



# 포팅 매뉴얼

 작성일 2023.11.16

 작성자 신준호

 제목 희소식 포팅 매뉴얼



개발 환경

## 형상 관리

- Gitlab

## 이슈 관리

- Jira

## Communication

- Mattermost
- Webex
- Notion

## OS

- Windows 10

## UI/UX

- Figma

## IDE

- IntelliJ IDEA
- VSCode
- Android Studio

## Database

- MySQL 8.0.33
- Redis 7.0.12
- MongoDB

## BackEnd

- JAVA JDK 11
- SpringBoot 2.7.17
  - Gradle 8.3
  - Spring Data JPA
  - Lombok
  - Springdoc Swagger

## FrontEnd

- Node 18.16.1
- NPM 9.6.7
- Typescript + TailWindCss
- React 18.2.0
  - Recoil
  - React-query

- Spring Security
- JWT

- Framer-motion
- ChakraUI
- Axios

## APP

- 안드로이드스튜디오 giraffe
- CompileSDK 34
- TargetSDK 33
- MinSDK 30
- Java 8
- Realm

- 설치
  - npm install react-router-dom
  - npm install --save-dev @types/react-router-dom
  - npm install --save-dev @types/react

## 기타 편의 툴

- Postman 10.15.8

## Infra

- AWS EC2
  - Ubuntu 20.04
  - Docker 24
- Nginx 1.25.1
- Jenkins



### EC2 사용 포트

포트	설명
22	SSH
80	HTTP
443	HTTPS
3000	React

5012	MySQL
27017	MongoDB
8761	Spring Eureka
8001	Spring Gateway
64000	member-service
65000	map-service
9090	Jenkins
6379	Redis



## Nginx 설정

/etc/nginx/sites-available/default 파일에 도메인 설정

```
server {
    listen 80;
    server_name saveurlife.kr, www.saveurlife.kr;
    ...
}
```

Nginx 리로드하여 default파일 설정 적용

```
# default 파일이 정상적인지 테스트
$ sudo nginx -t

# 리로드
$ sudo service nginx reload
```

SSL 인증서 발급(Certbot으로 Nginx에 SSL 인증서 자동 적용)

```
# Let's Encrypt 설치
$ sudo apt-get install letsencrypt

# Certbot 설치
sudo apt-get install certbot python3-certbot-nginx
```

```
# Certbot 동작
sudo certbot --nginx
```

## SSL 인증서 적용 후 최종적으로 설정한 /etc/nginx/sites-available/default 파일

```
server {

    server_name saveurlife.kr www.saveurlife.kr;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        # try_files $uri $uri/ =404;
        proxy_pass http://localhost:3000;
    }

    location /api {
        proxy_pass http://saveurlife.kr:8001;
    }
    location /images {
        alias /home/ubuntu/upload;
        try_files $uri $uri/ =404;
    }

    # pass PHP scripts to FastCGI server
    #
    #location ~ \.php$ {
    #    include snippets/fastcgi-php.conf;
    #
    #    # With php-fpm (or other unix sockets):
    #    fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
    #    # With php-cgi (or other tcp sockets):
    #    fastcgi_pass 127.0.0.1:9000;
    #}

    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    #location ~ /\.ht {
    #    deny all;
    #}

    listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/saveurlife.kr/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/saveurlife.kr/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
```

```

}

server {
    if ($host = www.saveurlife.kr) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    if ($host = saveurlife.kr) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    listen 80 default_server;
    listen [::]:80 default_server;

    server_name saveurlife.kr www.saveurlife.kr;
    return 404; # managed by Certbot

}

```



## Docker 컨테이너

### Jenkins

```

# jenkins 이미지 pull
$ sudo docker pull jenkins/jenkins:jdk11

# jenkins 컨테이너 실행
$ sudo docker run -d -p 9090:9090 -v /home/ubuntu/jenkins:/var/jenkins_home --name jenkins -u root jenkins/jenkins:jdk11

