**TÓM TẮT**

Tên đề tài: Ứng dụng IoT xây dựng hệ thống giám sát và cảnh báo ngập lụt tại thành phố Đà Nẵng

Sinh viên thực hiện: Phan Đình Tùng

Số thẻ SV: 102140165, Lớp: 14T3

Nội dung:

Hệ thống bao gồm:

* Dịch vụ bản đồ số GeoServer.
* Dịch vụ tìm kiếm nhanh ElasticSearch.
* Hệ thống thiết bị thu thập dữ liệu mực nước.
* Dịch vụ giám sát và cảnh báo mực nước.
* Các dịch vụ hỗ trợ kiến trúc dịch vụ siêu nhỏ (microservice).

Các chức năng chính:

* Hiển thị bản đồ.
* Tìm kiếm, tra cứu địa điểm.
* Xem tình hình mức nước hiện tại ở các địa điểm có cảm biến.
* Đăng ký theo dõi các địa điểm có đặt cảm biến.
* Cảnh báo khi mực nước dâng cao.

………………………………………………………………………………………… ..

………………………………………………………………………………………… ..

|  |  |
| --- | --- |
| ĐẠI HỌC ĐÀ NẴNG  **TRƯỜNG ĐẠI HỌC BÁCH KHOA**  KHOA CÔNG NGHỆ THÔNG TIN | **CỘNG HÒA XÃ HÔI CHỦ NGHĨA VIỆT NAM**  Độc lập - Tự do - Hạnh phúc |

**NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

Họ tên sinh viên: Phan Đình Tùng, Số thẻ sinh viên: 102140165

Lớp:14T3 Khoa: Công nghệ thông tin, Ngành: Hệ thống nhúng

1. *Tên đề tài đồ án:*

Ứng dụng IoT xây dựng hệ thống giám sát và cảnh báo ngập lụt tại thành phố Đà Nẵng

1. *Đề tài thuộc diện:*  *Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện*
2. *Các số liệu và dữ liệu ban đầu:*

……………………………………..……………………………………………..……......……………………………………………………………………………………………..………………………………….…..………………………..………………………

1. *Nội dung các phần thuyết minh và tính toán:*

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

1. *Các bản vẽ, đồ thị ( ghi rõ các loại và kích thước bản vẽ ):*

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

…...………………………………………………………………………………………

1. *Họ tên người hướng dẫn:* …………………………………..……………………
2. *Ngày giao nhiệm vụ đồ án:*  *……../……./201…..*
3. *Ngày hoàn thành đồ án: ……../……./201…..*

|  |  |
| --- | --- |
|  | *Đà Nẵng, ngày tháng năm 201* |
| **Trưởng Bộ môn** …………………….. | **Người hướng dẫn** |

**LỜI NÓI ĐẦU**

Đến đây, kết thúc chặng đường 5 năm này, kết thúc chặng đường gắn liền với trường Đại học Bách Khoa – Đại học Đà Nẵng, với khoa Công nghệ thông tin dưới những sự giúp đỡ, hỗ trợ từ nhiều mặt từ vật chất đến tinh thần. Trong suốt thời gian từ khi bắt đầu học tập ở giảng đường đại học đến nay, em đã nhận được rất nhiều sự quan tâm giúp đỡ từ gia đình, bạn bè, quý thầy cô lẫn nhiều người xa lạ, các đồng nghiệp cơ quan cả những anh chị doanh nghiệp.

Đầu tiên em xin cám ơn chân thành thầy cô khoa Công nghệ Thông tin – Trường Đại học Bách khoa Đà Nẵng nói riêng và toàn bộ quý giáo viên, nhân viên Đại học Đà Nẵng đã tận tình giảng dạy, hỗ trợ xuyên suốt thời gian học tập ở trường.

