|  |  |  |  |
| --- | --- | --- | --- |
| **소프트웨어설계 프로젝트 결과보고서** | | | |
|  | | | |
| **프로젝트명:** | | 대코알 | |
| 8 | **조** | **팀원:** | 김상천, 김성우, 배진성, 장성욱 |

1. **프로젝트 개요** (프로젝트 설명, 목적 등)

(1)**목적 및 필요성**

코로나 19 시국에 따른 사회적 거리두기의 중요성이 대두되었고, 확진자 동선에 대한 시민들의 관심이 늘어남에 따라 확진자 동선을 능동적으로 회피하기 위한 프로그램을 기획하였다.

(2) **내용 및 추진 방법**

1. 일정 주기(설정 가능)로 대구 확진자 방문 장소를 대구시 홈페이지에서 크롤링해서 갱신

2. 사용자가 확진자 방문 장소 일정 반경(설정 가능, ex: 100m) 내로 들어옴->휴대폰에 확진자 방문장소 일정 반경 내로 들어왔다고 알림이 뜸(백그라운드 동작)

3. 앱을 실행 시, 확진자 방문 장소들과 설정 반경이 사용자를 중심으로 지도에 표시됨

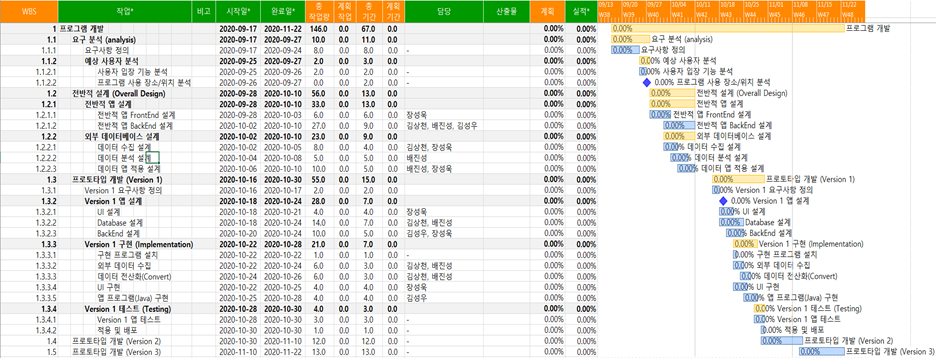
<개발환경>

안드로이드 스튜디오(IDE) - JAVA

(3) **기대효과 및 활용방안**

코로나 19 신규 확진자가 매일 수백명이 나오는 현재, 시민들은 코로나 19에 대한 불안감을 호소하고 있다. 이 앱을 통해 확진자가 방문한 장소의 일정 범위 내 접근 시 알림을 보내어 확진자가 방문한 장소와 동선을 피해 다님으로써 코로나 19의 감염기회를 줄일 수 있을 것으로 보인다. 또한, 대구뿐만 아니라 전국에서 사용할 수 있도록 만들어 코로나 19를 이겨내는 데에 기여할 수 있을 것으로 보인다.

1. **프로젝트 관리** (프로젝트 일정, 팀원별 역할 포함)