```

- EC2 서버의 /home/ubuntu/jenkins와 docker 컨테이너의 /var/jenkins\_home를 볼륨으로 연결하여 workspace에 생성되는 프로젝트 빌드 파일을 관리

### 사용된 주요 플러그인

- Gradle Plugin 2.8.2

- NodeJS Plugin 1.6.1
- SSH Agent Plugin 333.v878b\_53c89511
- GitLab Plugin 1.7.16

## Item

- Item1: goodnews-discovery
  - Pipeline script

```

pipeline {
    agent any

    stages {
        stage('Git clone') {
            steps {
                git branch: 'master', credentialsId: 'goodnews', url: 'http
s://lab.ssafy.com/s09-final/S09P31B307.git'
            }
        }
        stage('Build') {
            steps {
                dir("./BE/discovery") {
                    sh "chmod +x ./gradlew"
                    sh "./gradlew clean build"
                }
            }
        }
        stage('Deployment') {
            steps {
                sshagent(credentials: ['ssh_key']) {
                    sh '''
                        ssh -o StrictHostKeyChecking=no ubuntu@k9b307.p.ssafy.
io
                        scp /var/jenkins_home/workspace/goodnews-BE/BE/goodnew
s-discovery/build/libs/discovery-0.0.1-SNAPSHOT.jar ubuntu@k9b307.p.ssafy.io:/
home/ubuntu/goodnews-discovery
                        ssh -t ubuntu@k9b307.p.ssafy.io ./deploy_be.sh
                        ...
                    '''
                }
                timeout(time: 30, unit: 'SECONDS') {
                }
            }
        }
    }
}

```

- Item2: goodnews-gateway

- Pipeline script

```
pipeline {
    agent any

    stages {
        stage('Git clone') {
            steps {
                git branch: 'master', credentialsId: 'goodnews', url: 'http
s://lab.ssafy.com/s09-final/S09P31B307.git'
            }
        }
        stage('Build') {
            steps {
                dir("./BE/gateway") {
                    sh "chmod +x ./gradlew"
                    sh "./gradlew clean build"
                }
            }
        }
        stage('Deployment') {
            steps {
                sshagent(credentials: ['ssh_key']) {
                    sh '''
                        ssh -o StrictHostKeyChecking=no ubuntu@k9b307.p.ssafy.
io
                        scp /var/jenkins_home/workspace/goodnews-BE/BE/goodnew
s-gateway/build/libs/gateway-0.0.1-SNAPSHOT.jar ubuntu@k9b307.p.ssafy.io:/hom
e/ubuntu/goodnews-gateway
                        ssh -t ubuntu@k9b307.p.ssafy.io ./deploy_be.sh
                        ...
                    '''
                }
                timeout(time: 30, unit: 'SECONDS') {
                }
            }
        }
    }
}
```

- Item3: goodnews-member

- Pipeline script

```
pipeline {
    agent any

    stages {
        stage('Git clone') {
            steps {
                git branch: 'master', credentialsId: 'goodnews', url: 'http
s://lab.ssafy.com/s09-final/S09P31B307.git'
            }
        }
    }
}
```

```

    }
    stage('Build') {
        steps {
            dir("./BE/member") {
                sh "chmod +x ./gradlew"
                sh "./gradlew clean build"
            }
        }
    }
    stage('Deployment') {
        steps {
            sshagent(credentials: ['ssh_key']) {
                sh '''
                    ssh -o StrictHostKeyChecking=no ubuntu@k9b307.p.ssafy.
io
                    scp /var/jenkins_home/workspace/goodnews-BE/BE/goodnew
s-member/build/libs/member-0.0.1-SNAPSHOT.jar ubuntu@k9b307.p.ssafy.io:/home/u
buntu/goodnews-member
                    ssh -t ubuntu@k9b307.p.ssafy.io ./deploy_be.sh
                    '''
            }
            timeout(time: 30, unit: 'SECONDS') {
            }
        }
    }
}
}
}
}
}