Xin chân thành cảm ơn thầy - Ts. Phạm Công Thắng, người đã tận tình hướng dẫn, cũng như đã tạo mọi điều kiện để em có thể hoàn thành đồ án tốt nghiệp này. Nhờ những lời hướng dẫn, những đánh giá và chỉ bảo sâu sắc của thầy mà đồ án tốt nghiệp của em có thể hoàn thiện viên mãn như hôm nay. Một lần nữa em xin chân thành cảm ơn thầy.

Cảm ơn tập thể lớp 14T3 và lớp chuyên ngành Hệ thống Nhúng, cũng như gia đình đã luôn sát cánh bên cạnh và là nguồn động lực vô cùng lớn giúp cho em không ngừng nỗ lực trong 5 năm qua.

Đồ án được thực hiện trong khoảng thời gian hơn ba tháng. Bước đầu đi vào thực tế, tìm hiểu tường tận các lĩnh vực mới, với kiến thức còn hạn chế và nhiều bỡ ngỡ. Do vậy, không thể tránh khỏi những thiếu sót, em rất mong nhận được những ý kiến đóng góp quý báu của quý thầy cô để kiến thức trong các lĩnh vực này của em được hoàn thiện hơn.

Sau cùng, em xin kính chúc quý thầy cô trong Khoa Công nghệ Thông tin và thầy Phạm Công Thắng dồi dào sức khỏe, tràn đầy niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

Trân trọng!

**CAM ĐOAN**

Tôi xin cam đoan :

1. Nội dung trong đồ án này là do tôi thực hiện dưới sự hướng dẫn trực tiếp của Ts. Phạm Công Thắng.

2. Các tham khảo dùng trong luận văn đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian, địa điểm công bố.

3. Nếu có những sao chép không hợp lệ, vi phạm, tôi xin chịu hoàn toàn trách nhiệm.

……………………………………………………………………..……………………

……………………………………………………………………..……………………

……………………………………………………………………..……………………

……………………………………………………………………..……………………

……………………………………………………………………..……………………

Sinh viên thực hiện

{Chữ ký, họ và tên sinh viên}

**MỤC LỤC**

Tóm tắt

Nhiệm vụ đồ án

Lời nói đầu và cảm ơn i

Lời cam đoan liêm chính học thuật ii

Mục lục iii

Danh sách các bảng biểu, hình vẽ và sơ đồ v

Danh sách các cụm từ viết tắt vi

Trang

**Chương 1 ...................................................**

**1.1 ................................................................** 1

1.1.1

1.1.2

**1.2**  7

1.2.1

1.2.2….................................

**1.3 ........................................** 22

**Chương 2 .......................................................**

**2.1**…..

2.1.1….

2.1.2…..

**2.2**

…..

**Chương 3**  50

**3.1**…..

3.1.1….

3.1.2….

**3.2** ………………

**KẾT LUẬN** 68

**TÀI LIỆU THAM KHẢO** 70

**PHỤ LỤC**

**DANH SÁCH CÁC BẢNG, HÌNH VẼ**

BẢNG 1.1 {size 13}..........................................................................................................

BẢNG 1.2 …….................................................................................................................

BẢNG 1.3 …….................................................................................................................

…….……..........................................................................................................................

HÌNH 1.1 ...........................................................................................................................

HÌNH 1.2 ...........................................................................................................................

HÌNH 1.3 ...........................................................................................................................

…….……..........................................................................................................................

***Ghi chú:***

* Mỗi bảng, hình vẽ/ sơ đồ phải được đánh số và có tên;
* Đánh số bảng và đánh số hình vẽ/ sơ đồ riêng. Quy luật đánh số như sau:
  + Chữ số thứ nhất chỉ tên chương;
  + Chữ số thứ hai chỉ thứ tự bảng biểu, sơ đồ, hình,…trong mỗi chương.

**DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT**

|  |  |
| --- | --- |
| Từ viết tắt | Diễn giải |
| GIS | Geographic Information Systems |
| API | Application Programming Interface |
| URL | Uniform Resource Locator |
| ES | Elasticsearch |

*Ghi chú:*

* Ký hiệu: mỗi mục ký hiệu gồm ký hiệu và phần tên gọi, diễn giải ký hiệu.
* Cụm từ viết viết tắt là các chữ cái và các ký hiệu thay chữ được viết liền nhau, để thay cho một cụm từ có nghĩa, thường được lặp nhiều lần trong đồ án.

MỞ ĐẦU

1. Lý do chọn đề tài

Ngày nay, đi đôi cùng với biến đổi khí hậu, nhiều thiên tai ập đến không báo trước gây nên nhiều hậu quả nghiêm trọng. Ví dụ:

* Nhiều trận lũ lụt diễn ra rất bất ngờ mà không hề báo trước trên toàn quốc, điển hình như trận lụt cuối năm 2018 tại thành phố Đà Nẵng
* Nhiều đường ở các thành phố lớn bị ngập sâu sau các trận mưa lớn hay triều cường, xuất hiện thường xuyên tại thành phố Hồ Chí Minh



Đây là vấn đề nhức nhối của Việt Nam trong quá trình Trái Đất đang dần nóng lên. Khó có thể đẩy lùi trong thời gian ngắn dù cho con người cố gắng không để môi trường ô nhiễm hơn. Do đó một giải pháp giúp người dân tại các thành phố lớn nói riêng và toàn thế giới nói chung có thể sống chung với các tình trạng như triều cường hay nước ngập cục bộ rất cần thiết trong giai đoạn này.

1. Mục đích và ý nghĩa của đề tài
   1. Mục đích

Mục đích của đề tài này hướng đến là:

* Hiển thị khách quan tình hình mức nước hiện tại ở các địa điểm có đặt cảm biến
* Cảnh báo tức thời khi có dữ liệu mực mước vượt quá mức cho phép, giúp sơ tán hoặc phong tỏa các tài sản nến bị ảnh hưởng do mước nước cao.
* Hỗ trợ quá trình điều tiết nguồn nước, giúp giảm đến tối thiểu các tình trạng ngập lụt cục bộ.
  1. Ý nghĩa

Hệ thống cho phép người dùng sử dụng bản đồ như các công cụ bản đồ khác (như Google Maps hay Open Street Map). Trong đó gồm cả hệ thống bản đồ lẫn hệ thống tìm kiếm trên bản đồ.

Hệ thống cung cấp giải pháp giám sát mực nước tại các vị trí có đặt cảm biến bằng bản đồ dễ hiểu và dễ hình dung.

Cảnh báo tức thời đến người dùng qua email nếu mực nước tại các địa điểm vượt quá ngưỡng quy định.

Hệ thống đi vào hoạt động sẽ hỗ trợ trực tiếp đến người dùng sống hoặc sinh hoạt quanh các địa điểm có lượng mưa cao hoặc triều cương thường xuyên xảy ra.

* 1. Công cụ thực hiện

Công cụ bản đồ số: Geoserver.

Các hệ cơ sở dữ liệu: PostgresSQL, MongoDB, Firebase, Elasticsearch.

Các framework hỗ trợ backend: Nodejs, Java Spring Boot.

Các thư viện hỗ trợ frontend: Jquery, Openlayer, HTML, CSS, Bootstrapt.

Hệ thống thiết bị và cảm biến: ESP8266 V12 và cảm biến siêu âm. Ngoài ra còn một số thiết bị hỗ trợ khác.

