 3	
 영통구 책방 회원 수: 3 도서 수: 2	
 공대 전공도서 회원 수: 5 도서 수: 1	
 대학생 회원 수: 5	

## 메인 리스트 - 도서

그룹	도서
 쉽게 배우는 JSP 웹 프로그래밍 저자: 송미경 카테고리: 컴퓨터과학 대여가능	 네트워크 해킹과 보안 저자: 양대인 카테고리: 컴퓨터과학 대여불가

## 메인 리스트 - 도서2

그룹	도서
 C언어로 쉽게 풀어쓴 자료구조 저자: 천인국 카테고리: 컴퓨터과학 대여가능	 컴퓨터구조론 저자: 송미영 카테고리: 컴퓨터과학 대여불가

## 그룹 정보



**경기대**  
비공개  
경기대 학생들 들어오세요.

접속하기 닫힘



**경기대**  
비공개  
경기대 학생들 들어오세요.

그룹 정보

관리자	runak
도서 수	1
회원 수	1
생성일자	2022-04-11

접속하기 닫힘



**서울 책방**  
공개  
서울 사람들 모여라

접속하기 관리하기





**서울 책방**  
공개  
서울 사람들 모여라

접속하기



내 그룹	그룹 메인	그룹 메인(우측 하단 플로팅 버튼)																		
<div>9:32</div> <div>≡ 공책</div> <div>내 그룹</div> <table> <tr> <td></td><td>경기대</td><td>회원 수: 1</td></tr> <tr> <td></td><td>졸업만 모여요</td><td>회원 수: 4</td></tr> <tr> <td></td><td>서울 책방</td><td>회원 수: 3</td></tr> <tr> <td></td><td>영동구 책방</td><td>회원 수: 3</td></tr> <tr> <td></td><td>공대 전공도서</td><td>회원 수: 5</td></tr> <tr> <td></td><td>대학생</td><td>회원 수: 5</td></tr> </table>		경기대	회원 수: 1		졸업만 모여요	회원 수: 4		서울 책방	회원 수: 3		영동구 책방	회원 수: 3		공대 전공도서	회원 수: 5		대학생	회원 수: 5	<div>9:23</div> <div>≡ 공대 전공도서</div> <div>도서 사용자</div> <div>   </div> <div> <div>쉽게 배우는 JSP 웹 프로그래밍</div> <div>네트워크 해킹과 보안 개정3판</div> </div> <div> <div>저자: 송미영</div> <div>저자: 양대인</div> </div> <div> <div>카테고리: 컴퓨터과학</div> <div>카테고리: 컴퓨터과학</div> </div> <div> <div>대여가능</div> <div>대여불가</div> </div>	<div>9:24</div> <div>≡ 공대 전공도서</div> <div>도서 사용자</div> <div>   </div> <div> <div>쉽게 배우는 JSP 웹 프로그래밍</div> <div>네트워크 해킹과 보안 개정3판</div> </div> <div> <div>저자: 송미영</div> <div>저자: 양대인</div> </div> <div> <div>카테고리: 컴퓨터과학</div> <div>카테고리: 컴퓨터과학</div> </div> <div> <div>대여가능</div> <div>대여불가</div> </div>
	경기대	회원 수: 1																		
	졸업만 모여요	회원 수: 4																		
	서울 책방	회원 수: 3																		
	영동구 책방	회원 수: 3																		
	공대 전공도서	회원 수: 5																		
	대학생	회원 수: 5																		