```

- Item4: goodnews-map
  - Pipeline script

```

pipeline {
    agent any

    stages {
        stage('Git clone') {
            steps {
                git branch: 'master', credentialsId: 'goodnews', url: 'http
s://lab.ssafy.com/s09-final/S09P31B307.git'
            }
        }
        stage('Build') {
            steps {
                dir("./BE/map") {
                    sh "chmod +x ./gradlew"
                    sh "./gradlew clean build"
                }
            }
        }
        stage('Deployment') {
            steps {
                sshagent(credentials: ['ssh_key']) {

```



```

        sh '''
            ssh -o StrictHostKeyChecking=no ubuntu@k9b307.p.ssafy.
io
            scp /var/jenkins_home/workspace/goodnews-BE/BE/goodnew
s-map/build/libs/map-0.0.1-SNAPSHOT.jar ubuntu@k9b307.p.ssafy.io:/home/ubuntu/
goodnews-map
            ssh -t ubuntu@k9b307.p.ssafy.io ./deploy_be.sh
        '''
    }
    timeout(time: 30, unit: 'SECONDS') {
    }
}
}
}
}
}

```

- Item5: goodnews-fe

- Pipeline script

```

pipeline {
    agent any

    stages {
        stage('Git clone') {
            steps {
                git branch: 'fe-web-dev', credentialsId: 'goodnews', url: 'htt
ps://lab.ssafy.com/s09-final/S09P31B307.git'
                echo "Current workspace: ${workspace}"
            }
        }

        stage('Deployment') {
            steps {
                sshagent(credentials: ['ssh_key']) {
                    sh '''
                        ssh -o StrictHostKeyChecking=no ubuntu@k9b307.p.ssafy.
io "rm -rf /home/ubuntu/goodnews-front/node_modules"
                        scp -r /var/jenkins_home/workspace/goodnews-FE/good-ne
ws/* ubuntu@k9b307.p.ssafy.io:/home/ubuntu/goodnews-front
                        ssh -t ubuntu@k9b307.p.ssafy.io ./deploy_fe.sh
                    '''
                }
            }
        }
    }
}

```

## Dockerfile & deploy.sh 생성

- Dockerfile

- BE(discovery, gateway, member, map)

```
FROM adoptopenjdk/openjdk11
COPY ./goodnews-0.0.1-SNAPSHOT.jar /goodnews-0.0.1-SNAPSHOT.jar
CMD ["java", "-jar", "goodnews-0.0.1-SNAPSHOT.jar"]
```

- FE

```
FROM node:18.16.1

#현재 컨테이너 디렉토리 루트 경로명이 frontend 이다.
WORKDIR /frontend

# goodnews-front 전체 복사 -> FE 프로젝트 전체
COPY . ./

# 이걸 docker-compose에서 포트 매핑 해줬으니까 안해도 됨
EXPOSE 3000

# 컨테이너 안에서 node_module 설치
RUN npm install --silent

# 컨테이너 내에서 서버 실행
CMD ["npm", "run", "start"]
```

- deploy.sh(소유자에게 파일 실행 권한 필요: -rwxrw-r—)

- BE(discovery, gateway, member, map)

```
cd goodnews-back
sudo docker compose down
sudo docker compose up -d --build
yes | sudo docker system prune
```