1. Bố cục trình bày

{Font: Time New Roman; thường; cỡ chữ: 13; dãn dòng: 1,3; căn lề: justified}

{Trong phần này, cần trình bày về: Mục đích thực hiện đề tài, mục tiêu đề tài, phạm vi và đối tượng nghiên cứu, phương pháp nghiên cứu, cấu trúc của đồ án tốt nghiệp}

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

# CƠ SỞ LÝ THUYẾT

1.1. Tổng **quan về hệ thống thông tin địa lý GIS và GeoServ**er

### 1.1.1. Hệ Thống thông tin địa lý GIS

1.1.1.1. Giới thiệu

Hệ thống thông tin địa lý GIS [1] (Geographic Information Systems) là một công cụ tập hợp những quy trình dựa trên máy tính để lập bản đồ, lưu trữ và thao tác dữ liệu địa lý, phân tích các sự vật hiện tượng thực trên trái đất, dự đoán tác động và hoạch định chiến lược. Thuật ngữ này được biết đến từ những năm 60 của thế kỉ 20 và Giáo sư Roger Tomlinson được cả thế giới công nhận là cha đẻ của GIS.

Đơn giản hơn, GIS là một tập hợp có tổ chức của phần cứng, phần mềm, dữ liệu địa lý và con người đuợc thiết kế để thu nhận, lưu trữ, cập nhật, thao tác, phân tích và hiển thị tất cả các dạng thông tin địa lý.

1.1.1.2. Các thành phần của GIS

Các thành phần của hệ thống GIS rất nhiều, được chia làm các phần sau:

* Con người: Đối tượng tạo và sử dụng hệ thống
* Dữ liệu: Dữ liệu bản đồ
* Phương pháp phân tích: là các hoạt động, cách xử lý dữ liệu theo mỗi bài toán.
* Phần mềm: thực hiện những hoạt động của GIS.
* Phần cứng: được dùng để xử lý, lưu trữ hiển thị dữ liệu không gian dạng số.

1.1.1.3. Chức năng của GIS

Gồm những chức năng cơ bản sau:

* Nhập dữ liệu: GIS cung cấp các phương pháp để nhập cả hai loại dữ liệu: dữ liệu không gian và phi không gian.
* Hiển thị dữ liệu: hiển thị kết quả phân tích thông tin không gian và phi không gian có thể được hiển thị trong hai dạng: trên màn hìnhvà trên giấy in.
* Thao tác và phân tích dữ liệu: chuẩn bị dữ liệu để truy xuất và sử dụng dễ dàng, và phân tích dữ liệu nhằm trả lời những câu hỏi hoặc tìm những giải pháp cho những vấn đề khác nhau.
* Quản lý dữ liệu: chức năng này liên quan tới cơ sở dữ liệu, nó bao gồm những chức năng lưu trữ, xóa, phục hồi.

1.1.2. Giới thiệu dịch vụ bản đồ GeoServer

GeoServer [2] là phầm mềm chứa một số phương pháp phân tích, là một máy chủ mã nguồn mở với mục đích kết nối những thông tin địa lý có sẵn tới các Geoweb (trang Web địa lý) sử dụng chuẩn mở. Được bắt đầu bởi một tổ chức phi lợi nhuận có tên The Open Planning Project (TOPP), nhằm mục đích hỗ trợ việc xử lý thông tin không gian địa lý với chất lượng cao, đơn giản trong sử dụng, là phần mềm mã nguồn mở nhằm cung cấp và chia sẻ dữ liệu.

GeoServer được viết bằng ngôn ngữ Java, cho phép người sử dụng chia sẻ và chỉnh sử dữ liệu không gian địa lý (geospatial data). Đây là một dự án mang tính cộng đồng, GeoServer được phát triển, kiểm thử và hỗ trợ bởi nhiều nhóm đối tượng và tổ chức khác nhau trên toàn thế giới. GeoServer là sự phối hợp các chuẩn hoạt động của Open Geospatial Consortium (OGC), Dịch vụ bản đồ (WMS-Web Map Service), Web Feature Service (WFS).