도서 정보 (타인 도서)	도서 정보 (사용자 소유 도서)																								
<div>9:25</div> <div>← 공책</div> <div></div> <div>쉽게 배우는 JSP 웹 프로그래밍</div> <div>송미영</div> <div>대여가능</div> <div>JSP입문자를 위한 도서</div> <div>책 정보</div> <table> <tr><td>소유 그룹</td><td>공대 전공도서</td></tr> <tr><td>주인</td><td>송미영</td></tr> <tr><td>ISBN</td><td>979-11-5664-338-8</td></tr> <tr><td>카테고리</td><td>컴퓨터과학</td></tr> <tr><td>출판사</td><td>한빛아카데미</td></tr> <tr><td>등록일</td><td>2022-04-08</td></tr> </table> <div>대여하기</div>	소유 그룹	공대 전공도서	주인	송미영	ISBN	979-11-5664-338-8	카테고리	컴퓨터과학	출판사	한빛아카데미	등록일	2022-04-08	<div>9:25</div> <div>← 공책</div> <div></div> <div>컴퓨터구조론</div> <div>송미영</div> <div>대여불가</div> <div>컴퓨터구조 공부를 위한 도서</div> <div>책 정보</div> <table> <tr><td>소유 그룹</td><td>공대 전공도서</td></tr> <tr><td>주인</td><td>송미영</td></tr> <tr><td>ISBN</td><td>978-89-7050-969-3</td></tr> <tr><td>카테고리</td><td>컴퓨터과학</td></tr> <tr><td>출판사</td><td>생능출판</td></tr> <tr><td>등록일</td><td>2022-05-22</td></tr> </table> <div>빌려주기 삭제</div>	소유 그룹	공대 전공도서	주인	송미영	ISBN	978-89-7050-969-3	카테고리	컴퓨터과학	출판사	생능출판	등록일	2022-05-22
소유 그룹	공대 전공도서																								
주인	송미영																								
ISBN	979-11-5664-338-8																								
카테고리	컴퓨터과학																								
출판사	한빛아카데미																								
등록일	2022-04-08																								
소유 그룹	공대 전공도서																								
주인	송미영																								
ISBN	978-89-7050-969-3																								
카테고리	컴퓨터과학																								
출판사	생능출판																								
등록일	2022-05-22																								
<div>9:44</div> <div>← 공책</div> <div></div> <div>C언어 콘서트</div> <div>누구나 쉽게 즐기는 C언어 콘서트</div> <div>친인국</div> <div>대여가능</div> <div>C언어 쉽게 배우보라</div> <div>대여하기 삭제</div>																									


그룹 검색

<div>9:29</div> <div>← 검색어를 입력해주세요.</div> <div>전체 이음순</div>	
그룹	도서
	경기대 회원 수: 1 도서 수: 1
	졸업만 모여요 회원 수: 4 도서 수: 1
	서울 책방 회원 수: 3 도서 수: 3
	영동구 책방 회원 수: 3 도서 수: 2
	공대 전공도서 회원 수: 5 도서 수: 1

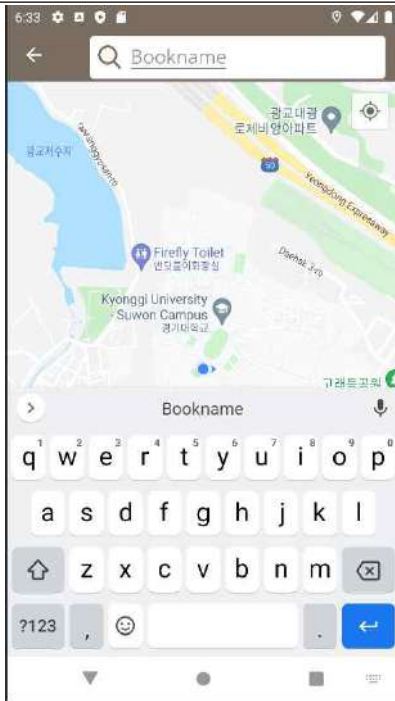
도서 검색

<div>9:30</div> <div>← 검색어를 입력해주세요.</div> <div>전체 이음순</div>	
그룹	도서
	경기대 회원 수: 1 도서 수: 1
	졸업만 모여요 회원 수: 4 도서 수: 1
	서울 책방 회원 수: 3 도서 수: 3
	영동구 책방 회원 수: 3 도서 수: 2
	공대 전공도서 회원 수: 5 도서 수: 1

<div>9:31</div> <div>← 검색어를 입력해주세요.</div> <div>전체 이음순</div>	
그룹	도서
	
쉽게 배우는 JSP 웹 프로그래밍	네트워크 해킹과 보안 개정3판
저자: 송미영	저자: 양대인
카테고리: 컴퓨터과학	카테고리: 컴퓨터과학
대여가능	대여불가
	

<div>9:31</div> <div>← 검색어를 입력해주세요.</div> <div>전체 이음순</div>	
그룹	도서
	
쉽게 배우는 JSP 웹 프로그래밍	네트워크 해킹과 보안 개정3판
저자: 송미영	저자: 양대인
카테고리: 컴퓨터과학	카테고리: 컴퓨터과학
대여가능	대여불가
	

