

감염병 전파 예측 시스템

컴퓨터공학과 201211330 김민혁

수학과 201410334 고길재

미디어커뮤니케이션학과 201510285 조수빈

목차

1. 프로젝트 소개

1) 개요

2. 개발목표

1) 기능 요구사항

2) 비기능 요구사항

3) 위기 관리

4) success criteria

3. 사용 라이브러리

4. 아키텍처 설계

1)Architecture Diagram

2)Sequence Diagram

3)Class Diagram

4)Component Diagram

5. 질병 분석

1)처리 과정

2)결과

6. UI 및 이용 매뉴얼

7.개발 완성도

8.개발환경

9.참고자료

1. 프로젝트 소개

1) 개요

2020년 3월, 세계보건기구(WHO)는 코로나19 바이러스에 대한 팬데믹을 선언했다. 팬데믹이란 전세계적으로 특정 전염성 질병이 최악의 수준으로 유행하는 것을 의미한다. 이번 팬데믹 선언은 홍콩 독감과 신종 플루에 이어 사상 세번째이며, 세계 곳곳으로 확산된 전염병들은 수많은 목숨을 잃게 만들었다.

대한민국에서도 이러한 전염병의 유행은 사실 사스, 메르스 등의 전래가 있었기 때문에 낯설기만 한 것은 아니기에 빠른 대응을 할 수 있었다. 하지만 현재 아직 끝나지 않은 코로나와 앞으로 다가올 전염병에 대해 파악하고, 세계의 전염병 전파를 빠르게 알기 위한 필요성을 느끼고, 감염병 전파 예측 시스템을 프로젝트 목표로 잡게 되었다.

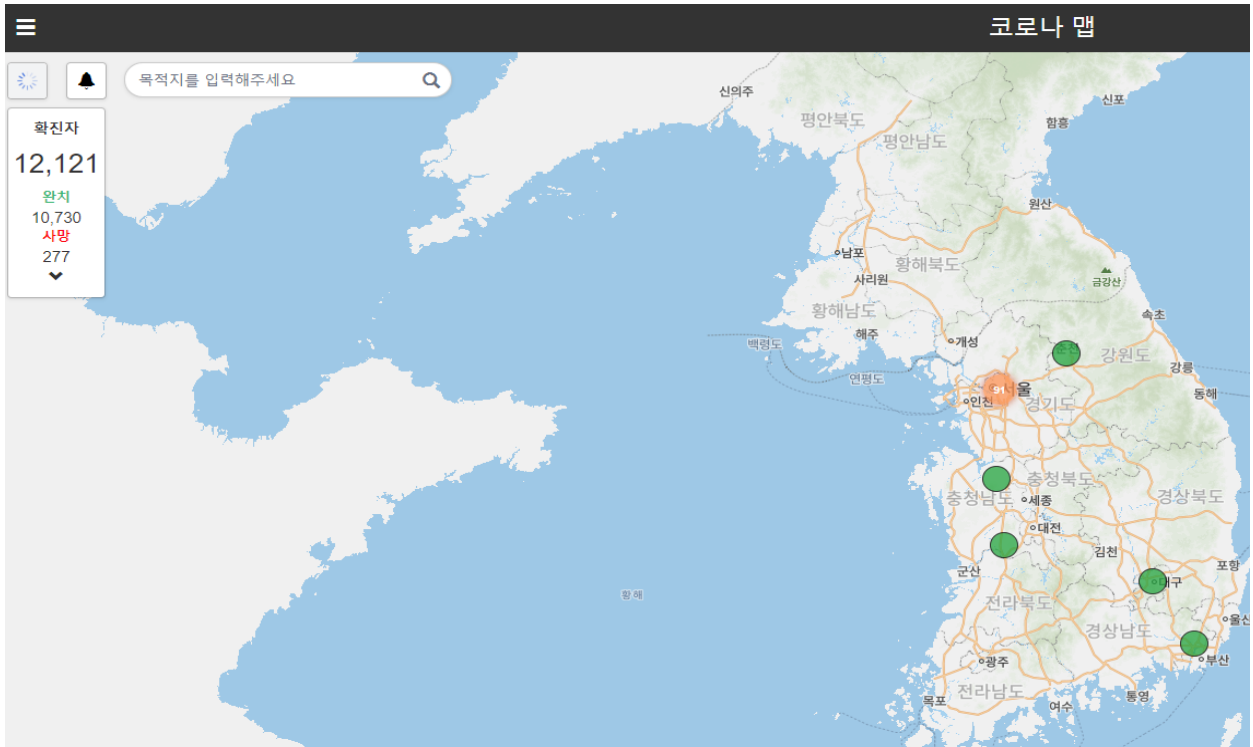
기존에도 그림 1.1, 1.2, 1.3과 같은 많은 전염병 확진자 수 파악, 진료 파악 등의 시스템이 시중에 나와있으나 그러한 기존의 시스템들은 사용자에게 전염병의 현황만을 제공한다는 한계가 있다. 현황 파악이 아닌 사용자의 차후 계획을 위해서 전세계적인 감염병 전파 정도를 예측하는 자료라는 차별점을 두기로 했다.

● 라이브 코로나 맵

UPDATE : 06.15. 00시 (한국)



[그림 1.1 <https://livecorona.co.kr/>]



[그림 1.2 <https://coronamap.site/>]

코로나19(COVID-19) 실시간 상황판

KR한국 ▾

마지막 업데이트: 2020. 6. 15. 오후 6:41:02 ↻

전 세계

8,013,477 (+101,964) 확진자	435,944 (+2,827) 사망자	4,133,716 (+71,209) 격리해제	5.44% 치사율	214 (-) 발생국
--------------------------------	----------------------------	--------------------------------	--------------	-------------------

대한민국

12,121 (+37) 확진자	277 (-) 사망자	10,730 (+12) 격리해제	2.29% 치사율	1,105,719 (+5,392) 총검사자	20,793 (-563) 검사중	1,072,805 (+5,918) 결과음성
------------------------	-------------------	-------------------------	--------------	-------------------------------	-------------------------	-------------------------------

① 용어 설명

*대한민국 수치는 질병관리본부 보도자료 기준
*대한민국 확진자 증감수치는 전일 발표 대비 변화량

[그림 1.3 <https://coronaboard.kr/>]

2. 개발목표

1) 기능 요구사항

기능 요구사항은 컴포넌트 분류를 보다 편하게 하기 위해 데이터 수집, 가공, 시각화로 구분하였다.

데이터 수집 부분에서는 파이썬을 이용해 구현한 웹 크롤러를 통해 데이터 소스로부터 필요한 데이터를 크롤링 하고 시계열 데이터로 변환해 데이터 베이스에 저장한다.

데이터 가공 부분에서는 데이터 베이스에 저장된 데이터를 C++를 이용해 구현한 질병 예측 모델의 input에 알맞은 형태로 변환한 후 데이터 피팅을 수행해 전염병 유행 종료일까지의 확진자 수를 예측한다.

데이터 시각화 부분에서는 데이터 수집 부분에서 수집한 국가별 전염병 현황 데이터와 데이터 가공 부분에서 예측한 데이터를 웹 상에 보여준다.

(1) 데이터 수집

1.1. 약 210개국의 국가별 확진자 수, 사망자 수, 회복 인원을 크롤링 한다.

- 1.2. 크롤링한 데이터를 시계열 형태로 가공한다.
- 1.3. 가공된 데이터를 데이터 베이스에 저장한다.
- 1.4. 24시간에 한 번씩 실행된다.

(2) 데이터 가공

- 2.1. 데이터 베이스에 저장된 데이터를 불러와 모델의 Input에 알맞은 형태로 가공한다.
(날짜, 전체 인구, 확진자 수, 사망자 수, 회복 인원) -> (날짜, 감염 가능군, 감염군, 회복군)
- 2.2. 데이터 피팅을 통해 전파율, 회복율을 계산한다.
 - 2.2.1. ODE Solver를 이용해 모델에 주어진 상미분 방정식을 계산한다.
 - 2.2.2. 2.2.1.의 결과를 이용해 데이터 피팅 알고리즘으로 전파율과 회복율을 계산한다.
 - 2.2.3. SIR, SEIR 모델 두 가지를 사용하여 여러 결과를 추출해 정확도를 비교한다.
- 2.3. 전파율, 회복율을 이용해 유행 종료일까지의 확진자 수를 예측한다.
- 2.4. 성능 평가를 위한 결정 계수를 계산한다.
 - 2.4.1. 0.5 이상의 결과가 나오면 피팅된 상수가 데이터를 비교적 잘 설명하고 있다고 파악.

(3) 데이터 시각화

- 3.1. 원하는 국가를 검색하여 전염병 데이터를 확인할 수 있다.
- 3.2. 국가별 전염병 확진자 데이터를 표의 형식으로 보여준다.
- 3.3. 국가별 전염병 확진자 추이를 그래프로 보여준다.
- 3.4. 국가별 전염병 종식 일자를 제공한다.