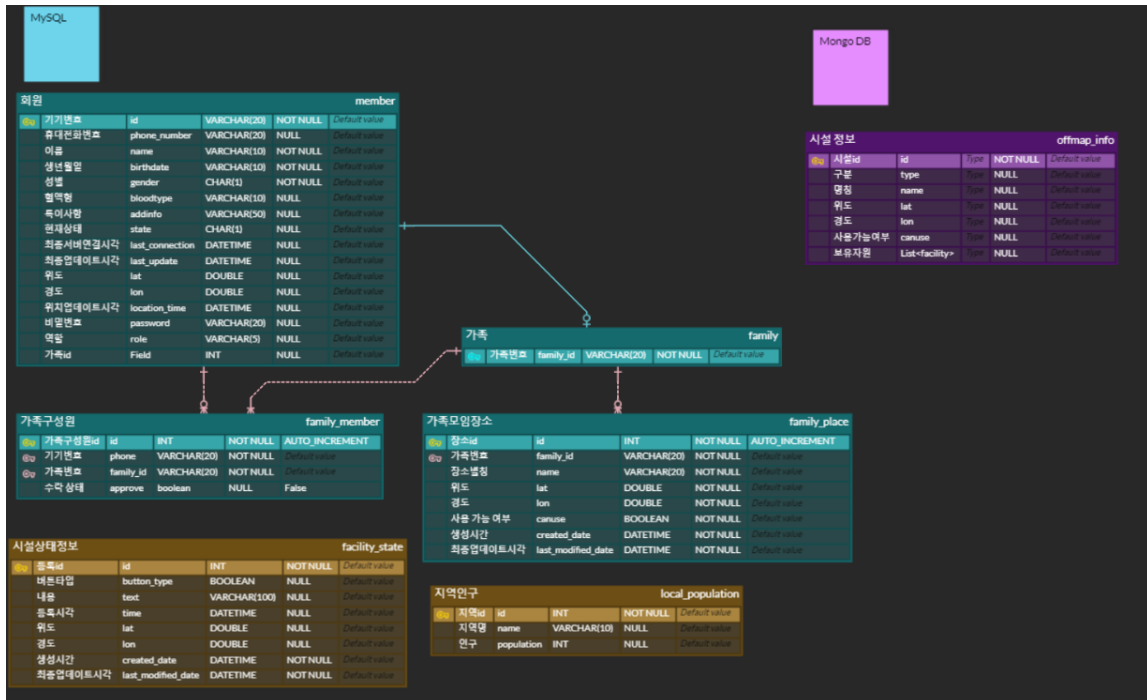
- FE

```
cd goodnews-front
sudo docker compose down
```

```
sudo docker compose up -d --build
yes | sudo docker system prune
```



## WEB ERD



## APP ERD

REALM			
회원			
키	값	타입	비고
memberId	memberId	Domain	String
휴대전화번호	phone	Domain	String
생년월일	birthDate	Domain	String
이름	name	Domain	String
성별	gender	Domain	String
혈액형	bloodType	Domain	String
특이사항	adInfo	Domain	String
현재상태	state	Domain	String
최근 서버 연결 시각	lastConnection	Domain	RealInstant
최근 업데이트 시각	lastUpdate	Domain	RealInstant
경도	latitude	Domain	Double
위도	longitude	Domain	Double
가족id	familyId	Domain	Int
가족모임장소			
키	값	타입	비고
장소id	placeId	Domain	Int
명칭	name	Domain	String
주소	address	Domain	String
경도	latitude	Domain	Double
위도	longitude	Domain	Double
사용 가능 여부	canUse	Domain	Boolean
순서	seq	Domain	Int
수신알림			
키	값	타입	비고
알림id	id	Domain	Int
내용	content	Domain	String
발행 시각	time	Domain	RealInstant
모스부호			
키	값	타입	비고
코드id	code	Domain	Int
텍스트	text	Domain	String
지도정보			
키	값	타입	비고
상태id	state	Domain	String
내용	content	Domain	String
등록 시각	time	Domain	RealInstant
위도	lat	Domain	Double
경도	lon	Domain	Double
오프라인 지도 정보			
키	값	타입	비고
지도정보id	id	Domain	Int
구분	type	Domain	String
명칭	name	Domain	String
위도	latitude	Domain	Double
경도	longitude	Domain	Double
사용 가능 여부	canUse	Domain	Boolean
추가 정보	adInfo	Domain	String
가족 구성원 정보			
키	값	타입	비고
구성원id	id	Domain	String
이름	name	Domain	String
휴대전화번호	phone	Domain	String
최근 서버 연결 시각	lastConnection	Domain	RealInstant
상태	state	Domain	String
경도	latitude	Domain	Double
위도	longitude	Domain	Double
가족id	familyId	Domain	Int
그룹 구성원 정보			
키	값	타입	비고
그룹id	group_id	Domain	Long
이름	name	Domain	String
사용자 이름	userName	Domain	String
최근 연결 시각	lastConnection	Domain	RealInstant
상태	state	Domain	String
경도	latitude	Domain	Double
위도	longitude	Domain	Double
Alert			
키	값	타입	비고
알림id	id	Domain	Int
내용	content	Domain	String
발행 시각	time	Domain	RealInstant
MorseCode			
키	값	타입	비고
코드id	code	Domain	Int
텍스트	text	Domain	String
Chat			
키	값	타입	비고
채팅id	id	Domain	String
상대 이름	name	Domain	String
메시지 리스트	messageList	Domain	RealmList<ChatMessage>
ChatMessage			
키	값	타입	비고
메시지id	id	Domain	String
채팅방id	chatRoomId	Domain	String
송신자	senderId	Domain	String
송신자 이름	senderName	Domain	String
내용	content	Domain	String
시간	time	Domain	String
AidKit			
키	값	타입	비고
장비id	name	Domain	String
자원상태	state	Domain	String