## GPS



## 마이페이지



## 1. 개발 기획

### 1.1 개발 필요성

60개국 3만 명 이상의 소비자들을 대상으로 진행한 설문조사 결과 68%의 소비자들은 소유하고 있는 자원을 공유 또는 대여할 용의가 있다고 응답하였다. 이러한 공유경제는 숙박, 자동차, 여행 등 다방면으로 발전하고 있다. 그중에서도 도서를 공유하는 도서관은 사용자들 간에 물건과 지식이 반복적으로 공유되며 그 가치와 효율성은 무한하다고 볼 수 있다. 누군가에게 쓰임이 다한 도서를 다른 누군가에게 다시 제공하는 선순환 구조를 큰 규모의 도서관 뿐만 아니라 소규모 그룹과 개인으로부터 형성한다면 긍정적인 효과를 기대할 수 있다. 본 프로젝트는 동아리나 학회와 같은 소규모 그룹 혹은 가정에서의 도서를 관리하는 것이 용이하지 못하다는 점에서 착안하여 도서 관리 시스템을 모바일 어플리케이션의 형태로 개발하기로 했다.



## 1.2 목표

본 프로젝트는 개인 혹은 소규모 그룹에서의 도서를 보다 편리하게 등록, 관리, 추적 및 공유하는 것을 목표로 하고 있으며, 이미 존재하는 서비스들은 서버 설치와 같은 구축 난도에 대한 부담이 있을뿐더러, 초기 도입 비용이 발생한다는 문제점을 해결하기 위하여, 기본적으로 서비스를 무료로 제공하며, 본 팀에서 서버를 제공함에 의의를 두고 있다. 또한, 쉽게 도서를 검색할 수 있으며, 그룹 내 회원 및 도서 관리를 보다 편리하게 하는 것 역시 목표로 삼고 있다. 도서 공유를 원하는 개인 또는 그룹장이 그룹을 생성한다. 그룹 가입은 그룹장을 통해 가능하며, 그룹에 속해있다면 공유하고자 하는 도서를 그룹에 등록할 수 있다. 도서 대여를 원하는 경우 검색 엔진을 통해 주변의 검색한 도서를 보유하고 있는 그룹을 파악하여 가입한다. 그리고 가입한 그룹의 위치로 이동하여 그룹이 보유하고 있는 도서를 대여할 수 있다. 도서 대여는 ISBN 기반의 고유 코드 센싱을 통해 진행된다. 도서 반납의 경우 도서 대여자가 아닌 공유한 도서의 소유자가 반납 상태를 확인하고 처리한다. 또한, 도서 공유 로그를 통한 도서 추적 기능과 그룹장에 의한 그룹 관리 기능까지 탑재하는 것을 목표로 한다.

## 1.3 요구사항 :

- 사용자는 각자의 구글 이메일을 활용하여 본 서비스를 이용할 수 있다.
- 사용자는 특정한 그룹에 속해있을 수 있으며, 그룹 내에서 도서를 대여하거나 공유할 수 있다.
- 그룹의 구성원에는 그룹장과 그룹원이 있으며, 그룹장은 그룹원에게 그룹장을 위임할 수 있고, 그룹원을 강퇴시킬 수 있다.
- 사용자는 그룹을 새로 생성할 수 있다. 이때, 사용자의 위치를 기반으로 한다.
- 그룹 생성 시 썸네일, 그룹명, 그룹 설명, 그룹 공개여부를 지정한다.
- 그룹장은 그룹 생성 시 상세정보에 카카오톡 오픈채팅 링크를 명시한다. 그룹 내 사용자들은 해당 오픈채팅을 사용하여 대여하려는 도서와 관련된 대화 혹은 담소를 나눌 수 있다.
- 그룹은 그룹명, 관리자, 조회수, 그룹 생성일시를 저장한다.
- 도서는 소속된 그룹, 조회수, 도서 등록일시를 저장한다.
- 사용자는 가입한 그룹에 접속할 수 있다. 그룹에 접속 시 그룹이 보유 중인 도서들과 그룹원들 목록을 확인할 수 있다.
- 사용자는 현재 위치를 기반으로 근방에 있는 그룹들을 검색할 수 있다.
- 그룹은 공개 혹은 비공개로 설정이 가능하다.
- 사용자는 자신이 속해있는 그룹 혹은 자신이 보유 중인 도서를 조회할 수 있다.
- 사용자는 처음 로그인 시 임의의 닉네임을 부여받는다. 해당 닉네임은 마이페이지에서 변경이 가능하다.
- 사용자는 마이페이지에서 자신의 정보(이름, 닉네임, 이메일, 도서 수, 대여 수)와 도서 대여 기록을 확인할 수 있다.
- 사용자는 그룹 내 다른 사용자에게서 도서를 대출할 수 있다. 이때, 대출 및 반납 처리는 도서의 주인이 한다.
- 사용자는 소속된 그룹에 도서를 등록할 수 있다. 이때 필요한 정보로 도서의 썸네일, 도서명, 도서 설명, 그리고 해당 도서의 ISBN이 있다.
- 사용자는 도서를 등록할 때 카메라를 이용하여 ISBN 코드를 스캔하여 해당 도서에 대한 정보를 자동으로 불러오게 하거나 수동으로 도서에 대한 정보들을 입력할 수 있다.
- 사용자는 도서나 그룹을 검색할 수 있다.
- 사용자는 도서를 검색할 때 카테고리별로 분류해서 검색할 수 있다.
- 사용자는 도서를 조회할 때 카테고리별로 분류해서 조회할 수 있다.
- 사용자는 그룹을 조회할 때 이름순 또는 날짜순으로 정렬할 수 있다.

