SENG3011 DELIVERABLE 1

Design Details

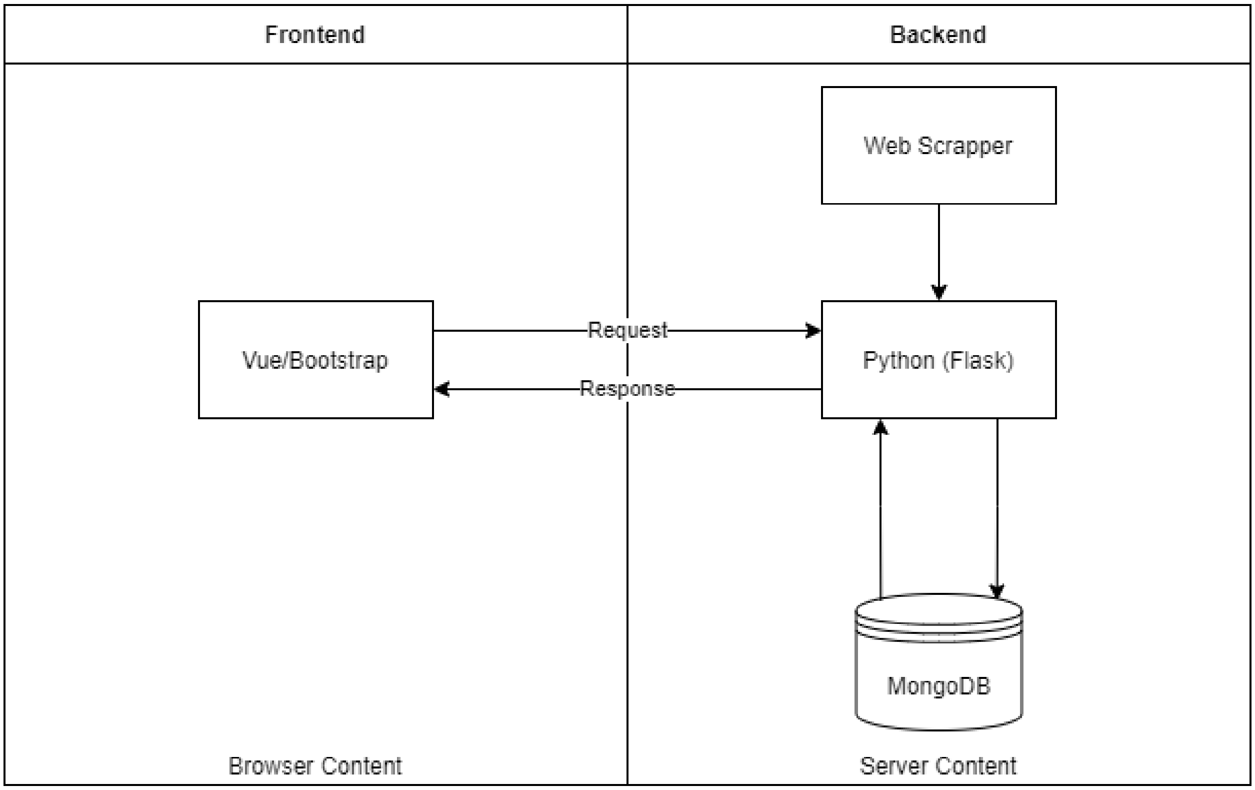
Team Brickwalls

Mentor: Yi Zhuang

Z5205003 Allen Wu
Z5207915 Monica He
Z5207001 Vishnu Pillai
Z5228933 Kshitiz Saini
Z5190299 Sheina Edeline Tengara

System Architecture

Our system will have 4 main components, the frontend, database, API and the web scraper which we will be utilising. The front end will utilise the Vue.js javascript framework, the backend database will be utilising a MongoDB database and the API will be written in Python utilising Flask and will be deployed on AWS.