김상천: 거리계산 기능 및 핀, 반경원, 확진자정보표시 기능 개발

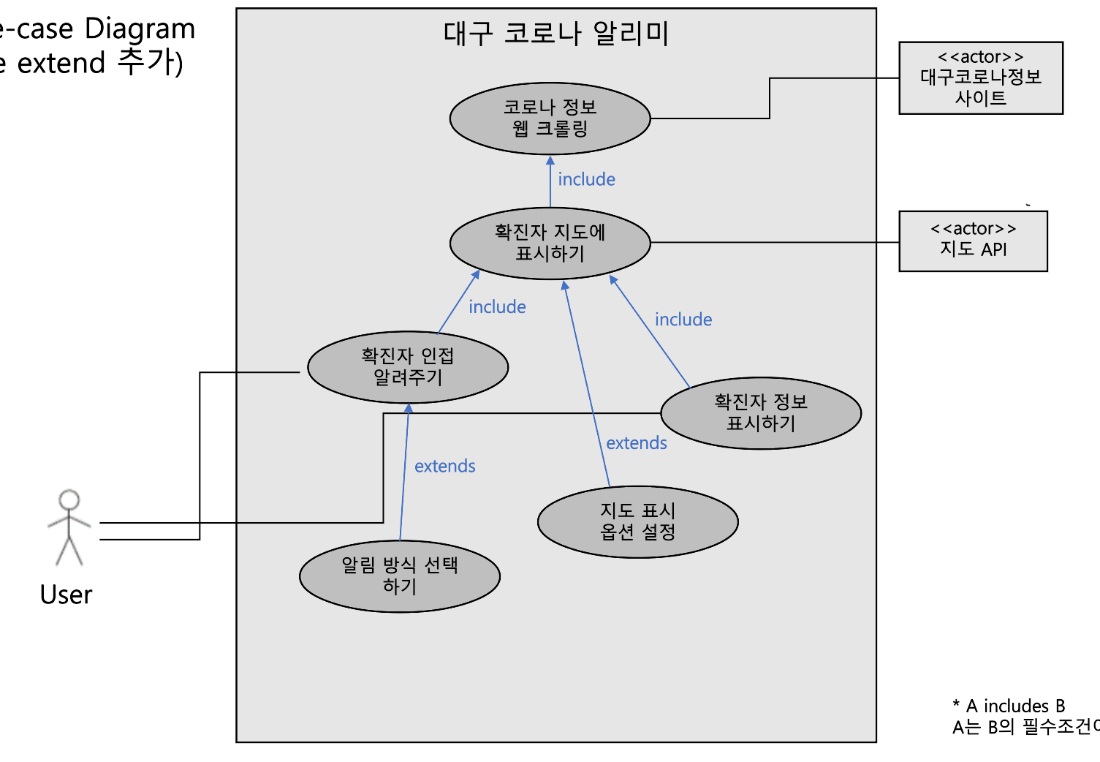
김성우: 접근, 이탈 알림 기능 개발

배진성: 대구 확진자 방문 장소 크롤링 기능 개발 및 지도 API연동, 현재위치 갱신

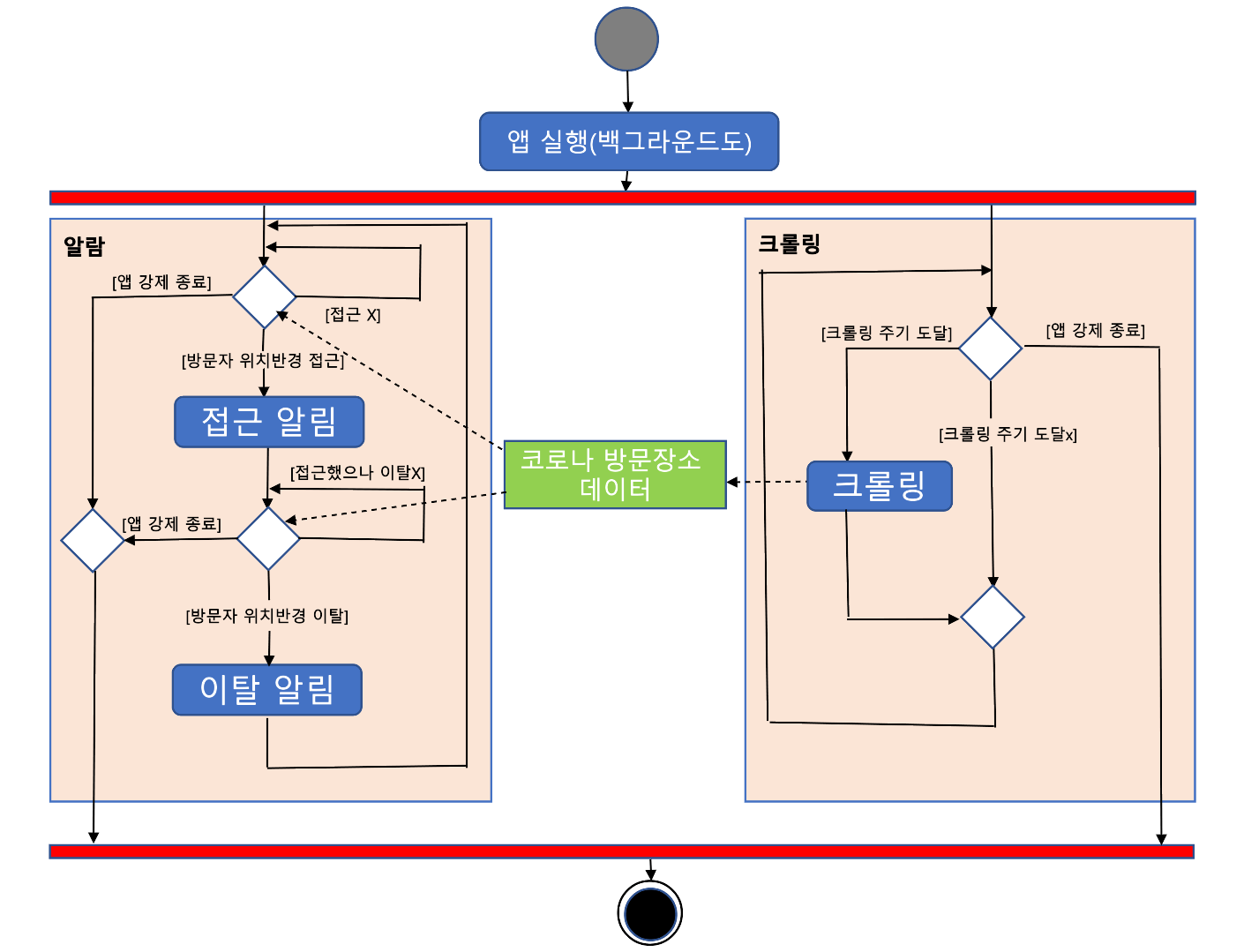
장성욱: 설정 기능 개발

1. **시스템 기능 및 동작** (유스케이스 다이어그램, BPM 액티비티 다이어그램)