2) 비기능 요구사항

1. 사용자의 편의성을 위해 UI가 직관적이어야 한다. 사용자가 웹에서 쉽게 원하는 정보를 찾고 파악할 수 있도록, 쉽고 직관적인 UI를 제공한다.
2. 새로운 전염병에도 그대로 사용할 수 있도록 모듈화가 되어있는 엔진이어야 한다. 새로운 전염병에 대한 데이터(날짜, 전체 인구, 확진자 수, 사망자 수, 회복 인원)만 제공되면 동일한 엔진으로 전파율, 회복율을 계산해 유행 종료일까지의 확진자 수를 예측할 수 있다. 또한 급변하는 대응 상황을 묘사하기 위해서 엔진에 내장된 모델이 빠르게 추가될 수 있어야 한다.

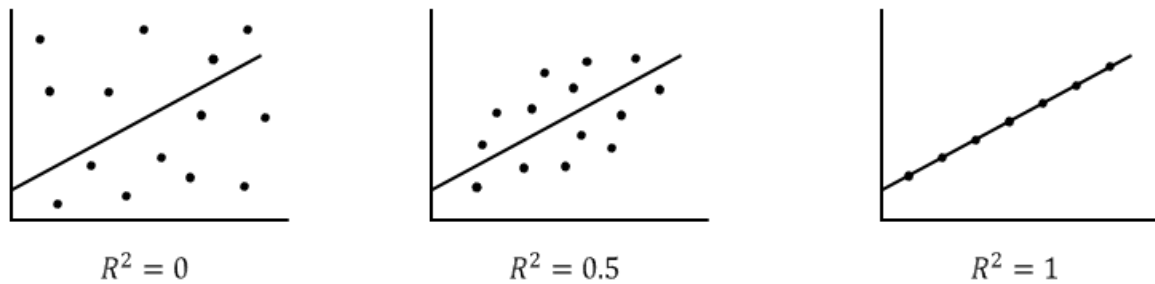
3. 새로운 전염병의 데이터를 제공해주는 데이터 소스를 찾아 크롤러만 변경하면 확진자 추이를 예측할 수 있도록 크롤러와 데이터 피팅 엔진을 모듈화 한다.

3) 위기관리

각 국가별 Covid19 확진자 관련 데이터는 WHO와 Johns Hopkins University 등 다양한 곳에서 데이터를 제공하지만 회복 인원 데이터는 몇몇 국가는 집계되지 않는 경우가 있다. 이 경우 0으로 두거나 아니면 전 세계 평균 값으로 설정하는 등 다양한 대체 방법이 있겠지만 인접한 국가들의 평균값으로 두는 것이 비슷한 문화 환경과 가까운 물리적 거리 등의 요인으로 더 유효할 것이라 생각해 인접 국가들의 평균값으로 설정해 확진자 추이를 예측하도록 했다.

4) Success Criteria

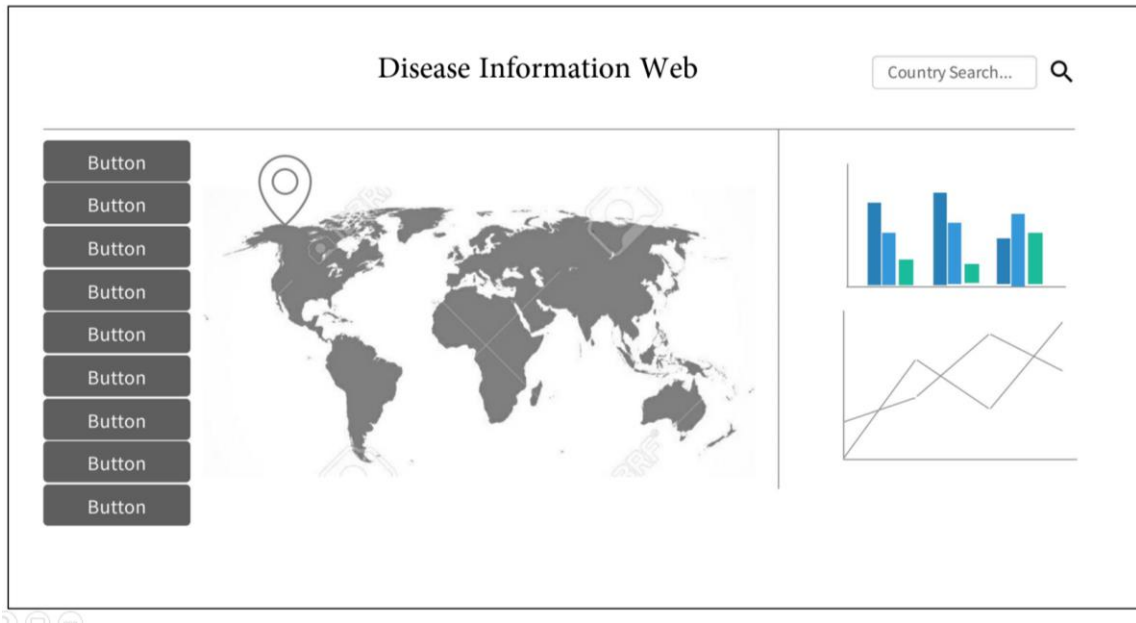
프로젝트의 Success Criteria는 우리가 설계한 예측 모델이 얼마나 유효한 예측을 하는가를 측정하기 위해 결정 계수(R^2)를 사용했으며 결정계수가 0.5 이상일 때 프로젝트가 성공했다고 판단하기로 했다. 여기서 결정 계수란 0과 1 사이의 값으로 0에 가까우면 예측한 값이 유효하지 않고, 1에 가까울수록 유효한 값이라고 할 수 있다.



[그림 1.4 결정 계수의 값에 따른 데이터 유효성]

5) 초기 프로토타입

Prototype - Illustration



3. 사용 라이브러리

3.1. 데이터 수집

- ① BeautifulSoup4 : BeautifulSoup는 HTML 또는 XML 파일에서 데이터를 추출하기 위한 파이썬 라이브러리로 웹사이트의 데이터를 크롤링할 수 있다는 것을 이용하여 많은 정보를 수집하고 수집한 정보를 활용하여 데이터 분석을 통해 새로운 결과물을 생성하도록 활용.

3.3. 데이터 시각화

- ① Express : Node.js를 위한 빠르고 개방적인 웹 프레임워크. http 유틸리티 메소드 및 미들웨어를 통해 쉽고 빠르게 API를 작성 가능할 수 있게 해준다.
- ② React : 페이스북에서 개발한 자바스크립트 라이브러리의 하나로서 사용자 인터페이스를 만들기 위해 사용한다.재사용 가능한 UI를 생성할 수 있게 해준다. Virtual DOM 개념을 사용하여 컴포넌트의 상태 변화에 따라 선택적으로 유저 인터페이스를 렌더링 해준다. 따라서 최소한의 DOM 처리로 상태 변화가 있는 컴포넌트들만 재렌더링하여 업데이트를 할 수 있게 해준다. 서버 사이드 렌더링으로 브라우저측의 초기 렌더링 딜레이를 줄

여 사용할 수 있다. 웹의 초심자로 사용하기 간편하고 빠른 개발을 할 수 있으므로 선택하여 사용했다.

- ③ Chartjs : 자바스크립트 라이브러리로, 심플하고 유연하게 디자인 가능한 차트를 제공.
- ④ sequelize : Node.js 기반의 ORM(Object-Relational-Mapping)이다. 공식적으로 PostgreSQL, MySQL, MariaDB, SQLite, MS-SQL등을 지원한다. 프로젝트의 MySQL을 백엔드에서 쉽게 사용할 수 있도록 사용하였다.
- ⑤ axios : HTTP 통신 라이브러리로, Promise 기반이며 다른 HTTP 통신 라이브러리들에 비해 문서화가 잘 되어있고 API가 다양하다. 본 프로젝트에서 Country 페이지에서 국가를 클릭할 시, 클릭을 감지하면 axios를 통해 웹서버로 request를 던져 서버에서 제공해주는 JSON 데이터를 갖고 온다. 이 데이터를 이용해 뷰 컴포넌트를 만들어 시각 자료들을 생성해 제공하였다.

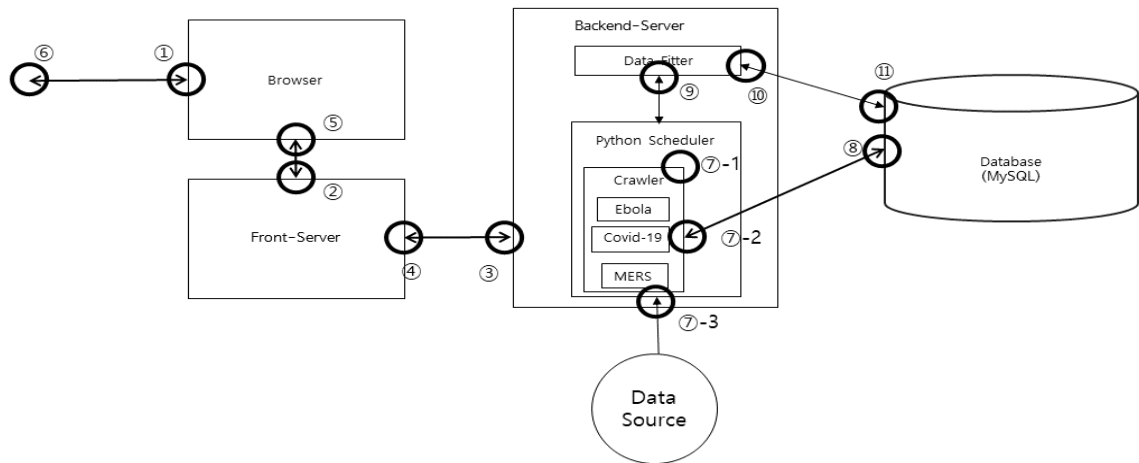
* 백엔드 서버 API

Method	URI	Description
GET	http://3.34.3.66/api/cases/:disease_name /country_name	해당 질병의 해당 국가모든 cases를 json 형식으로 response 한다.
GET	http://3.34.3.66/api/cases/:disease_name /confirmed/1	국가별로 현재 날짜의 확진자 수만 json 형식으로 response 한다
GET	http://3.34.3.66/api/predicted /:disease_name /country_name	해당 질병의 해당 국가의 예측 자료를 json 형식으로 response 한다.
GET	http://3.34.3.66/api/rsquare /:disease_name /country_name	해당 질병의 해당 국가의 rsquare를 json 형식으로 response 한다.
POST	http://3.34.3.66/api/cases/	테스트를 위한 더미 데이터를 만들기 위해 cases를 create하여 DB에 저장한다.