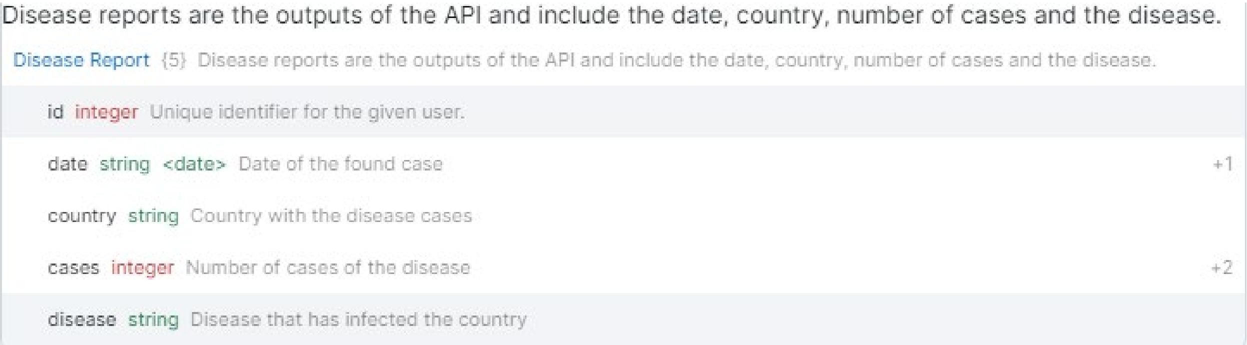
Backend design

The backend consists of 3 components, the python API, MongoDB database and the web scraper. The web scraper will be utilised to extract data from the datasource which is http://outbreaknewstoday.com. Data will be extracted regularly with the use of the API calls to ensure that the data is current and up to date. This data will be stored in the database for use for the frontend.

The API itself will be implemented in Python and will utilise the lightweight web development framework, Flask. Flask is currently one of the most popular web frameworks for python and has a plethora of resources and documentation to assist our development of our system. It also allows for the handling of API calls and can be integrated with a wide variety of database implementations, such as the MongoDB database we are planning on utilising.

API Design

The main models that the API will output or take in are disease reports. The disease reports are in JSON format and have the following structure; id, date, country, cases and disease.



Disease reports are the outputs of the API and include the date, country, number of cases and the disease.

Disease Report {5} Disease reports are the outputs of the API and include the date, country, number of cases and the disease.

id integer Unique identifier for the given user.

date string <date> Date of the found case +1

country string Country with the disease cases

cases integer Number of cases of the disease +2

disease string Disease that has infected the country

In the case that an article contains multiple cases for multiple diseases then 2 separate disease reports will be created.

The API was designed on Stoplight.io and the JSON file of the API can be viewed in the github repository, in PHASE_1/API_Documentation/Disease-Reports-API.v1.json.

The endpoints for the API will be:

GET

- /getReportID/{id}: retrieves disease report with the same corresponding id
- /getReportCountry/{country}: retrieves a list of disease reports with the same corresponding country
- /getReportDisease/{disease}: retrieves a list of disease reports with the same corresponding disease
- /getReportDate/{date1}/{date2}: retrieves a list of disease reports within the range of the passed dates
- /getReportDiseaseCountry/{disease}/{country}: retrieves a list of disease reports with the corresponding disease and country