**3.1 use-case Diagram**



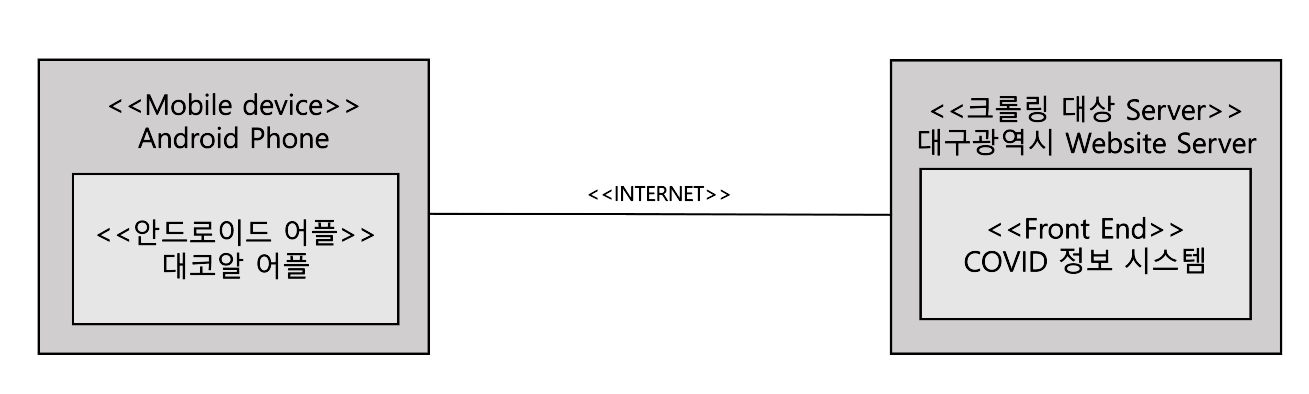
**3.2 Activity Diagram**



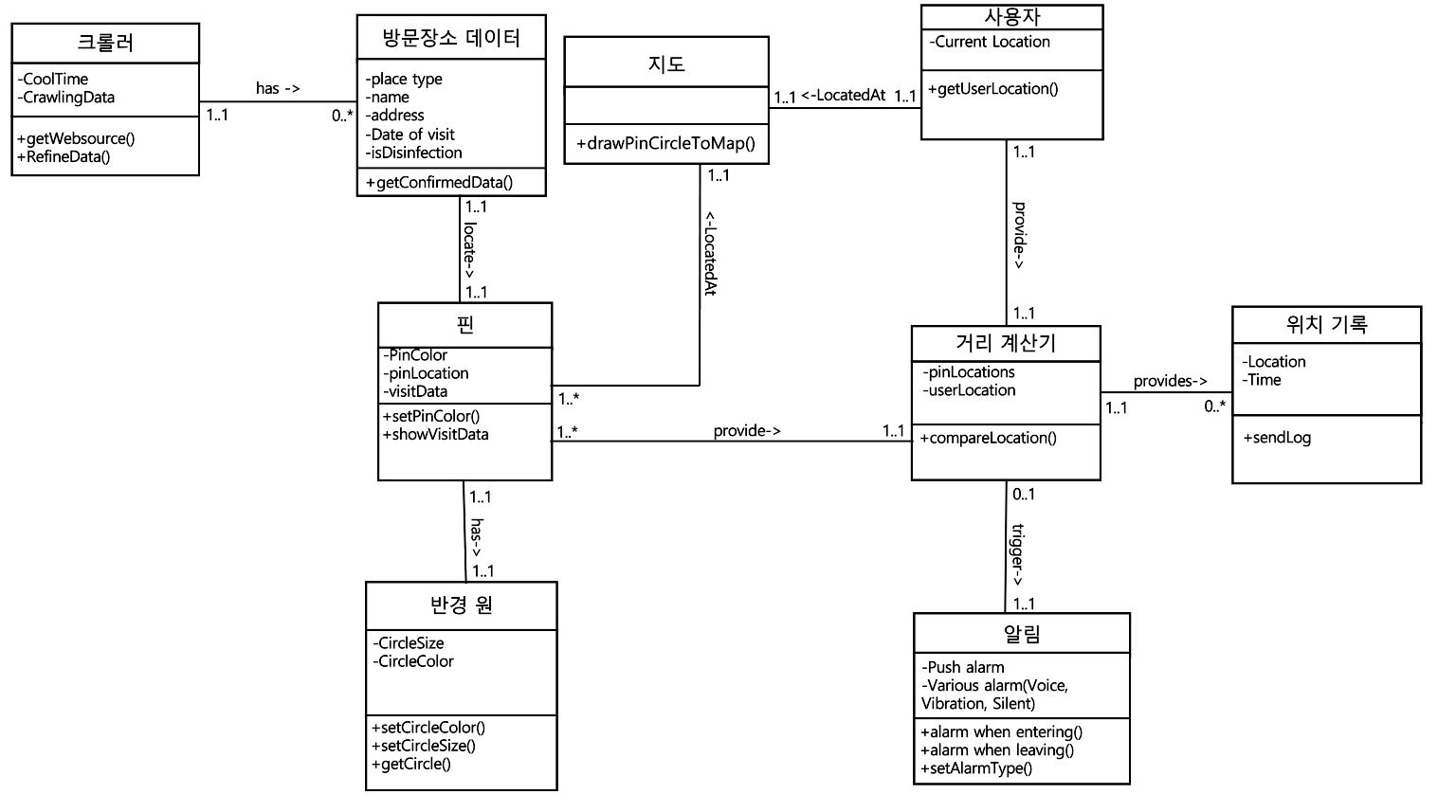
1. **시스템 아키텍처** (사용된 HW/SW 툴, 시스템 모델 등)

4.1 사용된 HW/SW 툴: android studio

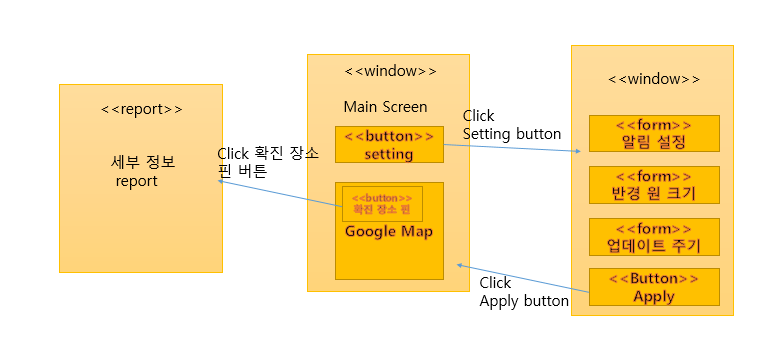
4.2 시스템 모델(Deployment Diagram)