## 2. 관련 기술 현황

### - 구글맵 API

API는 정의 및 프로토콜 집합을 사용하여 두 소프트웨어 구성 요소가 서로 통신할 수 있게 하는 메커니즘이다. 세계적인 기업인 구글에서 지원하는 구글맵 API를 활용하여 사용자의 위도와 경도를 기반으로 근방의 지도를 확인할 수 있다.

### - ISBN

ISBN은 국제적으로 책에 붙이는 고유한 식별자이다. 한 책의 모든 판본 종류의 하나의 ISBN이 부여되며, 10자리 혹은 13자리의 10진수로 이루어진 코드이다. 대한민국의 경우 ISBN 13자리를 표기한 후 추가적으로 5자리 부가번호를 병기함으로써 이를 한국표준도서번호로 사용하고 있다.

### - Okhttp

Okhttp는 REST API, http 통신을 간편하게 구현할 수 있도록 다양한 기능을 제공해주는 Java 오픈소스 라이브러리이다. Okhttp는 최신 http 통신에 대해 간편히 하고 데이터와 미디어를 교환하고 구성하는데 도움을 주며, 동기화 및 비동기 방식을 각각 제공한다.

### - Retrofit

Retrofit은 Okhttp를 기반으로 하는 서버와 통신을 해서 서버로부터 받은 데이터를 앱으로 출력해 사용자가 확인할 수 있게 하는 라이브러리이다. Retrofit은 REST 기반의 웹 서비스를 통해 JSON 구조의 데이터를 쉽게 가져오고 업로드 할 수 있다.

### - JSON

JSON은 Javascript 객체 문법으로 구조화된 데이터를 표현하기 위한 문자 기반의 표준 포맷이다. 웹 어플리케이션에서 데이터를 전송할 때 일반적으로 사용한다. JSON은 문자열 형태로 존재하기 때문에 네트워크를 통해 통신할 경우 상당히 유용하다는 이점을 갖고 있다.

### - GSON

GSON은 구글에서 만든 JSON 구조를 띄는 직렬화된 데이터를 Java의 객체로 역직렬화 또는 직렬화 해주는 Java 라이브러리이다. GSON은 내부적으로 Reflection을 사용하기 때문에 사용자가 추가적인 옵션을 첨가하지 않아도 되어 편리하다는 이점을 갖고 있다.

### - OAuth

OAuth는 인터넷 사용자들이 비밀번호를 제공하지 않고 다른 웹사이트 상의 자신들의 정보에 대해 웹사이트나 어플리케이션의 접근 권한을 부여할 수 있는 공통적인 수단으로서 사용되는, 접근 위임을 위한 개방형 표준이다.

### - ORM

ORM은 객체와 관계형 데이터베이스의 데이터를 자동으로 연결해주는 프레임워크다. ORM을 사용함으로써 재사용 및 유지보수의 편리성이 증가하고, DBMS에 대한 종속성이 줄어든다는 이점을 갖고 있다.