## MySQL 설정

### 운영 환경 계정

- database: goodnews
- host: k9b307.p.ssafy.io
- port: 5012
- username: goodnewsteam307
- password: gnteam307



## Redis 설정

- port: 6379
- password: redis



## MongoDB 설정

- port: 27017
- username: goodnewsteam307
- password: gnteam307



## Spring Boot properties 파일

### application.yml (discovery)

```
server:
  port: 8761
spring:
  application:
    name: discoveryservice

eureka:

  client:
    register-with-eureka: false
    fetch-registry: false
    service-url:
      defaultZone: http://saveurlife.kr:8761/eureka
```

### application.yml (gateway)

```
server:
  port: 8001
# ssl:
#   enabled: true
#   key-store-type: pkcs12
#   key-store: classpath:ssl/keystore.p12
#   key-store-password: gnteam307

spring:
  application:
```

```

    name: gateway-service
  profiles:
    active: prd
  jwt:
    secretKey: aWQgyiORSXixtGdZLzfh
    access:
      expiration: 3600000
    refresh:
      expiration: 18000000

```

## application-dev.yml (gateway)

```

eureka:
  client:
    fetch-registry: true
    register-with-eureka: true
    service-url:
      defaultZone: http://localhost:8761/eureka

spring:
  cloud:
    gateway:
      routes:
        - id: member-service
          uri: http://localhost:64000
          predicates:
            - Path=/api/members/app/**, /api/family/app/**, /api/facility/app/**
        - id: member-service
          uri: http://localhost:64000
          predicates:
            - Path=/api/members/web/reissue
          filters:
            - AuthorizationHeaderFilter
        - id: member-service
          uri: http://localhost:64000
          predicates:
            - Path=/api/members/web/login
        - id: map-service
          uri: http://localhost:65000
          predicates:
            - Path=/api/map/**
          filters:
            - AuthorizationHeaderFilter

```

## application-prd.yml (gateway)

```
eureka:
  client:
    fetch-registry: true
    register-with-eureka: true
    service-url:
      defaultZone: http://saveurlife.kr:8761/eureka
spring:
  cloud:
    gateway:
      routes:
        - id: member-service
          uri: http://saveurlife.kr:64000
          predicates:
            - Path=/api/members/app/**, /api/family/app/**, /api/facility/app/**
        - id: member-service
          uri: http://saveurlife.kr:64000
          predicates:
            - Path=/api/members/web/reissue
          filters:
            - AuthorizationHeaderFilter
        - id: member-service
          uri: http://saveurlife.kr:64000
          predicates:
            - Path=/api/members/web/login
        - id: map-service
          uri: http://k9b307a.p.ssafy.io:65000
          predicates:
            - Path=/api/map/**
          filters:
            - AuthorizationHeaderFilter
```