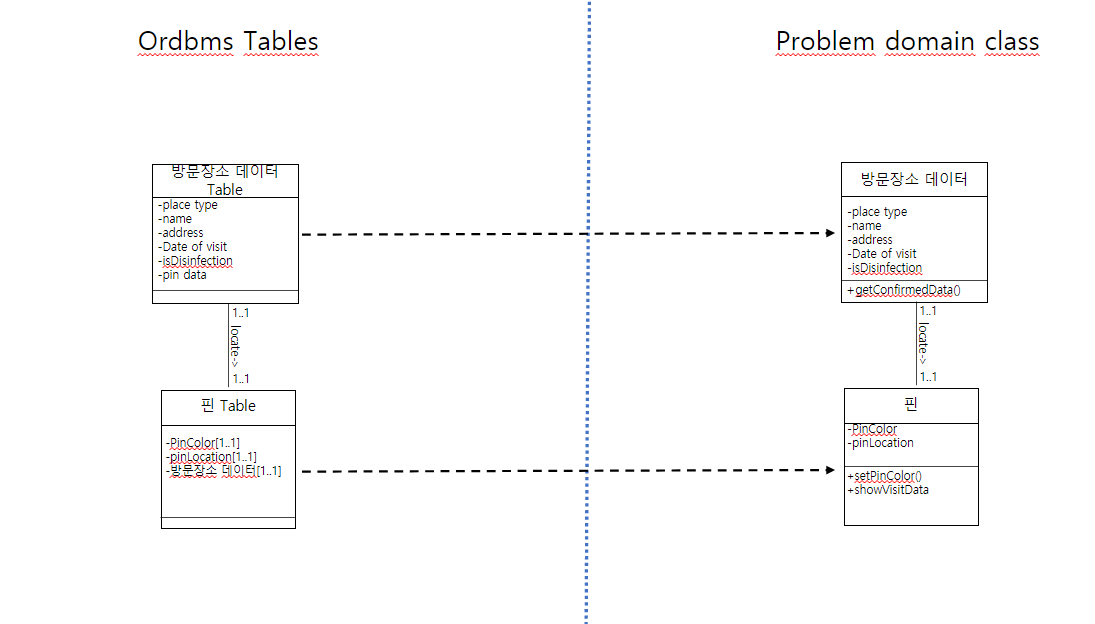
1. **소프트웨어 상세 설계**

**5.1 정적 모델** (클래스 다이어그램, 컴포넌트 다이어그램 등 구조 표현)

**5.2 GUI 설계**

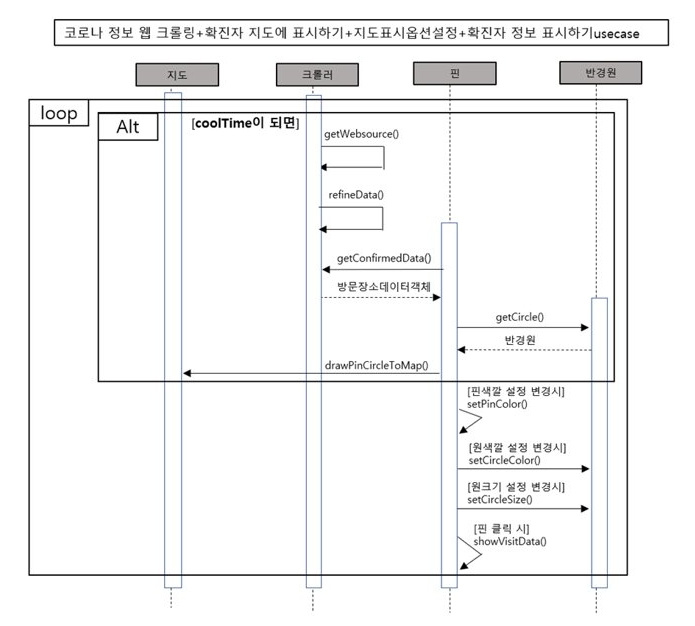


**5.3 데이터 설계**

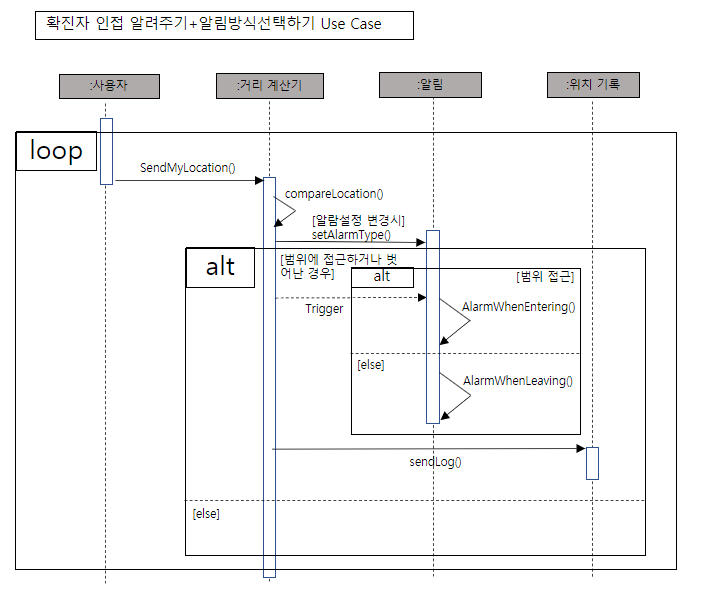


1. **소프트웨어 주요 기능** (동적 모델, 시퀀스 다이어그램 등으로 동작 표현)