PUT

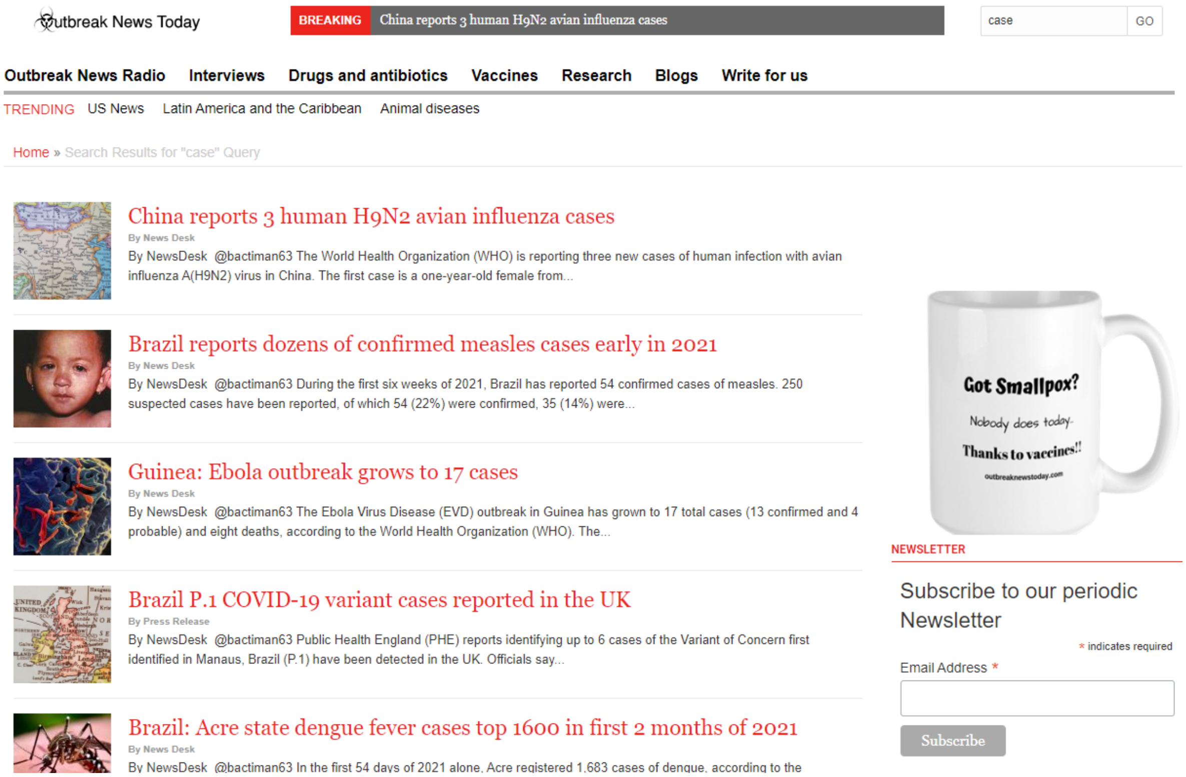
- /addReport: adds a disease report

Web Scraper

The Web Scraper utilises python and the corresponding Beautiful Soup library. The Beautiful Soup allows for the parsing of structured data to easily access data within the HTML.
Since the Web Scraper will be extracting data from http://outbreaknewstoday.com/, the web scraper will need to also act as a crawler and dynamically traverse through the many posts.

The main issue for the scraper is to determine which post within the website contains information on detected cases within the country. For example this post is a discussion on the Malaria situation in Venezuela and does not mention cases. To ensure that the web scraper can access these cases, it will start by searching "case" within the search field and then traverse all of the posts within the search field. This would increase the chance that the scraper encounters posts with relevant information.

The Web Scraper will then access these posts and extract the relevant information. These are the date of the article, country with the infected people, the disease that has infected people and also the number of cases of this disease. With this information the disease report can then be added to the database as a simple JSON file. The format of this disease report is shown in the previous section.



China reports 3 human H9N2 avian influenza cases

Home » Search Results for "case" Query

China reports 3 human H9N2 avian influenza cases

By News Desk

By NewsDesk @bactiman63 The World Health Organization (WHO) is reporting three new cases of human infection with avian influenza A(H9N2) virus in China. The first case is a one-year-old female from...

Brazil reports dozens of confirmed measles cases early in 2021

By News Desk

By NewsDesk @bactiman63 During the first six weeks of 2021, Brazil has reported 54 confirmed cases of measles. 250 suspected cases have been reported, of which 54 (22%) were confirmed, 35 (14%) were...

Guinea: Ebola outbreak grows to 17 cases

By News Desk

By NewsDesk @bactiman63 The Ebola Virus Disease (EVD) outbreak in Guinea has grown to 17 total cases (13 confirmed and 4 probable) and eight deaths, according to the World Health Organization (WHO). The...