### - Spring Boot

Spring Boot는 Spring을 더 쉽게 사용하기 위한 도구로, 라이브러리 관리 자동화, 라이브러리 버전 자동관리, 설정 자동화, Tomcat 내장, 독립적으로 실행 가능한 JAR 파일, 코드 생성 및 XML 구성 요구사항 없음 등의 이점을 갖고 있다.

### - QueryDSL

QueryDSL은 HQL Query를 타입에 안전하게 생성 및 관리할 수 있게 해주는 프레임워크이다. QueryDSL은 Query를 Java 코드로 작성할 수 있게 도와주고, Spring Data로 해결하지 못하는 복잡한 Query나 동적 Query를 해결할 수 있으며, Java 코드로 작성하기 때문에 문법적인 오류를 컴파일 시점에서 확인할 수 있다는 이점이 있다.

### - Swagger

Swagger란 REST API 서비스를 설계, 빌드, 문서화 할 수 있도록 하는 프로젝트이다. Swagger는 Back-end의 API를 호출하는 Front-end 프로그램을 제작하는 경우 사용하고, 명세를 관리하는 목적이 있으며, API가 수정되더라도 문서가 자동으로 갱신된다는 이점이 있다.

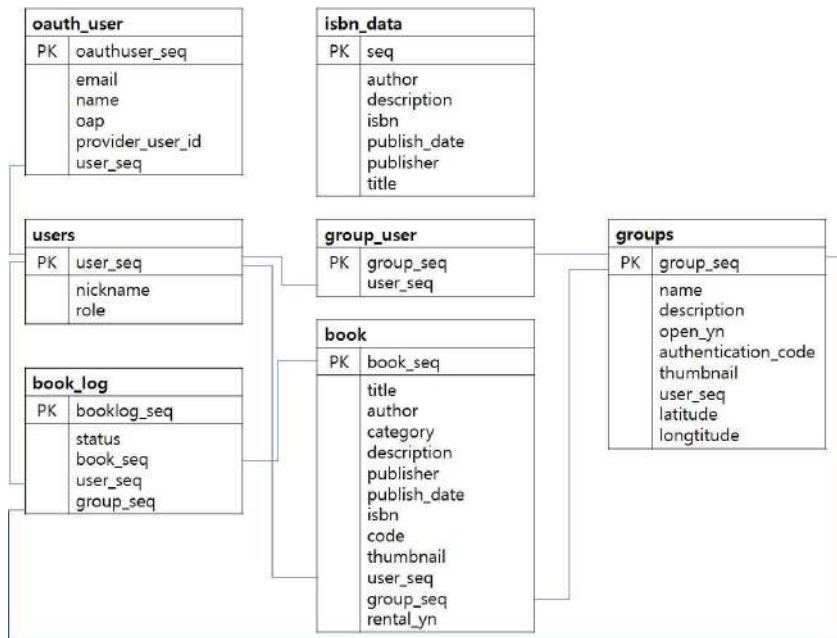
### 3. 주요 접근 방법

Spring Boot로 개발된 HTTP API를 통해 Database와 ORM으로 통신하고, 안드로이드 어플리케이션과 JSON으로 통신한다.