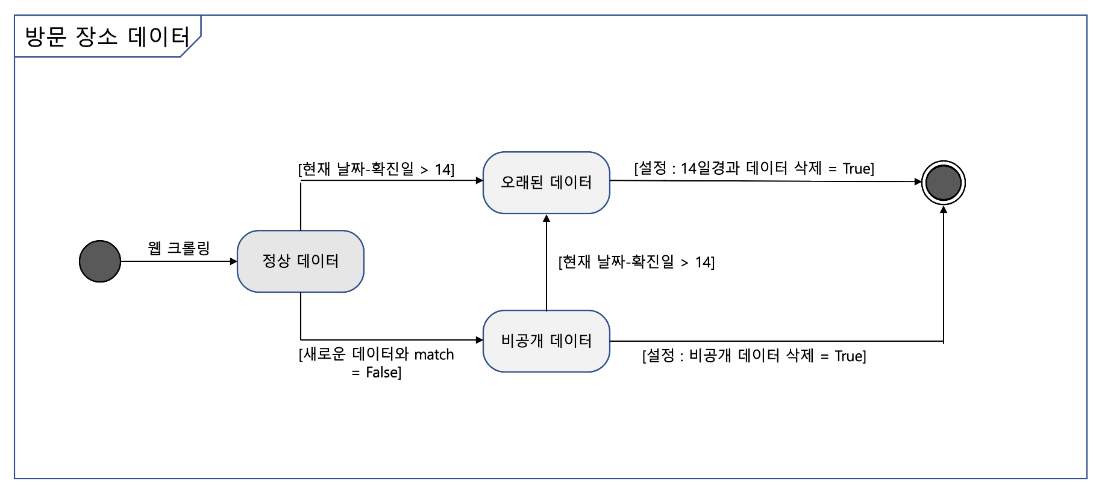
**6.1 주요기능명(시퀀스 다이어그램): 코로나 정보 웹크롤링, 확진자 지도에 표시하기, 지도표시옵션설정, 확진자 정보 표시**

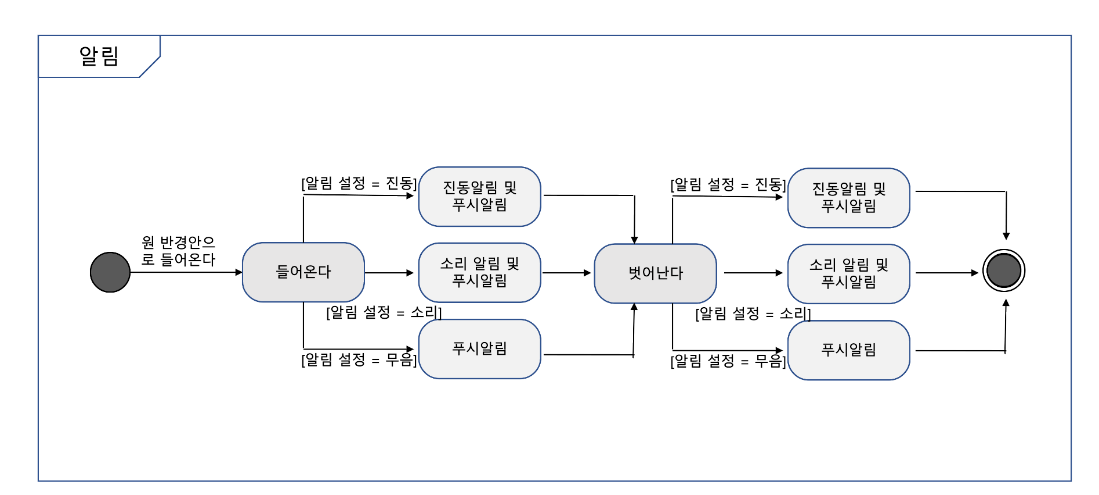


**6.2 주요기능명(시퀀스 다이어그램): 확진자 인접 알려주기, 알림방식 선택하기**



**6.3 주요기능명(상태 다이어그램): 방문장소 데이터, 알림**





1. **테스트 계획 및 결과** (유스케이스별 테스트케이스 및 실제 테스트결과 등)
   1. **코로나 정보 웹 크롤링 Use-case**
   2. 웹 소스코드 크롤링 테스트(인터넷o): 인터넷 연결 후, 크롤링 주기 변수 설정(coolTime: 24시간 이상 되면 크롤링 시작, 24시간 미만이면 크롤링 안함)

Test case 1: 24시간 값으로 설정

-> 크롤링을 실제로 해서 데이터를 반환하고, 그 크롤링 한 데이터가 사이트에 나온 원래 데이터와 일치하는 지 확인

-> 일치하면 성공

**결과: 일치(성공)**

Test case 2: 24시간 미만의 값 18시간, 12시간, 6시간 값으로 설정

-> 크롤링을 안하면 성공

**결과: 크롤링 안함(성공)**

* 1. 웹 소스코드 크롤링 테스트(인터넷 x. 예외처리 필요): 인터넷 연결 끈 후, 크롤링 주기 변수 설정

Test case1: 24시간 값으로 설정

-> 앱 실행이 중단되고, 크롤링 중단시 성공.

**결과: 앱 실행 중단 후 크롤링 중단됨(성공)**

Test case2: 24시간 미만의 값 18시간, 12시간, 6시간 값으로 설정

-> 크롤링 안하면 성공

**결과: 크롤링 안함(성공)**

* 1. **확진자 지도에 표시하기 Use-case**

2.1 지리 좌표 조회 Test

주소 String이 주어졌을 때 이를 지리 좌표로 잘 변환하는지 테스트한다. (0.001 내 오차 허용)

Test case 1:

Input - 대구 북구 대학로 80

Output – latitude = 35.8903, longtitude = 128.6120

**결과: Output- latitude = 35.8903, longtitude = 128.6120(성공)**

Test case 2:

Input – 대구 중구 달구벌대로 2100

Output – latitude = 35.8655, longtitude = 128.5933

**결과: Output- latitude = 35.8655, longtitude = 128.5933(성공)**

2.2 핀, 반경 원 정상표시 Test

방문장소 데이터 Object 필드(place type / name / address / Date of visit / isDisinfection)에 Input 값을 할당하고, 이 객체를 매개로 하는 drawPintoMap()을 호출하였을 때, 지도에 정상적으로 표시되는지 테스트한다.