Brazil P.1 COVID-19 variant cases reported in the UK

By Press Release

By NewsDesk @bactiman63 Public Health England (PHE) reports identifying up to 6 cases of the Variant of Concern first identified in Manaus, Brazil (P.1) have been detected in the UK. Officials say...

Brazil: Acre state dengue fever cases top 1600 in first 2 months of 2021

By News Desk

By NewsDesk @bactiman63 In the first 54 days of 2021 alone, Acre registered 1,683 cases of dengue, according to the

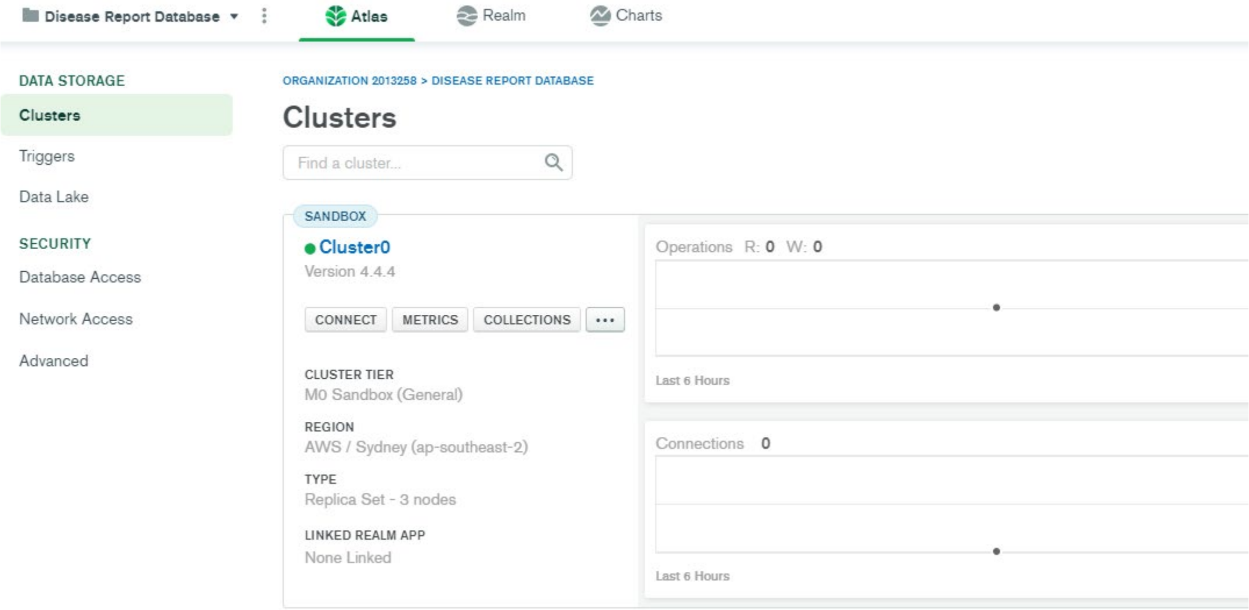
Subscribe to our periodic Newsletter

* indicates required

MongoDB database

The MongoDB database will be hosted on MongoDB Atlas, which is a dedicated cloud based database service designed for MongoDB users. It can also utilise AWS and allows for the manual editing and viewing of the database entries.

The backend API will link with the use of the PyMongo library which allows for ease of integration with the dedicated database.