## application.properties (member)

```
spring.profiles.active=prd

# jpa
spring.jpa.show-sql=true
spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect
spring.jpa.properties.hibernate.format_sql=true
logging.level.org.hibernate.type.descriptor.sql=trace

# server port
server.port=64000

# JWT
jwt.secretKey=awQgyi0RSXixtGdZLzfH
#jwt.access.expiration=3600000
jwt.access.expiration=3600000
jwt.access.header=Authorization
```

```
jwt.refresh.expiration=18000000
jwt.refresh.header=Authorization-refresh

# swagger
springdoc.swagger-ui.path=/swagger-ui

# application
spring.application.name= member-service
```

### application-dev.properties (member)

```
spring.datasource.url=jdbc:mysql://localhost:3306/goodnews?serverTimezone=Asia/Seoul&characterEncoding=UTF-8
spring.datasource.username=ssafy
spring.datasource.password=ssafy
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.show_sql=true

redis.host=localhost
redis.port=6379
redis.password=redis

spring.data.mongodb.host=localhost
spring.data.mongodb.port=27017
#spring.data.mongodb.authentication-database=admin
spring.data.mongodb.database=goodnews

# eureka
eureka.client.register-with-eureka=true
eureka.client.fetch-registry=true
eureka.client.service-url.defaultZone= http://localhost:8761/eureka
```

### application-prd.properties (member)

```
spring.datasource.url=jdbc:mysql://k9b307.p.ssafy.io:5012/goodnews?serverTimezone=Asia/Seoul&useUnicode=yes&characterEncoding=UTF-8
spring.datasource.username=goodnewsteam307
spring.datasource.password=gnteam307
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.show_sql=false
```



```

redis.host=saveurlife.kr
redis.port=6379
redis.password=gnteam307

file.profile.path=/upload/
#file.routinecheck.path=/hello_today/upload/routine/
host.profile.url=https://k9b307.p.ssafy.io/

spring.servlet.multipart.max-file-size=20MB
spring.servlet.multipart.max-request-size=100MB

# eureka
eureka.client.register-with-eureka=true
eureka.client.fetch-registry=true
eureka.client.service-url.defaultZone= http://saveurlife.kr:8761/eureka
eureka.instance.hostname=saveurlife.kr

```

## application.properties (map)

```

spring.profiles.active=prd

# server port
server.port=65000

# swagger
springdoc.swagger-ui.path=/swagger-ui

# application
spring.application.name= map-service

```

## application-dev.properties (map)

```

spring.data.mongodb.host=localhost
spring.data.mongodb.port=27017
#spring.data.mongodb.authentication-database=admin
spring.data.mongodb.database=goodnews

# eureka
eureka.client.register-with-eureka=true
eureka.client.fetch-registry=true

```

```
eureka.client.service-url.defaultZone= http://localhost:8761/eureka
```

### application-prd.properties (map)

```
spring.data.mongodb.host=saveurlife.kr
spring.data.mongodb.port=27017
spring.data.mongodb.authentication-database=admin
spring.data.mongodb.database=goodnews
spring.data.mongodb.username=goodnewsteam307
spring.data.mongodb.password=gnteam307

# eureka
eureka.client.register-with-eureka=true
eureka.client.fetch-registry=true
eureka.client.service-url.defaultZone= http://saveurlife.kr:8761/eureka
```



### 시연 시나리오

## 등장인물

나연 - A : 뽀빠 찍는 사람, 푸바오

준용 - B : 뽀빠 찍힘 당하는 사람, E: 패널3

다영 - C : 패널 1, 블투맨

은경 - D : 패널 2, 박사님

준호 - F : 대피소 가는 길에 만난 사람 / 엄마 보고싶은 사람 , 성우

예진 - G : 구조되지 못한 사람, 실험맨

## 시나리오

### 1. 뽀빠뽀빠뽀(충남대 오늘 실외)

나연: 야 요즘 유행하는 밈 있는데 너 찍어줄까?