Test case 1:

Input – 공공시설 / 동대구역 / 대구 동구 동대구로 550 / 10.26(월)-11.6.(금) / true

Output – (동대구역에 핀 표시)

**결과: 동대구역에 핀 표시됨(성공)**

Test case 2:

Input – 대중교통 / 719번 버스 / 율하역(4번출구)\n영남대앞 / 11.3(화), 07:16(승차) \n11.3(화), 07:37(하차) / true

Output – (율하역 4번출구 정류장, 영남대앞 정류장에 핀 표시)

**결과: 율하역 4번출구 정류장, 영남대앞 정류장에 핀 표시됨(성공)**

* 1. **확진자 인접 알려주기 Use-case**

3.1. 확진자 인접 테스트 (반경 원 300m 설정)

김성우는 수업이 끝나고 쪽문을 통해서 확진자 방문지 그린피아사우나를 향해 걸어간다. 그린피아사우나의 반경 300m안으로 들어오자 김성우에게 벗어나라고 알림을 준다.

**결과: ‘위험합니다’ 상단 알림 실행됨(성공)**

3.2 확진자 이탈 테스트

깜짝 놀란 김성우는 그린피아 사우나의 반대쪽으로 선회하여 가기 시작한다. 이후 김성우가 그린피아사우나와 반경 300m밖으로 가자 벗어나자, 안심하라고 알림을 준다.

**결과: ‘안전합니다’ 상단 알림 실행됨(성공)**

* 1. **확진자 정보 표시하기 Use-case**

4.1 핀을 터치할 때 데이터 자료구조에 접근하여 정보를 받는다.

Test case 1: 정보를 잘 받은 경우

정보를 띄워주는 팝업 창을 띄워서 정보를 보여준다. 이때 누락된 정보나, 불필요한 텍스트가 포함되지 않도록 한다.

**결과: 정보창 띄워짐(성공)**

* 1. **지도 표시 옵션 설정 Use-case**

5.1 옵션 변경 적용 Test

Test case 1: 변경된 옵션이 잘 적용되는지 테스트

-> 변경된 원 반경 값이 바뀐 즉시 실제로 지도에 반영되는지 확인

**결과: 원 반경 값을 300m에서 700m로 바꾸고 적용버튼을 누를 시 바로 반영됨(성공)**

-> 알림 설정 옵션이 바뀐 후 알림이 제대로 바뀌었는지 확인

**결과: 알림 설정을 소리에서 진동으로 바꿀 시, 이후 오는 알림들은**

**모두 진동으로 울림**

Test case 2: 반경 설정 변경으로 확진자 인접 여부가 변화한 경우

-> 알림 설정에 맞는 인접/이탈 알림을 즉시 띄우도록 함

**결과: 바로 알림이 띄워짐(성공)**

1. **프로젝트 이슈사항** (프로젝트 수행 시 문제점, 해결방법 및 느낀 점)

8.1 진성: 비대면 프로젝트 자체의 애로사항

코로나로 인하여 직접 만나서 프로젝트에 대하여 이야기를 나눌 기회를 가지기 어려웠다.

이로 인해 서로 빠르게 공유할 수 있는 부분이 목소리와 작업 화면 밖에 없다 보니 각자 수행한 내용에 대한 피드백을 받거나 각자 맡은 부분을 합치는 등의 활동이 많이 느려져서 능률이 떨어지기도 했다. 또한 비대면으로 했을 때 네트워크, 마이크 이슈 또한 진행에 지장을 줬다.

-> Microsoft Office의 웹 공유 툴을 활용하여, 각자 팀원의 진행 사항이 실시간으로 공유될 수 있게 되었고 공유 문서를 통해 각자 자료를 만든 뒤에 합칠 필요가 없게 되면서 능률이 떨어지는 문제를 해결하였다. 비대면으로 프로젝트를 하게 되어 처음에는 많이 헤맸지만 각종 협업 툴을 접했고 점차 비대면 활동에 익숙해지게 되면서 생각보다는 (준비를 철저히 하면) 비대면 회의도 충분히 괜찮음을 느꼈다.

8.2 상천: 이 프로젝트는 팀 프로젝트이므로 각 팀원간의 소통이 중요하다. 그러나, 설계를 하면서 각자의 역할만 나누고, 소통 없이 각자 맡은 작업을 진행한 적이 있는데, 이후 결과물의 통일성이 없어지는 문제가 발생했다.

->통일이 안된 작업물은 회의를 진행해 통일성을 확보하였고, 그 후 팀 활동부터는 각자 맡은 작업을 하면서도 연락망을 계속 유지하고, 각자 작업물에 대한 중간평가를 진행하는 등 여러 소통을 진행해 서로 통일성이 안 맞는 문제를 예방할 수 있었다. 앞으로는 각자 역할만 나누고 각자 작업을 하기만 해서는 안 되고, 각자 작업을 하면서도 서로 소통을 자주 해줘야 한다는 것을 느꼈다.

8.3 성우: 이 설계라는 과목 자체가 처음 접해보는 과목이다 보니 설계를 하면서 어려운 점, 설계를 하면서도 제대로 하는지에 대한 의문이 있었고 비대면으로 수업을 진행하다 보니 의문점이 있어도 선뜻 질의하기가 어려운 부분이 있었으며 팀프로젝트를 하는 과정에서도 안드로이드 앱을 이용하는 것이 되게 낯설고 어려웠다.

-> 설계 실습을 팀프로젝트를 통해서 진행을 많이 하여서 팀원에게 의견을 물어보고, 서로 팀플레이를 자주 진행하여 모르는 점을 가르쳐주고 도와가면서 실습을 진행해 나갔다. 팀원들과 함께, 설계 및 프로젝트를 진행을 해 어려운 점 및 힘든 점도 많았지만 돌아서 보니 팀원들 없이 나 혼자 했다면 절대 할 수 없었던 일을 팀원과 함께 하니 이 프로젝트를 성공할 수 있어서 기쁘고 팀원들에게 고마웠으며 내 스스로에게도 대견스러웠다.

8.4 성욱: GitHub에 익숙하지 않았음

조장이라는 직책을 맡아서 프로젝트를 이끌어 가야 하는 책임이 있는 만큼, 서로의 소스코드를 공유함으로써 더 빠르고 효율적인 개발에 도움이 되는 GitHub에 익숙해질 필요가 있었다.