4. 아키텍처 설계

1) Architecture Diagram

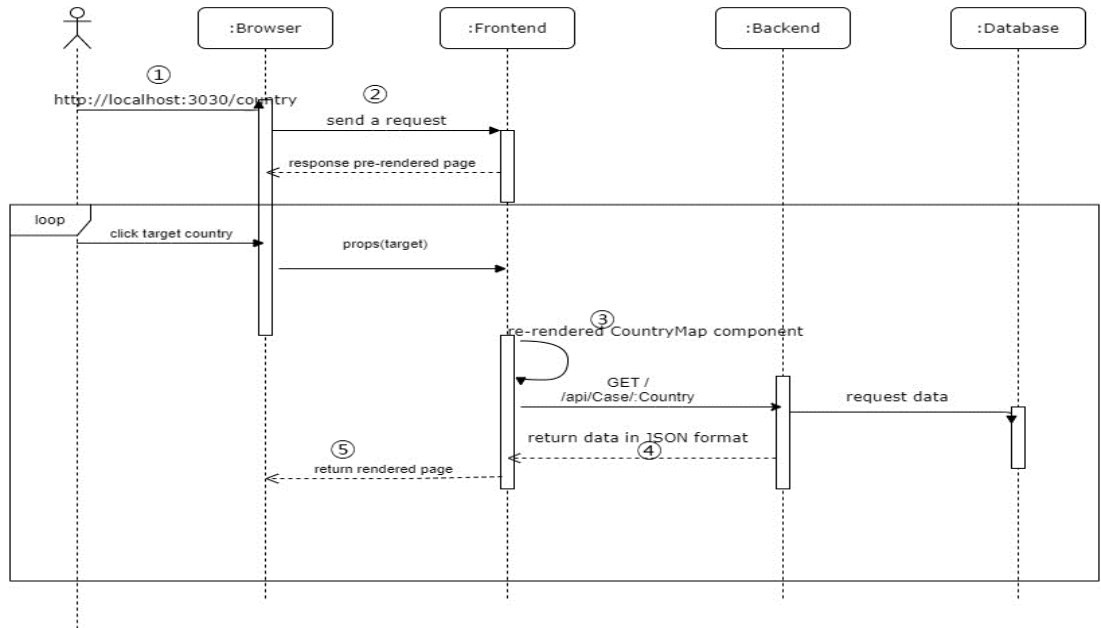
Architecture Diagram



2) Sequence Diagram

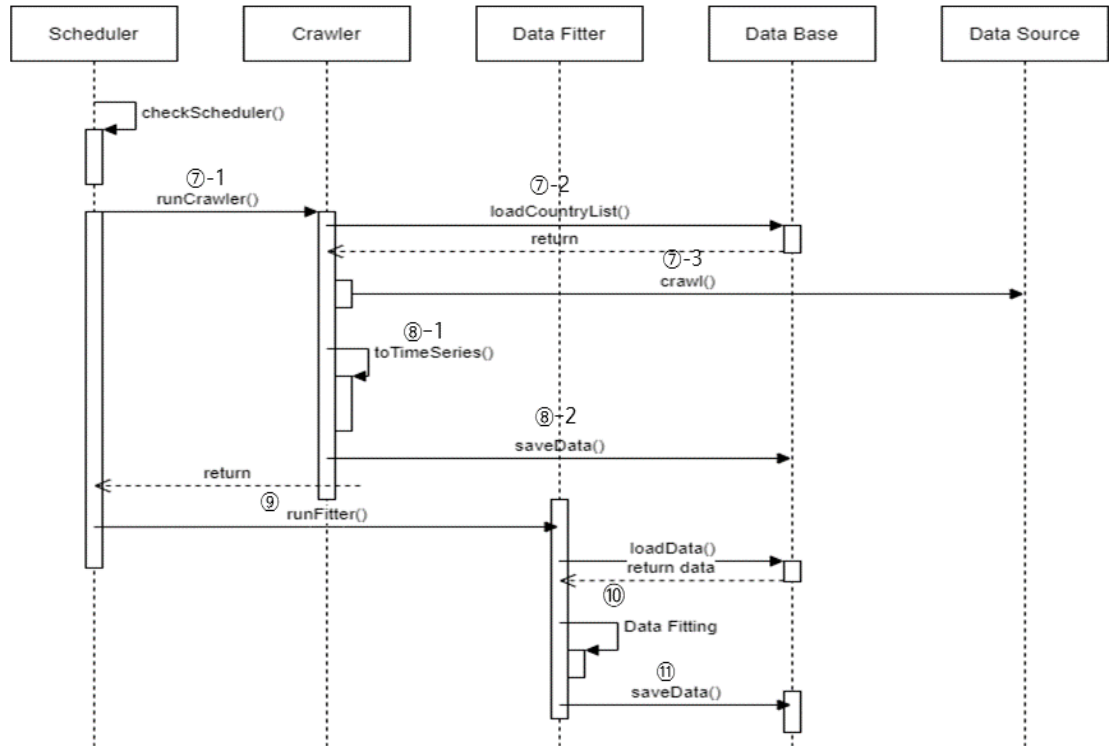
* Country 페이지에서 국가 정보 얻기

Sequence Diagram_01



- 1 - 사용자가 URL을 통해 브라우저로 접속.
- 2 - 브라우저에서 페이지를 요청.
- 3 - 서버에서 미리 렌더링된 페이지를 response 한다.
- 4 - 지도 그림에서 국가를 선택 시, 클릭한 국가에 대한 정보 property가 Frontend로 전달.
- 5 - 새로운 정보가 보내질리 해당 CountryMap 컴포넌트만 재렌더링 된다.
- 6 - 선택된 국가의 정보를 백엔드 서버로 요청한다.
- 7 - 백엔드 서버에서 DB로 정보를 요청하여 가져온다.
- 8 - JSON format으로 받아온 데이터를 프론트 서버에서 컴포넌트 안에 필요한 곳에 적용하여 재 렌더링된 페이지를 다시 브라우저로 보낸다.

Sequence Diagram_02



7-1 : 각 질병에 맞는 크롤러를 실행합니다.

7-2 : DB로부터 각 국가 명이 저장된 리스트를 불러옵니다.

7-3 : 각 Crawler가 Data source로부터 필요한 데이터 (확진자 수, 사망자 수, 회복 인원)을 크롤링 합니다.

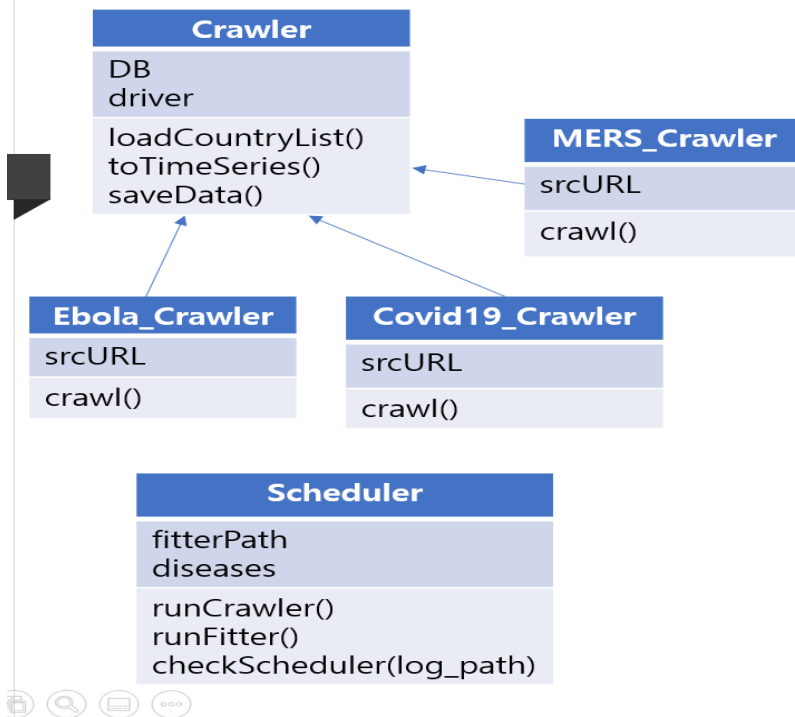
8-1 : 크롤링한 데이터를 시계열 데이터로 전환합니다.