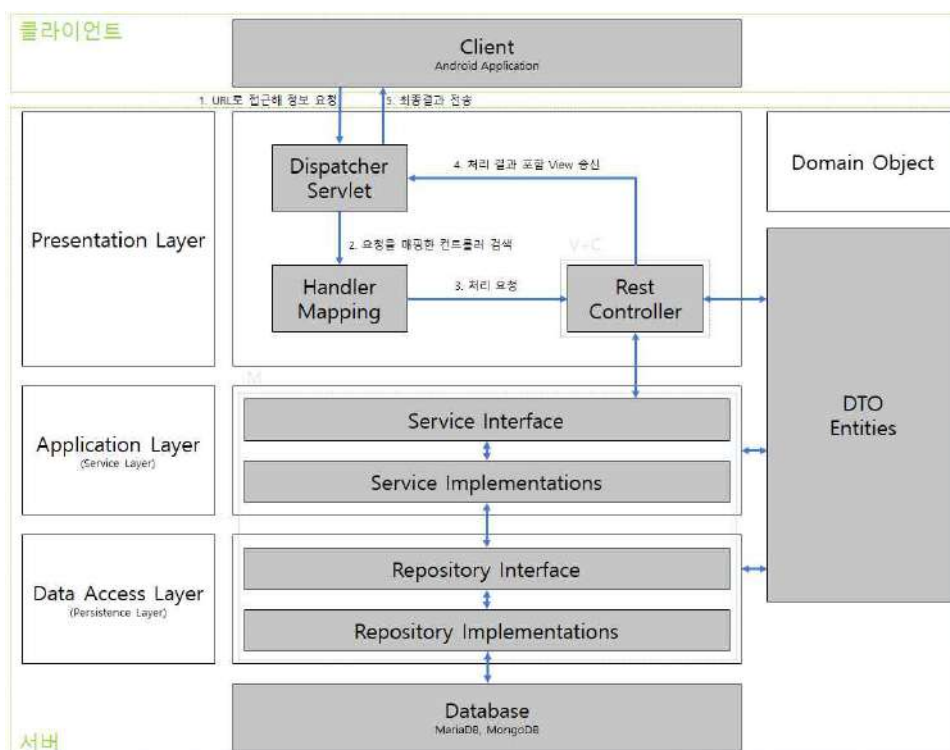
사용 기술로는 HTTP API를 구현하기 위해 사용한 JAVA 기반 프레임워크인 스프링부트, 사용자 편의성과 보안을 위해 인증을 위임하는 OAuth, Stateless 방식으로 토큰을 통해 세션 없이 클라이언트를 인증하는 JWT, 데이터베이스를 객체지향적으로 매핑하는 ORM, 위치 기반 검색을 지원하는 Google Map API, ISBN 코드를 통해 도서 정보를 추출하는 ISBN API, 지역 정보를 저장할 수 있는 NoSQL, 그리고 관계형 데이터베이스인 MariaDB가 있다.

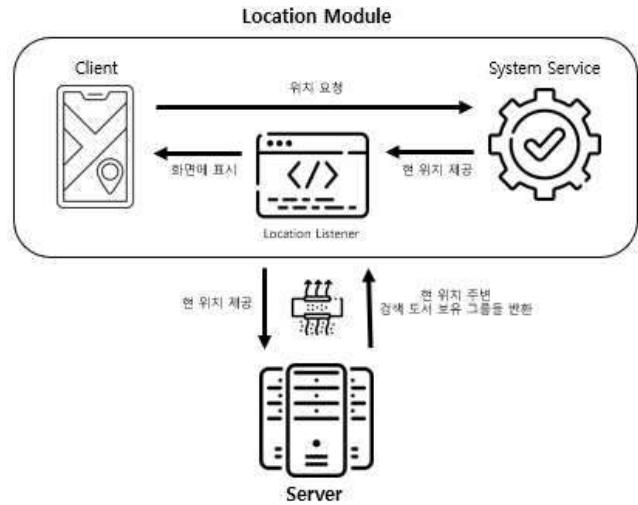
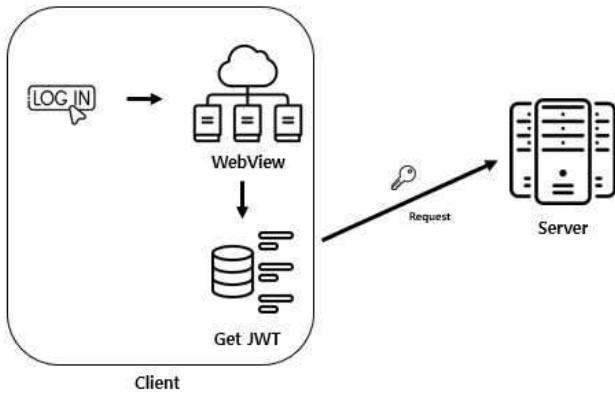
### 2. 설계