준용: 뭔데?

나연: 여기 서봐

나연: 뽀뽀뽀뽀뽀뽀 뽀뽀뽀뽀뽀 뽀뽀뽀

(잠시후)

준용: 야 뭐하냐 언제까지 움직일거 ?

나연: 응 나 지금 안 째고 있는데 왜 흔들리는거지?

나연, 준용 : 어? 지진이다 !!!!!!!

나연: 야 우리 어디로 가야돼 ? 이 근처 대피소 알아?

준용: 몰라.. 응 통신도 다 끊겼는데?

나연: 응 어떡해 ..

(좌절한다)

## 2. 스펀지

재난 상황에서 통신이 끊겼을 때는 네모면 해결된다

## 3. 패널(충남대 오늘 실내) 1

다영: (멍 때리기)

은경: 실? 우리 옛날에 실 전화기 기억나요 ? 실 전화기로 전화하고 그랬잖아요.

준용: 죽으면 해결된다 ..? ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ

다영: 왜 저래. 통신이 끊겼을 때 뭐가 필요하지 라디오 ?

## 4. 스펀지

재난 상황에서 통신이 끊겼을 때는 희소식이면 해결된다

## 5. 실험(충남대 오늘 야외)

(좌절한 나연, 준용의 어깨를 툭툭 치며 실험맨 등장해서 희소식 앱 보여준다)

준호 : 뭔가를 보여주는데 ...

은경 : 뭐야 뭐야 ? 희소식 ?

지도 대피소 화면 보여주고 가려고 함.

나연 : (실험맨 붙잡으며) 그냥 가시면 어떡해요  $\pi\pi$  저희랑 같이 있어주셔야죠  $\pi$

준용 : 저희 이 앱도 없고 데이터도 안 터져서 다운도 못 받아요  $\pi\pi$

(앱 뿌려주는 장면)

앱 받아서 설치하고 지도 화면 켜서

대피소 확인해서 가면서 지도 보고 있는데 점 같은게 움직임.

나연: 우리 여기에 있는 대피소로 가자

준용: 그래 근데 이 움직이는 점들은 뭐지? (점 클릭해서) 오 ? 채팅하기 기능이 있는데 한 번 해보자

(채팅)(블투맨 등장)

준용 : 안녕하세요 저희 지금 만나실래요?

준호 : 좋아요

준용 : 계신 곳으로 갈게요

그래서 채팅하기 있어서 채팅해서 F 만나서 같이 대피소에 가기

나연 : 어 ? 저있다

다같이 안녕하세요

준호 : 얼른 대피소로 가요

다영 : 패널이 에이 통신이 안 되는데 저런게 어떻게 가능해

## 6. 희소식 어플 제작한 박사님 인터뷰(블투맨 옆에서 모션)(충남대 오늘 실내)

질문 : 통신이 끊겼는데 이런 기능들이 가능한 것이 사실입니까?

은경: 네 사실입니다. 희소식 앱은 대부분의 기능이 오프라인 환경에서도 동작할 수 있습니다.

스마트폰의 Bluetooth Low Energy 기능을 Long Range 모드로 사용하여 인터넷, 전화 등의 통신이 불가능한 상황에서도 다른 사람과 통신할 수 있습니다.

Long Range 모드는 기본적으로 1대1인 경우 최대 1km까지 연결이 가능합니다.

이를 사용하여 mesh 네트워크를 구축하면 중간의 다른 스마트폰들을 거쳐 다대다로 더 멀리까지 연결이 가능합니다.