8-2 : 시계열로 변환된 데이터를 DB의 CASES테이블에 저장합니다.

9 : 데이터 피터를 실행합니다.

3) Class Diagram

Class Diagram_01



- Scheduler

24시간에 1번씩 크롤러들과 데이터 피터를 실행시킴

- `runCrawler()`

`diseases`에 저장되어 있는 질병들에 알맞은 크롤러들을 실행한다.

- `runFitter()`

데이터 피터를 실행한다.

- `checkScheduler(log_path)`

`log`파일을 확인해 정해진 시간마다 실행되는지 확인한다.

- Crawler

- `loadCountryList()`

데이터 베이스로부터 국가 이름이 저장된 리스트를 불러온다.

- `toTimeSeries()`

크롤링한 데이터를 시계열 데이터로 전환한다.

- saveData()

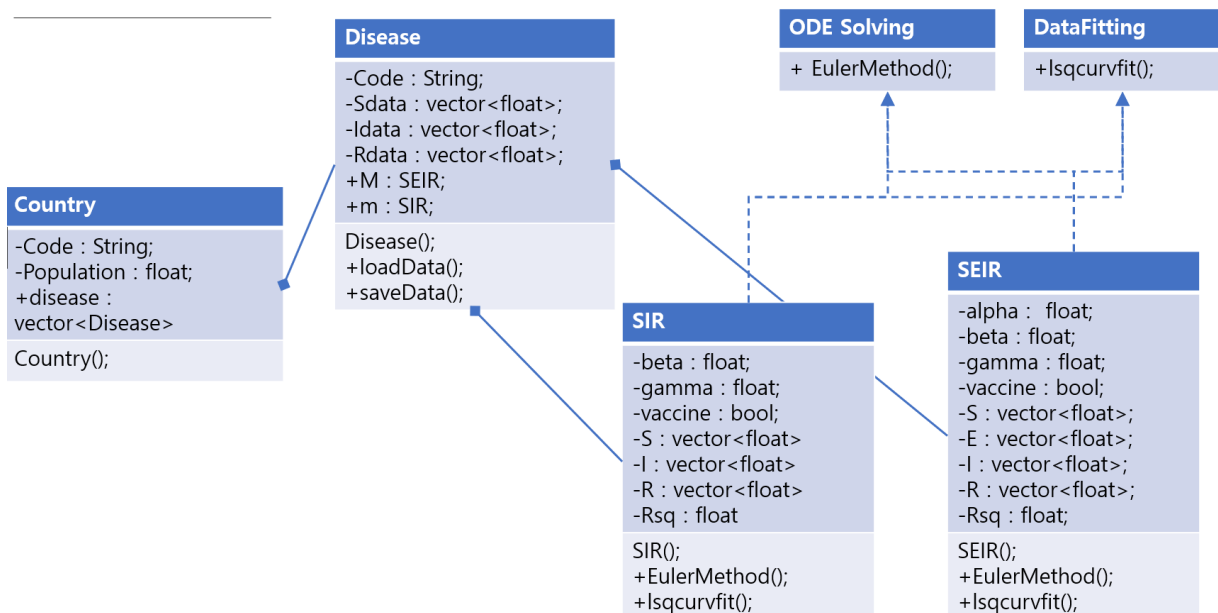
시계열 데이터로 전환된 데이터를 데이터 베이스에 저장한다.

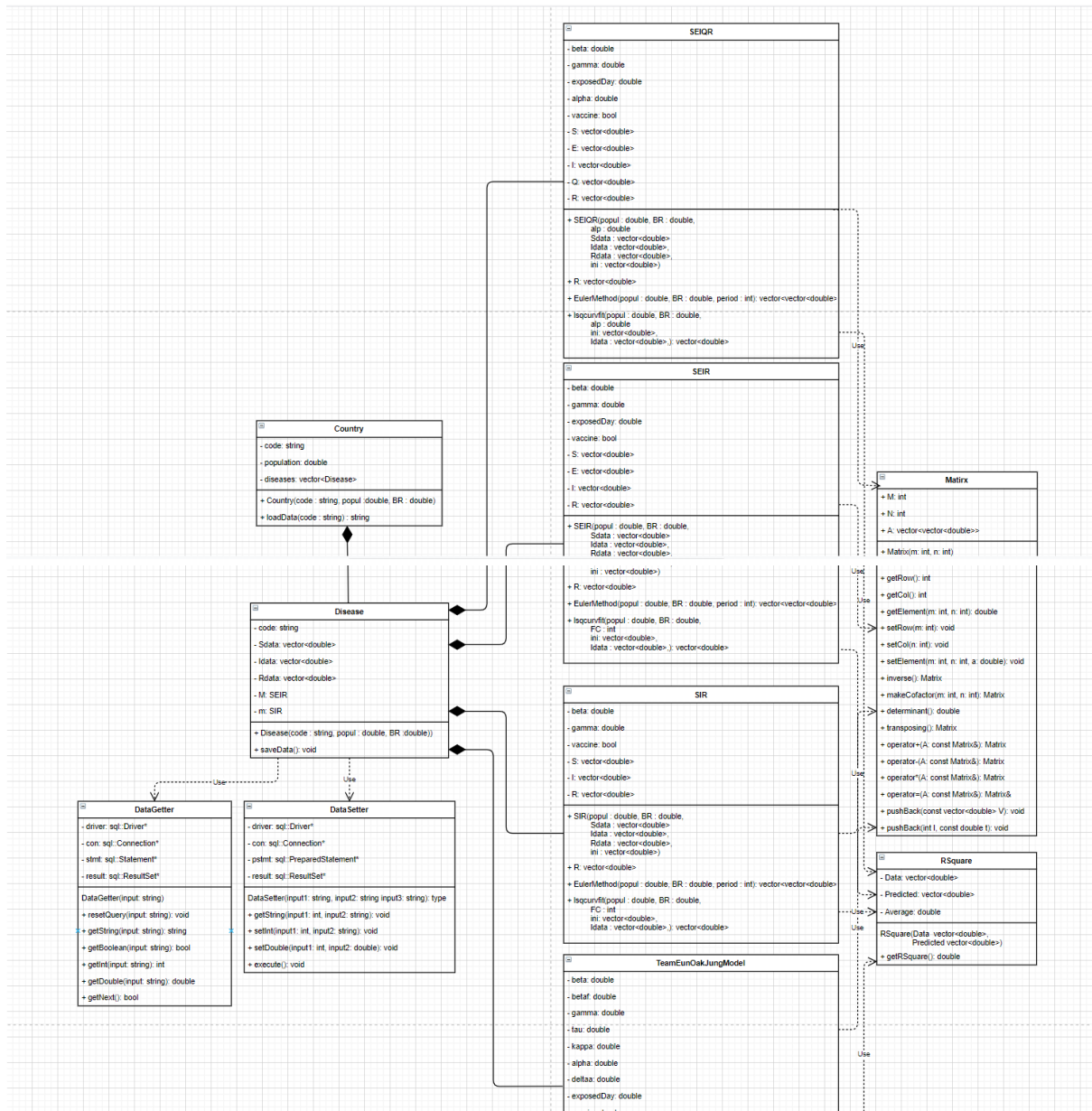
- 질병이름_Crawler

- Crawl()

srcURL에 저장되어 있는 데이터 소스로부터 필요한 데이터들 (국가별 확진자 수, 사망자 수, 회복 인원)을 크롤링 한다.

Class Diagram_02





1) Country(String counCode, float pop) 클래스

- ① Country 클래스는 main함수에서 생성된다
- ② Main에서 생성 당시에 데이터 베이스에 접근하여 국가 코드를 인식하고, 코드와 인구수를 읽어 Country를 실행한다.
- ③ Country는 데이터 베이스의 질병 테이블에 접근하여 질병 코드를 인식하고 그에 해당하는 Disease class를 생성 갖게 된다.

2) Disease(String counCode, String disCode, float pop);