Clusters

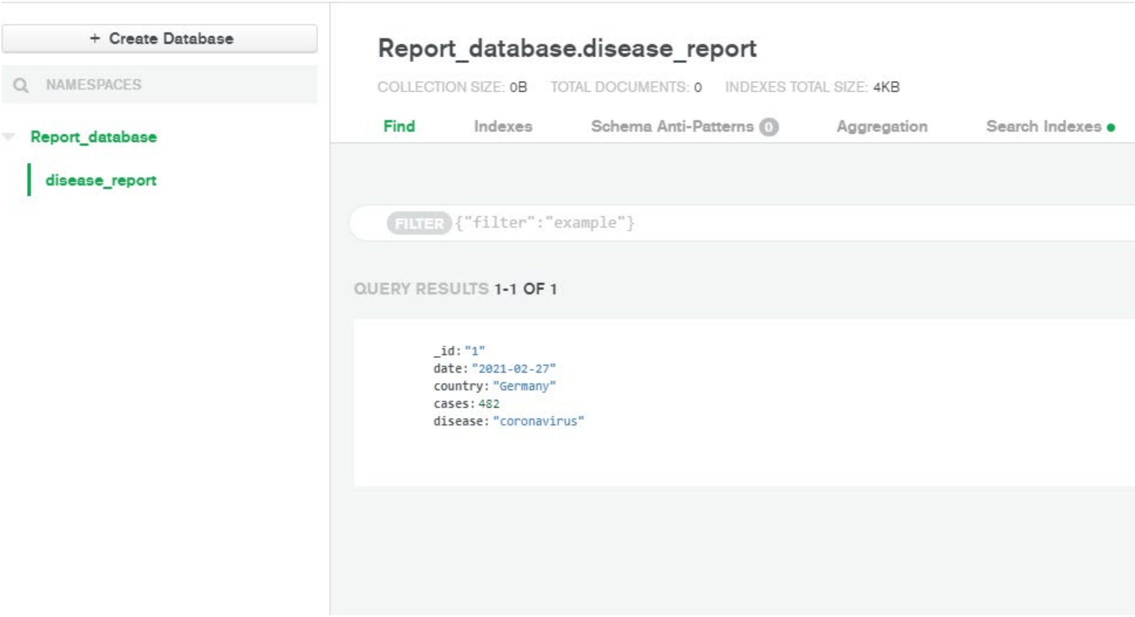
Cluster0
Version 4.4.4

CLUSTER TIER
M0 Sandbox (General)

REGION
AWS / Sydney (ap-southeast-2)

TYPE
Replica Set - 3 nodes

LINKED REALM APP
None Linked



Report_database.disease_report

COLLECTION SIZE: 0B TOTAL DOCUMENTS: 0 INDEXES TOTAL SIZE: 4KB

{"filter":"example"}

QUERY RESULTS 1-1 OF 1

```
_id: "1"
date: "2021-02-27"
country: "Germany"
cases: 482
disease: "coronavirus"
```

Justification

FRONTEND

Vue:
Vue is a Javascript framework which utilises various tools for user interface development. Our main reason for choosing Vue for the frontend development was due to its flexibility, its Model View ViewModel (MVVM) structure and its easy integration and compatibility with pre-existing templates that can be found publicly. The flexibility and availability of the Vue templates allow us to present our software in a more captivating interactive platform which helps us maximise our software's features and services for the users. The MVVM structure of Vue provides us with two-way-bind data within the views which means that any changes made to the model will be reflected in the views and vice-versa. This results in a safer and easier abstraction from models.

Bootstrap4:
Bootstrap is a free CSS and HTML based frontend framework that is used for creating dynamic websites and web applications. The main reason for choosing Bootstrap4 was due to its adaptable structure, easy and fast responsiveness as well as its customizability. The availability of pre-built codes allows for reusability and reduces code complexity which is beneficial during this limited time frame. Bootstrap4 has a large library of functions which will provide us many elements to work with when producing a visually appealing frontend.

BACKEND

MongoDB:
MongoDB is a free NoSQL document-oriented database platform that is compatible with many languages including Python. It stores data in JSON-like documents which eases linking of data to the backend for processing, storing, retrieving and updating information. Since data will be saved in JSON files, it is beneficial when working with data that is constantly being updated or new data. Furthermore, MongoDB is very easy to scale out and allows future possible expansion of the web application around the world.

Flask:
Flask is a Python web development framework and is one of the most popular frameworks for Python. This means that there are abundant resources and documentation for us to work with and its capability to be used in conjunction with MongoDB. Python will be used for building the backend where it will handle API calls for storing and retrieval of data from and to the database.