## 7. 대피소(밤)

(울면서 핸드폰만 붙잡고 있는 준호)

준호: 아 눈부셔 ㅠㅠ 핸드폰 배터리도 별로 없는데 ㅈㅈ

나연: 우린 다크모드도 있는데 ~

준호: (다크모드로 변경한 핸드폰)(울면서) 우리 엄마는 지금 잘 대피했을까...?

(알람으로 '엄마가 연결되었습니다.')

준호 : 어 ? 엄마 !!!!!!!

## 8. 마지막(밤)

(구조되지 못 한 어두운 곳에 있는 예진)

예진: (힘겹게 소리치며) 거기 아무도 없어요?

긴급 손전등 켜두면 멀리서 내 불빛 보고 구해주겠지?

아 긴급 요청도 해야겠다.

(SOS 요청하고 잠들어버림)

다영: 어? 저건 SOS?

(예진에게 다가와 기절한 예진을 깨운다)

(깨어난 예진)

다영 :

REALM

회원	member			
키	memberId	Domain	String	
휴대전화번호	phone	Domain	String	
생년월일	birthDate	Domain	String	
이름	name	Domain	String	
성별	gender	Domain	String	
혈액형	bloodType	Domain	String	
주소사항	addInfo	Domain	String	
현재상태	state	Domain	String	
최종서버연결시간	lastConnection	Domain	RealmInstant	
최종업데이트시간	lastUpdate	Domain	RealmInstant	
경도	latitude	Domain	Double	
위도	longitude	Domain	Double	
가족id	familyId	Domain	Int	

가족구성원 정보	FamilyMemInfo			
키	구성원id	id	Domain	String
이름	name	Domain	String	
휴대전화번호	phone	Domain	String	
최종서버연결 시간	lastConnection	Domain	RealmInstant	
상태	state	Domain	String	
경도	latitude	Domain	Double	
위도	longitude	Domain	Double	
가족id	familyId	Domain	Int	

그룹구성원 정보	GroupMemInfo			
키	구성원id	id	Domain	Int
그룹id	group_id	Domain	Long	
사용자이름	name	Domain	String	
최종 연결 시간	lastConnection	Domain	RealmInstant	
상태	state	Domain	String	
경도	latitude	Domain	Double	
위도	longitude	Domain	Double	

가족모임장소	FamilyPlace			
키	장소id	placeId	Domain	Int
별칭	name	Domain	String	
주소	address	Domain	String	
경도	latitude	Domain	Double	
위도	longitude	Domain	Double	
사용가능여부	canUse	Domain	Boolean	
순서	seq	Domain	Int	

수신알림	Alert			
키	알림id	id	Domain	Int
내용	content	Domain	String	
알림시간	time	Domain	RealmInstant	

모스부호	MorseCode			
키	부호id	id	Domain	Int
코드	text	Domain	String	

채팅	Chat			
키	채팅id	id	Domain	Long
채팅이름	name	Domain	String	
메시지리스트	messageList	Domain	RealmList<ChatMessage>	

메시지	ChatMessage			
키	메시지id	id	Domain	String
채팅방id	chatRoomId	Domain	String	
송신id	senderId	Domain	String	
송신자 이름	senderName	Domain	String	
내용	content	Domain	String	
시간	time	Domain	String	

지도정보	MapInstantInfo			
키	상태변동	state	Domain	String
내용	content	Domain	String	
등록시간	time	Domain	RealmInstant	
위도	lat	Domain	Double	
경도	lon	Domain	Double	

오프라인 지도 정보	OffMapFacility			
키	지도명id	id	Domain	Int
구분	type	Domain	String	
명칭	name	Domain	String	
위도	latitude	Domain	Double	
경도	longitude	Domain	Double	
사용가능여부	canuse	Domain	Boolean	
추가 정보	addInfo	Domain	String	

보유자원	AidKit			
키	자원id	id	Domain	Int
자원명칭	name	Domain	String	
자원상태	state	Domain	String	