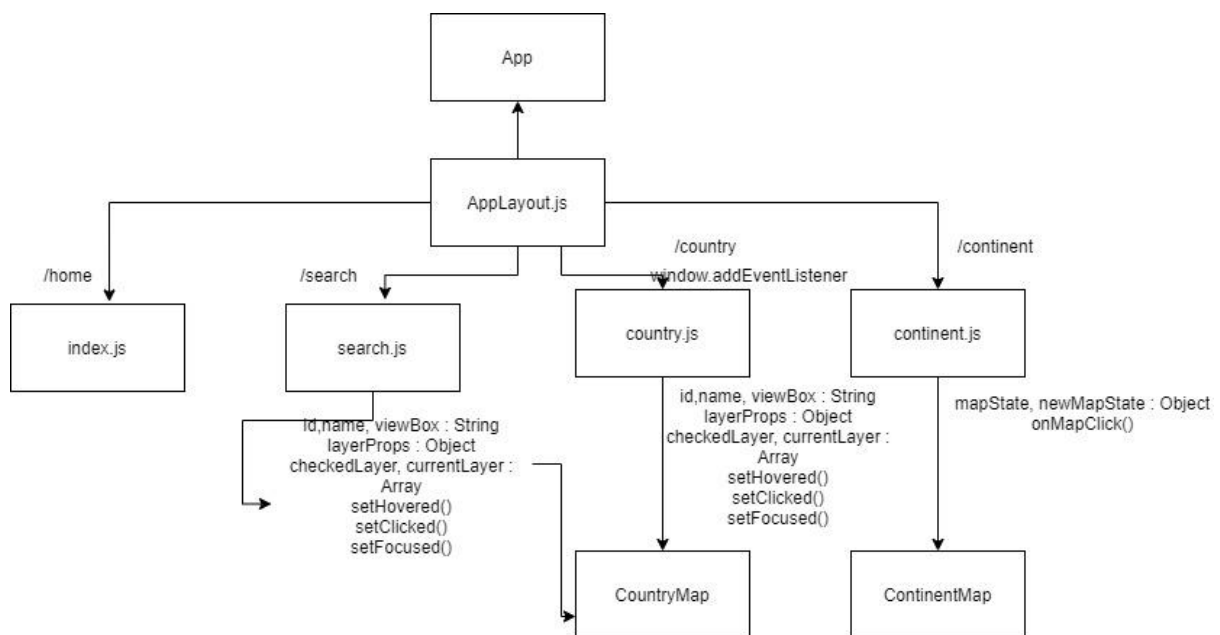
- ① Disease 클래스는 Country클래스 생성 당시에 생성되며 Country 클래스가 갖는다.
- ② 데이터 베이스에서 얻은 국가 코드와 질병 코드를 Primary Key로 데이터베이스의 Case 테이블에 접근하여 데이터를 load한다.
- ③ 로딩된 데이터를 가공한다. 데이터 베이스의 데이터와 Disease에 저장된 데이터의 관계는 다음과 같다.

$Sdata = pop - Infectious - Death - Recovery$

$Idata = Infectious - Death - Recovery$

$Rdata = Death + Recovery$

WEB Class Diagram



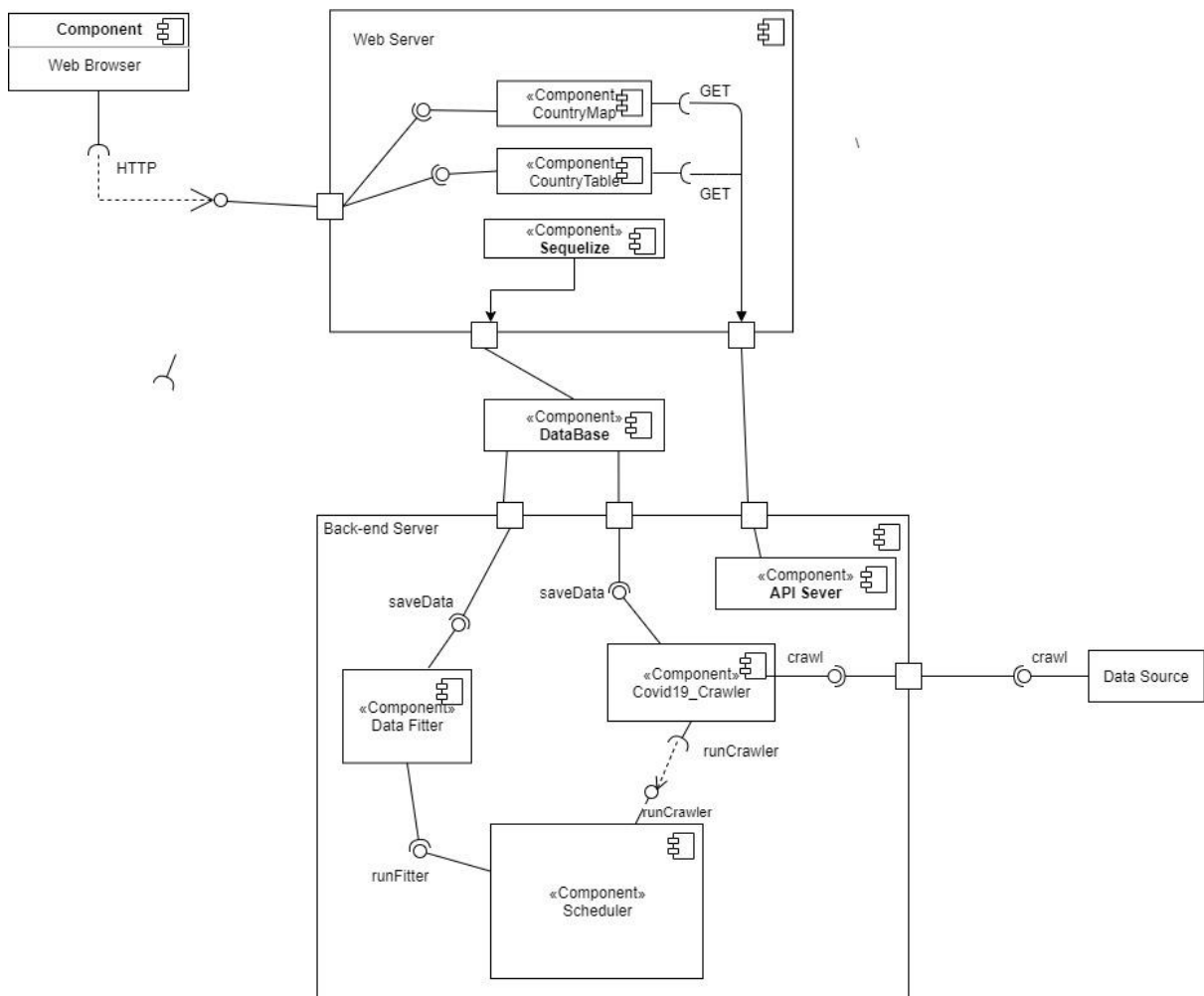
초기 search page, continent page를 따로 구성했으니 필요한 정보를 최대한 압축시키도록 하여 search page를 없애고 country 페이지에 search 란을 추가하여 검색이 가능하도록 하였다.

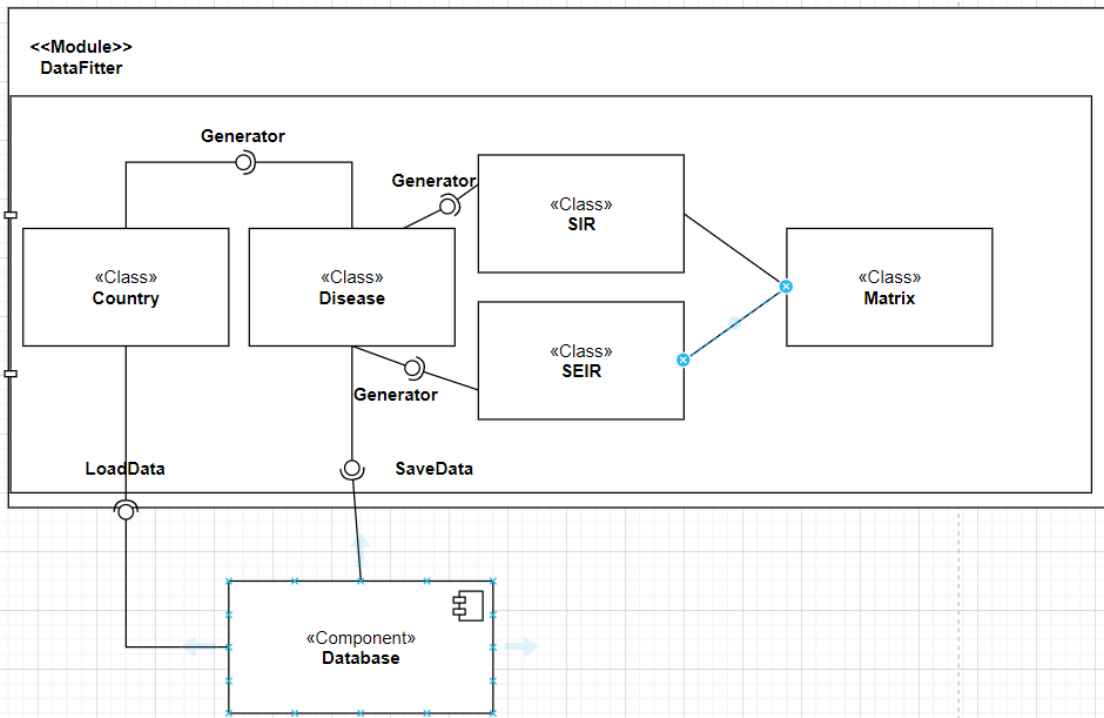
- ① AppLayout은 미리 정의된 컴포넌트로 모든 페이지에 정의된 컴포넌트들을 미리 렌더링 해있음(Menu ...)-index.js, search.js, continent.js, country.js는 웹의 pages
- ② state가 변하며 해당하는 대륙을 다른 색으로 보여줌. (아프리카 대륙 클릭 시 ->

this.mapState : {af : "map-selected"})

- ③ CountryMap : svg로 되어있는 worldmap을 클릭 시 클릭된 country의 id를 통해 클릭된 나라의 이름을 띄워줌. 이 컴포넌트 안에서 backend 서버와 통신하여 해당 국가의 데이터를 불러와 웹에 전시하게 됨.