-> 처음 써 보는 플랫폼이지만, 책임감을 느끼고 팀원들에게 물어보며, 스스로 연구하여 GitHub에 익숙해질 수 있도록 하였다.

8.5 상천: 구현을 하는 도중, 각 핀마다 확진자 정보가 달라야 하는데, 모든 핀들이 확진자 정보가 같은 상황 발생

-> 각 핀에 확진자 정보를 할당하는 과정에서 핀마다 고유값을 줘서 핀끼리 구별할 수 있게 해, 각 핀마다 다른 확진자 정보가 할당되게 해서 해결함. 핀마다 고유값을 주는 등의 로직도 상세설계 때 넣었으면 이런 오류가 발생하지 않았을 것이므로, 앞으로 상세설계를 할 때는 이런 버그가 일어나는 상황도 고려해서 좀 더 세밀한 설계가 필요함을 느끼게 되었다.

8.6 진성: 웹 크롤링 관련 기술적인 문제

웹에서 정보를 가져오는 작업은 매우 느린 작업이기 때문에 비동기(Thread)로 처리하는 것이 필수적인데 저장소에서 데이터를 가져오는 작업도 고려해야 해서 상당히 어려웠다. 또한 웹에서 가져온 HTML 소스에서 필요한 정보만을 패턴을 이용해 parsing 하는데, 대구 코로나 웹사이트의 HTML 소스의 패턴이 일관되지 않아서 데이터를 무결성으로 가공하기 어려웠다.

-> 비동기 관련 이슈는 프로그래밍을 더 정교하게, 데이터를 무결하게 가공하는 것은 무결성을 보장하는 알고리즘을 추가하여 해결할 수 있었다. 프로그램의 성능(쾌적함)과 프로그래밍 난도는 비례할 수밖에 없으며, 프로그램의 신뢰성을 위하여 성능을 희생해야 하는 경우도 있음을 느꼈다.

8.7 성욱: 기능 간 호환성 문제

각각의 모듈 클래스 내부에서 사용하는 변수의 이름이, 같은 역할임에도 중구난방이 된 일이 있었다. 이들을 설정 모듈을 통해 이어주는 것이 문제가 된 일이 있었다.

-> 변수명, 메소드명을 지어줄 때, 일정한 포맷을 지키면서 짓도록 하였다. (통일성 추구)

8.8 성욱: UI최적화 문제

처음 계획을 하면서 구상한 UI를 살펴본 결과, 지도 위에 이것저것 많아서 너무 난잡해 보일 수 있는 문제가 생길 것이라고 생각하였다.

-> 지도상의 핀에 원을 표시하던 기존 계획안을 수정하여, 사용자를 중심으로 원이 생성되는 방식으로 전환하였다. 핀은 핀 대로 직관적으로 위치를 나타낼 수 있었으며, 반경원이 너무 난잡하게 생기지 않고도 거리를 가늠할 수 있는 두 가지 토끼를 잡아낼 수 있었다. 또한, 화면 하단에 각종 메뉴들을 집어넣는 처음 계획안 대신, 우측에 플로팅 버튼을 하나 띄워 복잡함을 줄이고, 심플함을 추구하였다.

1. **실행 결과** (실행 결과 화면 캡처, 시연 동영상 url)

9.1 시연 동영상 url

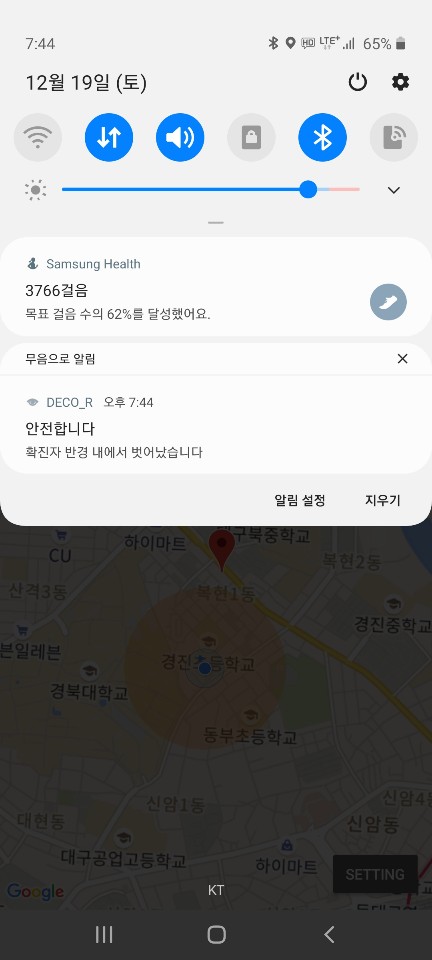
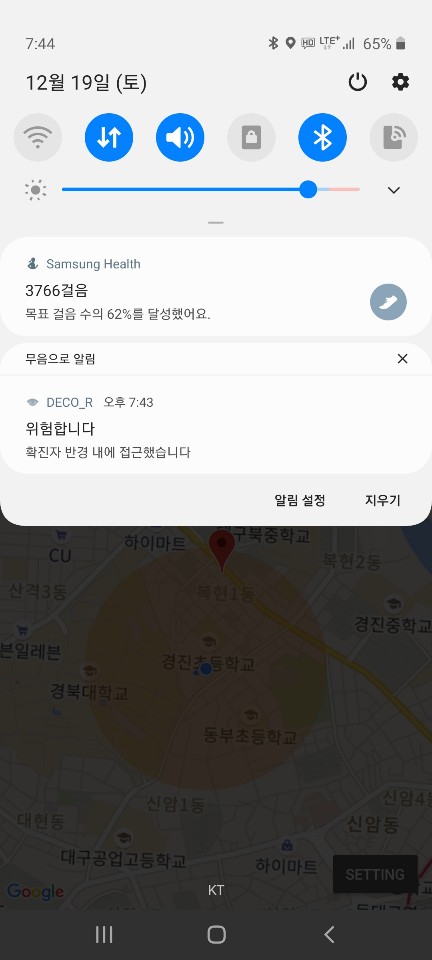
<https://www.youtube.com/watch?v=kALXEw27aB4>