#### -----ERD



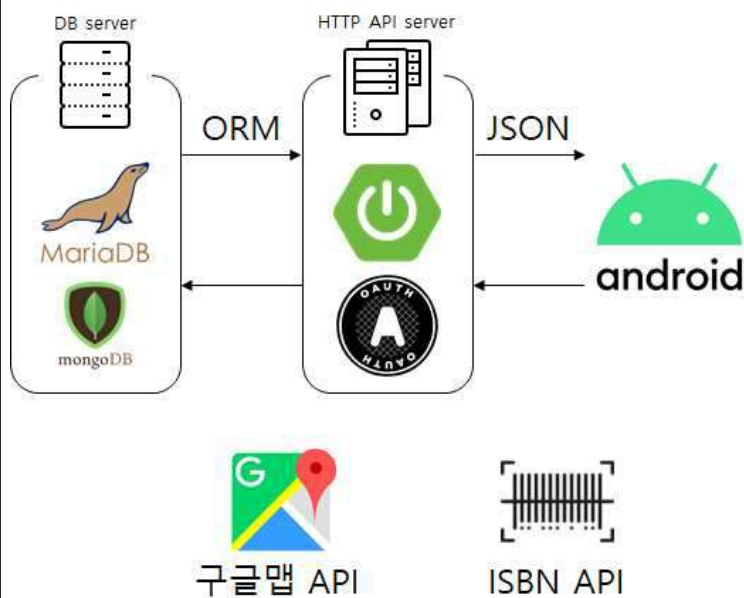
#### -----Architecture





- 로그인 실행 시 WebView를 통해 서버의 JWT를 받아온다.
- 해당 JSON 데이터를 캐시 데이터 형태로 저장한다.
- 서버에 데이터를 Request할 때 헤더에 포함하여 요청한다.

- 클라이언트가 시스템에 현재 위치를 요청하면 서비스는 Location Listener에게 현 위치를 제공한다.
- Location Listener가 서버에 받아온 현 위치를 제공하면 서버는 Location Listener에게 현 위치 주변에 있는 검색한 도서를 소유하고 있는 그룹을 반환한다.
- Location Listener는 반환받은 그룹들의 위치를 클라이언트의 화면에 표현한다.



Spring Boot 프레임워크를 이용하여 구성하였고, Presentation layer, Service layer, Persistence layer로 나타낼 수 있다. Google OAuth를 이용하여 인증을 처리하였고, RestController를 활용하여 서비스를 구현하였다. DB를 연동하기 위해 Hibernate와 QueryDSL 라이브러리를 사용했다. DBMS로는 MongoDB와 MariaDB를 사용하였다. Front-end와 Back-end 간의 API Endpoint를 문서화하는데 Swagger를 이용하여 API Document를 작성했다. Database 서버에서 ORM을 통하여 HTTP API 서버와 통신하고 HTTP API Server에서 데이터들이 JSON 파일을 통해 직렬화되면 해당 파일을 android studio에서 받아 통신하는 형태로 구현하였다.

맡은 역할	1. Front-End 2. DB연동 3. UI 설계 및 디자인
프로젝트를 진행하면서 어려웠던 점	<p>1. Android가 기본적으로 '<u>동기방식</u>'을 택하기에 그룹/도서 목록을 먼저 띄운 뒤에 DB에서 리스트 정보를 받아서 리스트가 뒤늦게 받아지는 현상이 발생했다. 이를 해결하기 위해 동기/비동기 방식의 차이를 확실히 인지하고 이용해야만 했다. 또한, 속도의 문제를 겪으며, 이를 위해 '<u>쓰레드</u>'의 개념을 익혀 멀티쓰레드를 이용해야했습니다.</p> <p>2. API를 팀이 원하는 방식으로 이용하기 위해 커스텀을 하는 과정(특히, Google Login API, Google MAP API)에서 어려움을 겪었다. 하지만, 하나하나 코드를 만져가며 구동방식을 이해하고 해결해나갔다.</p>
깨달은 점	<p>1. 팀원들 대부분이 이러한 프로젝트 경험이 없기에 개발 초기에 난항을 겪었다. 하지만, 팀원들끼리 각자 알고 있는 부분을 서로 설명해주고 공유하여 각자에게 큰 도움이 되어주었고, 서로가 서로에게 시너지 효과가 발동되어 긍정적으로 프로젝트를 성공할 수 있었다. 이 점을 통해, 실무에서도 팀원들끼리 서로 협력을 통해 난관을 헤쳐나갈 수 있을 것이라고 느꼈다</p> <p>2. 배워본 적 없는 Android 개발을 대형 프로젝트로 시작하여 초기에 큰 어려움을 겪었다. 하지만 포기하지 않고 끈기를 가지며, 어려운 부분을 배워나가고 다신 잊지 않고 나아간 결과, 팀원들과 함께 훌륭히 프로젝트를 완수할 수 있었다. 이 점에서 제 아무리 어려운 과제여도 끈기를 가지고 도전한다면, 그 고난은 큰 성장의 발판이 될 수 있음을 느꼈다.</p>