4) Component Diagram





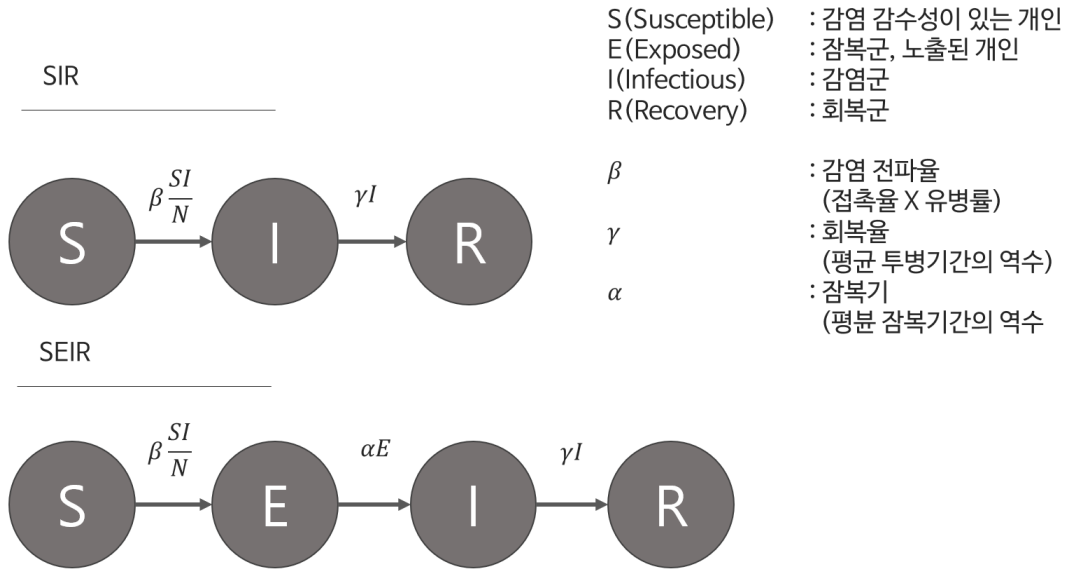
5. 질병 분석

1)처리 과정

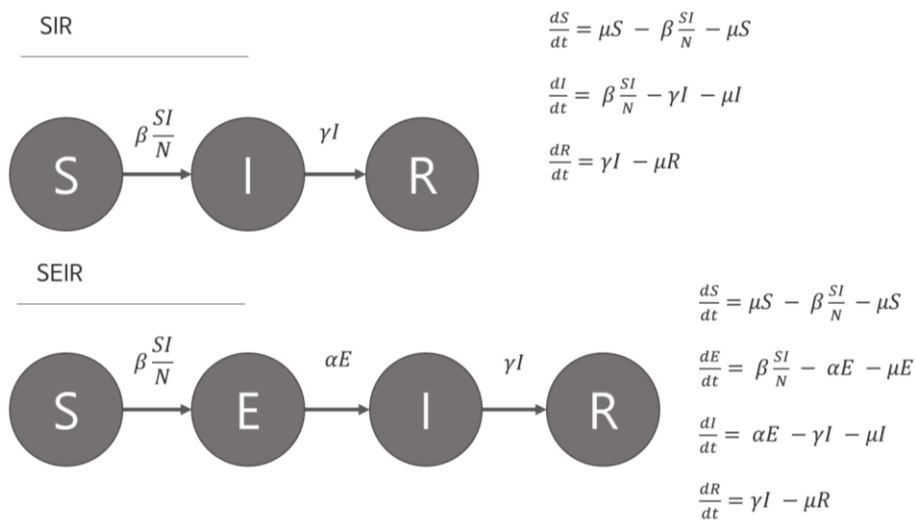
전염병은 인류 역사에 큰 영향을 미치면서 주기적으로 발생해 왔다. 멀게는 페스트, HIV 와 천연두부터 가깝게는 감기, 사스와 메르스 등 인류역사에 등장해 왔다. 이에 따라 각 전염병을 분석하기 위한 수리 모델이 필요했고, 이에 따라 SIR 과 SEIR과 같은 모델을 시작으로 최근까지 다양한 모델이 등장하고. 이러한 모델을 이용하여 모수 추정을 통해 전염병의 특성을 파악하고 다음 전염병을 대비하는 근거로서 사용되고 있다.

이러한 수리 모델의 모수 추정을 이용하여 이미 지나간 질병의 특성을 파악하는 것에 그치지 않고, 판데믹 중에 모수 추정을 통해서 판데믹의 종료를 예측 이용자의 계획에 도움을 주고자 했다.

Model Describe



Model Describe

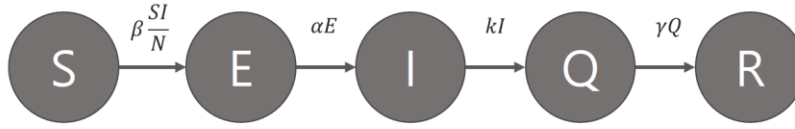


가장 기본적인 모델로 감염자와 감염 감수성이 있는 개인의 접촉, 접촉을 통해 발병 할 확률을 통해서 전염병의 전파를 묘사하는 기본적인 수리모델이다. SIR과 SEIR의 차이는 잠복기에 있으며, 잠복기가 없는 급성 전염병의 경우 SIR을 잠복기가 있는 질병의 경우 SEIR을 주로 사용한다.

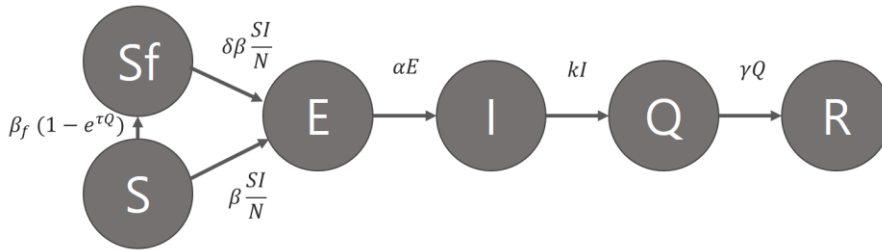
Model Describe

- S(Susceptible) : 감염 감수성이 있는 개인
- Sf(Susceptible) : 행동변화가 있는 감염 감수성군
- E(Exposed) : 잠복군, 노출된 개인
- I(Infectious) : 감염군
- Q : 격리된 감염군
- R(Recovery) : 회복군

SEIQR



TeamEunOakJungModel



SEIQR의 경우 국가가 감염자를 격리 및 치료하면서 전염성이 떨어지는 상황이 있으며 이를 묘사하기 위해서 고안된 모델이다. 대부분의 국가들이 기본적으로 이러한 방역 체계를 따르고 있으며, SEIR 이상의 묘사를 위해서 3rd iteration에 엔진 내부에 구현되었다.

TeamEunOakJungModel의 경우 건국대학교 생명수학 연구실에서 개발 및 모수추정을 하고 있는 모델로서, 국내의 확진자 데이터의 경우 인위적인 방역이 강력하게 들어가면서 자연적인 SEIR혹은 SIR모델에 너무 동떨어진 결과를 내면서 도입하게 되었다. 확진자의 수가 늘어가는 것에 따라서 감염 감수성군의 개인의 행동변화로 인해서 감염병의 확산이 억제되는 모델이다. 마찬가지로 3rd iteration에 한국 데이터 묘사를 위해서 도입되었다.

ODE Solver & Data Fitting Method

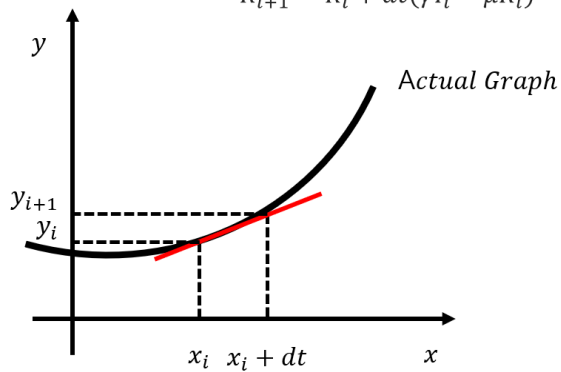
ODE Solver - Euler Method

$$S_{i+1} = S_i + dt (\mu S_i - \beta \frac{S_i I_i}{N_i} - \mu S_i)$$

$$f_{i+1} = f_i + df$$

$$I_{i+1} = I_i + dt (\beta \frac{S_i I_i}{N_i} - \gamma I_i - \mu I_i)$$

$$R_{i+1} = R_i + dt (\gamma I_i - \mu R_i)$$



수리모델을 풀어내기 위해서 엔진내부에 ODE Solver가 구현되어 있어야 한다. ODE Solver는 Forward Euler Method, Backward Euler Method와 Runge Kunta Method등 다양한 방법론이 있지만, SIR개열의 모델에서는 Forward Euler Method로 충분하여 Forward Euler Method로 구현되었다.

ODE Solver & Data Fitting Method

Data Fitting Method - Gauss-Newton Algorithm

Given Function f , Let r be residuals, $r = (r_1 \cdots r_m)$. Let β be Variables $\beta = (\beta_1 \cdots \beta_n)$

$$r_i(\beta) = u_i + f(x_i, \beta)$$

And then J is Jacobian Matrix.