9.2 시연 결과 화면

1) 앱 첫 화면

2) 접근할 시 알림

3) 벗어날 시 알림

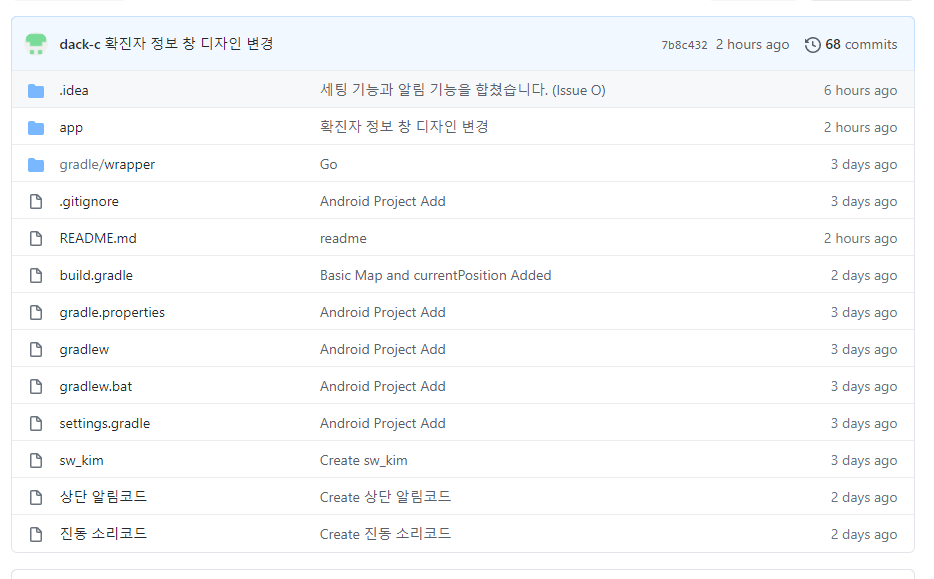




1. **GitHub 업로드 결과** (GitHub url, 업로드 현황, 팀원별 활동 현황 (Insights) 캡처 화면)

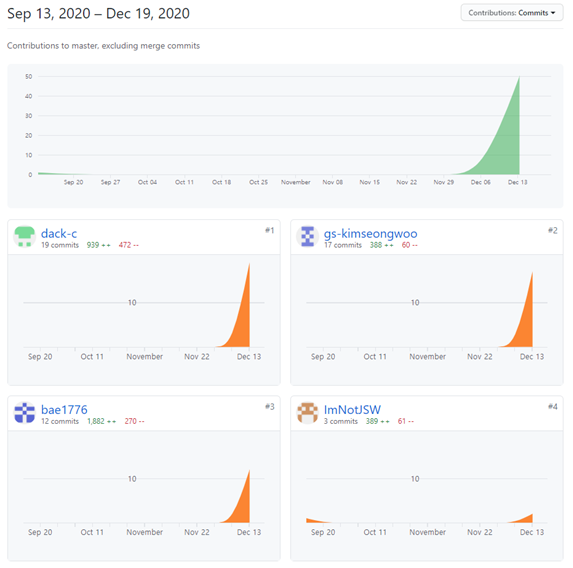
10.1 GitHub URL: <https://github.com/ImNotJSW/SWdesign-COVID19>

10.2 업로드 현황



10.3 팀원별 활동 현황

dack-c: 김상천, gs-kimseongwoo: 김성우, bae1776: 배진성, ImNotJSW: 장성욱



1. **참고자료** (프로젝트 수행 시 참고한 책 또는 사이트)

**- 참고 자료**

1. [안드로이드 앱 서적: http://www.yes24.com/Product/Goods/70932159](http://www.yes24.com/Product/Goods/70932159)
2. Java 언어 기본 서적 : <http://www.yes24.com/Product/Goods/69239293>
3. 간트 차트 프로그램: <https://xlworks.net/xlgantt-project-scheduler/>

**- 참고 사이트**

**1. 대구 확진자 동선정보:** <http://covid19.daegu.go.kr/00937400.html>

**2. 가상위치로 테스트하는 방법:** <https://webnautes.tistory.com/1064>

**3. 앱 권한 (공식 레퍼런스):**

: <https://developer.android.com/guide/topics/>[permissions/overview#permission-groups](https://developer.android.com/guide/topics/permissions/overview#permission-groups)

**4. 안드로이드 Java Thread:** <https://itmining.tistory.com/4>

**5. 주소->위경도 변환:** <https://bitsoul.tistory.com/135>

**6. Callback 디자인 패턴:** [https://medium.com/@wonjong.oh/-4941559c3370](https://medium.com/@wonjong.oh/%EC%95%88%EB%93%9C%EB%A1%9C%EC%9D%B4%EB%93%9C-%EC%BD%9C%EB%B0%B1-%ED%8C%A8%ED%84%B4-4941559c3370)

**7. 핀 정보 창**: <https://game-happy-world.tistory.com/16>

**8. 반경 원:** <https://ghj1001020.tistory.com/304>

**9. Jericho HTML Parser 공식 레퍼런스:** <http://jerichohtml.sourceforge.net/docs/javadoc/index.html>

**10. 데이터 저장 (SharedPreferences) 클래스 공식 레퍼런스:**

: <https://developer.android.com/training/data-storage/shared-preferences?hl=ko>

**11.구글 지도 API:** <https://webnautes.tistory.com/647>