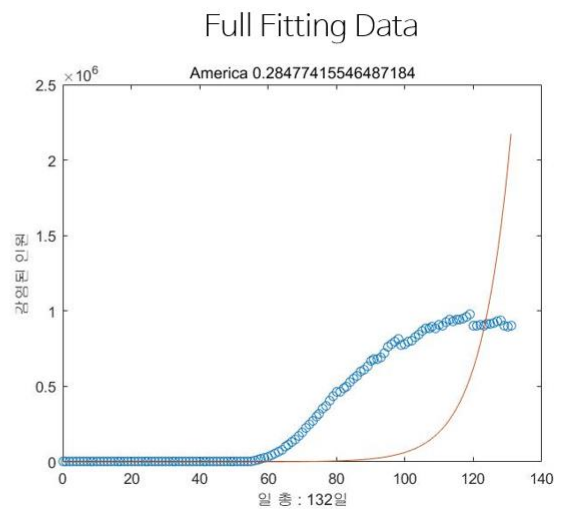
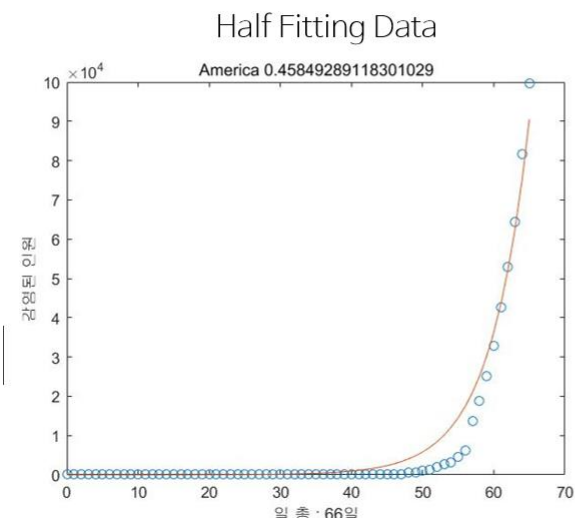
$$(J_r)_{ij} = \frac{\partial r_i(\beta^{(s)})}{\partial \beta_j}$$

Use this Jacobian Matrix, Calculate next β

$$\beta^{s+1} = \beta^s + (J_f^T J_f)^{-1} J_f^T r(\beta^{(s)})$$

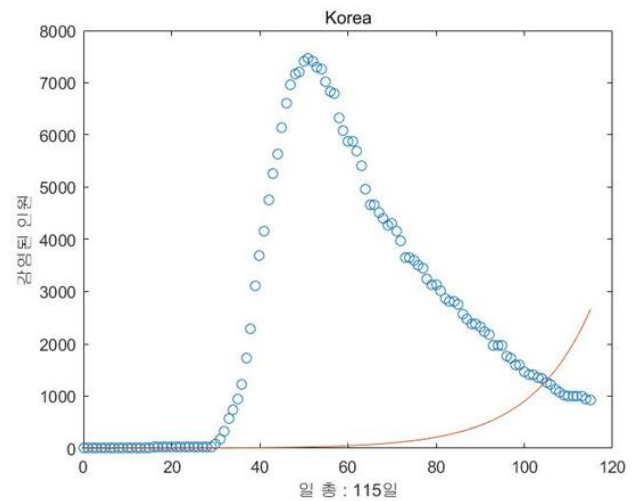
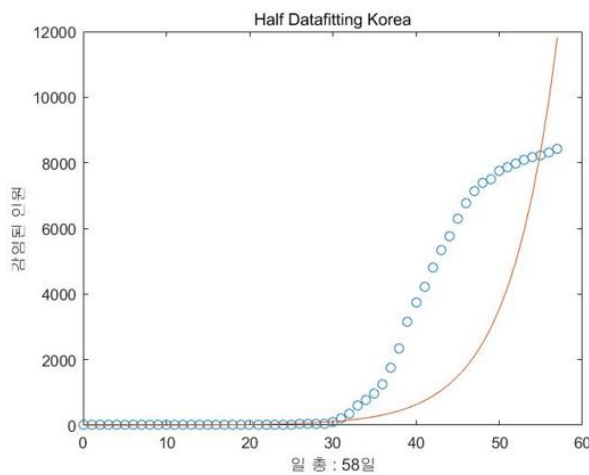
2) 결과

PASS / FAIL CRITERIA



해당 그래프의 실선은 예측 데이터, 점선은 실데이터이다.

미국의 경우, 66일치의 데이터를 피팅한 결과 실제 감염병의 진행과 유사한 예측 결과를 얻었으나, 132일치의 데이터를 피팅했을 때 유사도가 더 낮았다. 미국 외에 다른 국가도 비슷한 결과를 얻었는데, 이는 자연방목 상태일수록 데이터의 유효성이 크기 때문이다. 데이터 처리 시 인위적인 요소인 방역이 많이 투입될수록 예측 데이터와 실데이터의 차이가 크게 벌어진다. 따라서 개발도상국일수록 데이터의 유효성이 높게 나올 가능성이 커진다.



위에서 말했듯이, 한국과 같이 방역에 많은 노력을 쏟은 국가일수록 데이터 유효성이 떨어지므로, 한국과 같은 국가의 경우는 다른 모델을 적용했다. 한국에 적용한 SEIQR 모델의 경우, 국가가 감염자를 격리 및 치료하면서 전염성이 떨어지는 상황이 있으며 이를 묘사하기 위해서 고안된 모델이다. 대부분의 국가들이 기본적으로 이러한 방역 체계를 따르고 있으며, SEIR 이상의 묘사를 위해서 3rd iteration에 엔진 내부에 구현되어 있다.

6. UI 및 이용 매뉴얼

Home Country Please select

Disease Information Web

World Active Cases
8,126,277
Thu Jun 18 2020 00:34:20 GMT+0900 (대한민국 표준시)

World Death Cases
441,415
Recharge

Active
↑ 1.77 %

death
↑ 1.19 %

Country	Confirmed	Death	Recoverd
United States of America	2137707	116962	583503
Brazil	928834	45456	441729
Russia	545458	7284	294306
India	343091	9900	180013
United Kingdom	298136	41969	539
Spain	244328	27136	150376
Italy	237500	34405	178526
Peru	237156	7056	125205
Iran	192439	9065	152675
Germany	187184	8830	173600

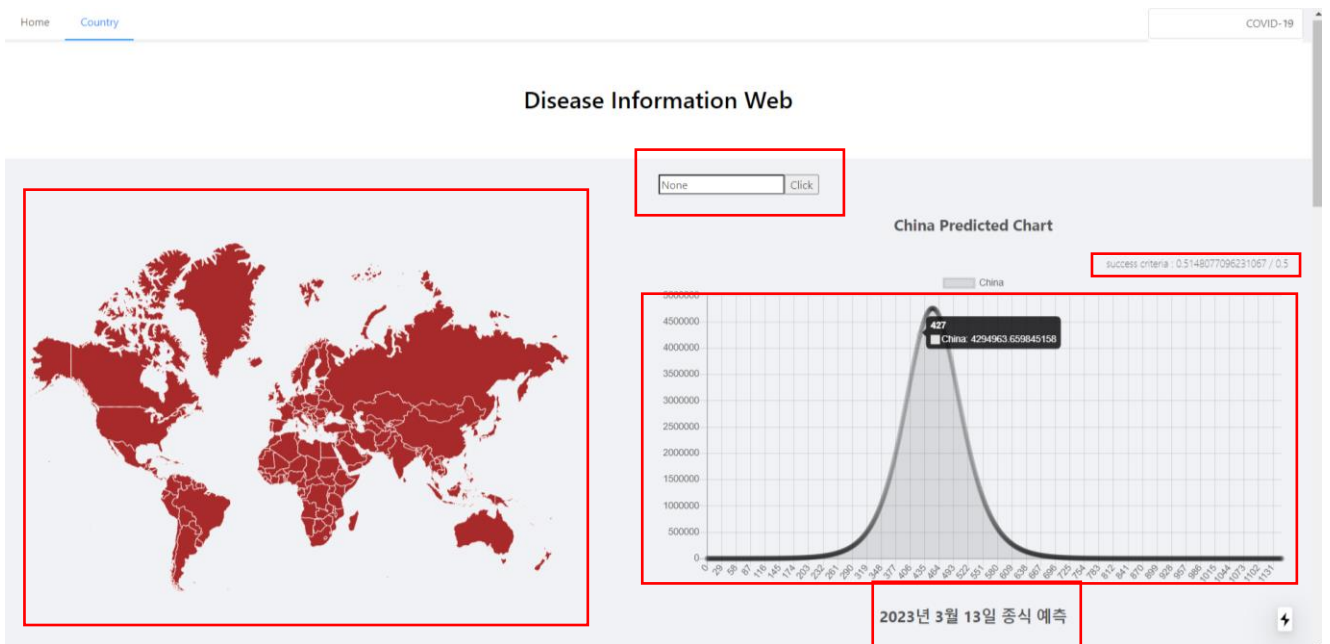
< 1 2 3 4 5 ... 22 > 10 / page

Footer

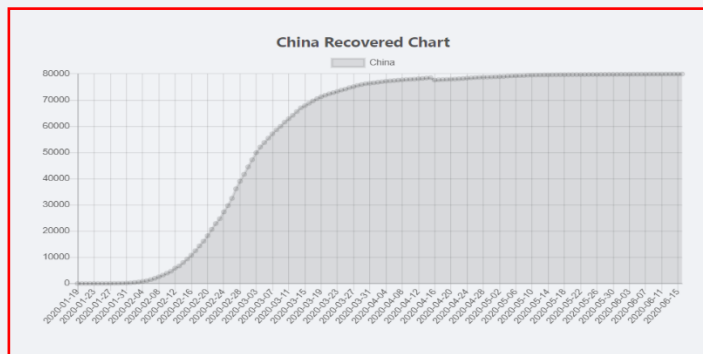
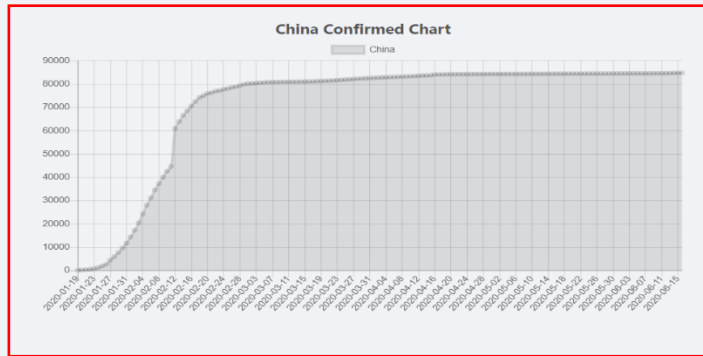
- ① 페이지 선택 Menu 컴포넌트
- ② 각종 질병을 선택할 수 있는 Cascader 컴포넌트
- ③ 전 국가의 현재까지 확진자 수와 사망자 수. 전날과 대비하여 몇 퍼센트 증가, 감소하였

는지를 보여주는 대시보드.

- ④ 국가별로 현재 날짜까지의 확진자, 사망자, 치료자 수 테이블.



- ① 국가를 클릭할 수 있는 Map 컴포넌트.
- ② 국가 이름으로 검색할 수 있는 search 컴포넌트.
- ③ 해당 국가의 Rsquare를 보여줌. 목표 success criteria 0.5에 미치지 알 수 있다.
- ④ 해당 국가의 예측 차트. 전염병 시작 날짜로부터 1131일 후에 종식 예측.
- ⑤ 전염병 첫 시작날짜로부터 1131일 후, 2023년 3월 13일 종식 예측.



- ① 클릭, 혹은 search한 국가의 확진자 차트
- ② 클릭, 혹은 search한 국가의 사망자 차트
- ③ 클릭, 혹은 search한 국가의 치료자 차트.

7. 프로젝트 목표 달성도

7.1. System testing Result

Test#	Test 항목	Pass/Fail Criteria
1	국가별 확진자 수, 사망자 수, 회복 인원을 성공적으로 크롤링하는지 확인한다	(PASS) 데이터 중 None값이 없다.
		(PASS) 데이터 중 0보다 작은 값이 없다.
		(PASS) 모든 데이터의 국가이름이 DB에 저장된 국가 이름과 같다.
2	크롤링한 데이터를 시계열 데이터로 전환하는지 확인한다.	(PASS) 모든 날짜 데이터가 date타입이다.
		(PASS) 모든 날짜 데이터가 대만 표준시 기준 1일전 값과 같다.
3	데이터를 DB에 성공적으로 저장하는지 확인한다.	(PASS) 데이터의 개수가 크롤링한 날짜로 DB에 저장된 데이터의 개수와 같다.
4	스케줄러가 24시간에 한번 실행되는지 확인한다	(PASS) log가 설정한 시간마다 기록되는지 확인
5	데이터 베이스에서 불러온 데이터가 모델에 알맞은 형태로 가공되는지 테스트한다.	(PASS) 데이터 베이스에 접근해데이터를 불러온다
		(PASS) 데이터를 알맞은 형태로 가공되는지 테스트한다.
6	ODE(Ordinary Differential Equation) Solver의 결과값이 계산되는지 테스트한다.	(PASS) ODE Solver가 ODE의 해를 구한다.
7	Matrix Class를 이용 행렬의 기본적인 연산 (+, -, *, inverse, transpose, 행렬식)이 되는지 확인한다.	(PASS) 행렬의 덧셈이 가능하다.
		(PASS) 행렬의 뺄셈이 가능하다.
		(PASS) 행렬의 곱셈이 가능하다.
		(PASS) 역행렬을 구할 수 있다.
		(PASS) 행렬식을 계산가능하다.
8	데이터를 ODE Solver를 이용해 피팅 후 전파율과 회복율이 계산되는지 테스트한다.	(PASS) 데이터 피팅을 통해 Parameter계산이 가능하다.
9	계산된 전파율과 회복율을 ODE Solver에 적용하여전염병 유행 종료시기를 예측할 수 있는지 테스트한다.	(PASS) 피팅된 Parameter를 통해 유행 종료시기 예측이 가능하다.
10	계산된 상관 계수가 Success Criteria를 만족하는지 테스트한다.	(PASS) 상관계수를 계산한 후 Success Criteria에 부합하는지 확인한다.
11	국가를 검색했을 때 해당 국가의 전염병 데이터가 웹에 출력되는지 확인한다.	(PASS)국가 검색시 확진자 데이터가 나타난다.
		(PASS)미구현으로 해당 국가의 종식날짜는 나타나지 않는다.
12	국가를 선택했을 때 해당 국가의 전염병 확진자 데이터가 표의 형식으로 웹에 출력되는지 확인한다.	(PASS)확진자 데이터가 정상적으로 나타난다.
13	국가를 선택했을 때 해당 국가의 전염병 확진자 추이가 그래프 형식으로 웹에 출력되는지 확인한다.	(PASS) 국가 선택시, 날짜별 확진자 추이가 그래프로 나타난다.
14	국가를 선택했을 때 지도 위에 전염병 종식 날짜가 웹에 출력되는지 확인한다.	(PASS)미구현으로 종식 날짜가 나타나지 않는다.

모든 테스트 통과 (100%)

7.2. Traceability Matrix

Functional Requirements	Test Case	High-Level Design	Low-Level Design	
			Method	Class
1.1	Test 1	runCrawler()	loadCountryList()	Crawler
1.2	Test 2	loadCountryList()	toTimeSeries()	
1.3	Test 3	crawl()	saveData()	Covid19_Crawler
1.4	Test 4	toTimeSeries()	crawl()	
2.1	Test 5	saveData()	runCrawler()	Scheduler
2.2	Test 6	runFitter()	runFitter()	
	Test 7		checkScheduler (log path)	
2.3	Test 8	DataFitter	Country()	Country
2.4	Test 9	countryMap()	Disease()	Disease
3.1	Test 10	continentMap()	loadData()	
	Test 11		saveData()	SIR & SEIR
3.2	Test 12		SIR(), SEIR()	
3.3	Test 13		EulerMethod()	
3.4	Test 14		lsqcurvfit()	Matrix
			operator+()	
			operator-()	
			operator*()	
			operator=()	
			makeCofactor()	
			determinant()	
			inverse()	
			transposing()	countryMap
			searchData()	
			layerProps()	continentMap
			WorldMap()	

7.3. Success Criteria

한국의 경우 기존의 모델로는 예측이 어려워 건국대학교 정은욱 교수님 팀의 모델을 사용하여 개선하였으며 다른 국가의 경우도 SEIQR 모델을 추가로 적용하여 예측 값의 유효성을 높이기 위해 노력했다. 그 결과 Covid-19 감염국 총 213개국 중 206개국 (96.7%) 의 결정 계수가 Success Criteria인 0.5를 넘는데 성공했다. Success Criteria를 넘지 못한 6개국 중 결정 계수가 가장 낮은 프랑스의 경우에도 0.487이 나옴으로서 목표치 대비 3.7% 정도 낮다. 전체적으로 보았을 때 213개국의 결정계수 평균 값은 0.67으로 프로젝트 목표였던 0.5 대비 34% 초과달성에 성공했다.

country_name	disease_name	Rsquare
The Republic of Fiji	covid19	0.999911865
Belize	covid19	0.999908981
Turks and Caicos Islands	covid19	0.999788109
Montserrat	covid19	0.999757543
Saint Lucia	covid19	0.998076338
Curacao	covid19	0.99775964
Saint Kitts and Nevis	covid19	0.997680859
Greenland	covid19	0.996265702
Ethiopia	covid19	0.987251438
Mozambique	covid19	0.984042647
Cambodia	covid19	0.50000176
Vietnam	covid19	0.500000849
Papua New Guinea	covid19	0.500000267
Saint Pierre and Miquelon	covid19	0.5
Switzerland	covid19	0.49899324
Germany	covid19	0.496521975
Israel	covid19	0.494238796
Ireland	covid19	0.487982658
New Zealand	covid19	0.486652112
France	covid19	0.483172964

상위 10개국과 하위 10개국의 국가를 정리한 표.

8. 개발 환경

OS	Ubuntu
IDE	Visual Studio,
Language	C++, JavaScript, Python

9. 참고 자료

1. Prediction of COVID19 transmission dynamics using mathematical model considering behavior changes – Eun Oak Jung

[그림 1.1] 코로나19 맵 <https://livecorona.co.kr>

[그림 1.2] 코로나맵 <https://coronamap.site/>

[그림 1.3] 코로나19 실시간 현황판 <https://coronaboard.kr>

[그림 1.4] 결정계수란?, 나부랭이의 수학 블로그, <https://math100.tistory.com/120>