Geoserver cho phép người dùng hiển thị thông tin không gian của mình về thế giới. Cung cấp chuẩn Dịch vụ bản đồ (Web Map Service - WMS), GeoServer có thể tạo bản đồ và xuất ra nhiều định dạng. OpenLayers, một thư viện bản đồ hoàn toàn miễn phí, được tích hợp cùng GeoServer giúp cho công việc tạo bản đồ trở nên đơn giản hơn bao giờ hết.

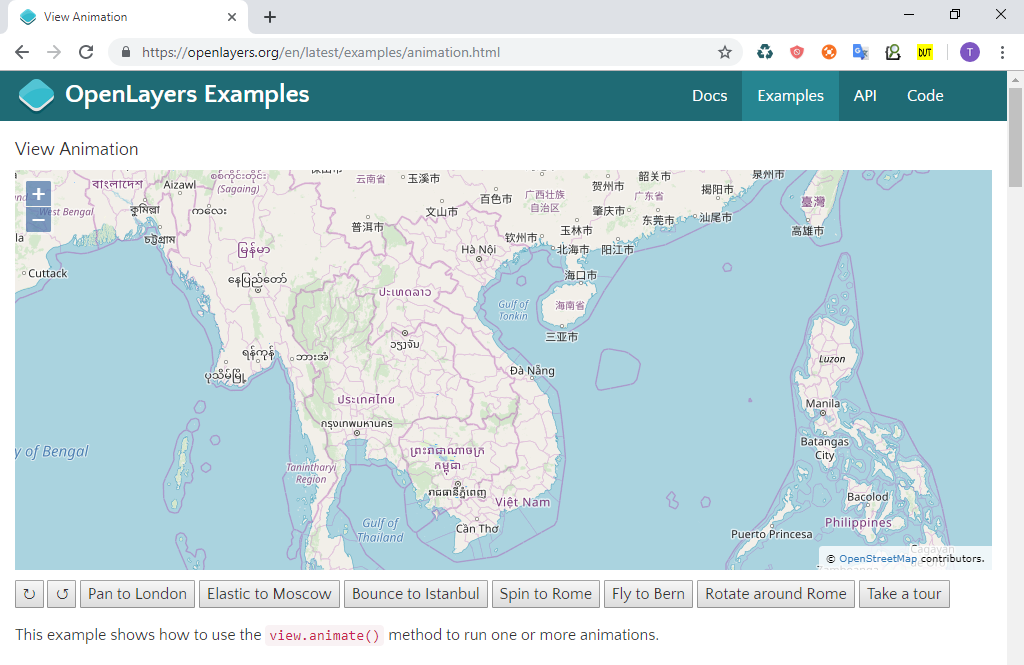
GeoServer hỗ trợ rất nhiều style bản đồ. Tương thích với chuẩn Web Feature Service (WFS), GeoServer cho phép chia sẻ và chỉnh sửa dữ liệu đang được dùng để hiển thị bản đồ.

1.1.3. OpenLayers

OpenLayers [3] là một dự án của tổ chức OGC. OpenLayers là một thư viện điện tử mã nguồn mở JavaScrip rất mạnh giúp nhúng bản đồ động lên trang web bất kỳ. Nó cung cấp một API để xây dựng nhiều ứng dụng dựa trên web địa lý tương tự như Google Maps và Bing Maps. OpenLayers có thể lấy bản đồ từ nhiều loại nguồn khác nhau và cung cấp một giao diện tương tác đẹp, phong phú cho người dùng.

OpenLayers được tạo ra bởi MetaCarta sau khi tổ chức hội nghị O'Reilly vào ngày 29-30 tháng 6 - 2005, và phát hành như là một phần mềm mã nguồn mở trước khi tổ chức hội nghị lần 2 ngày 13-14 tháng 6 - 2006, bởi MetaCarta Labs.

Tất cả các thông tin bao gồm tài liệu, các ví dụ, API và mã nguồn được cung cấp sẵn tại đường dẫn trang chủ của OpenLayers ([đây](http://openlayers.org)).

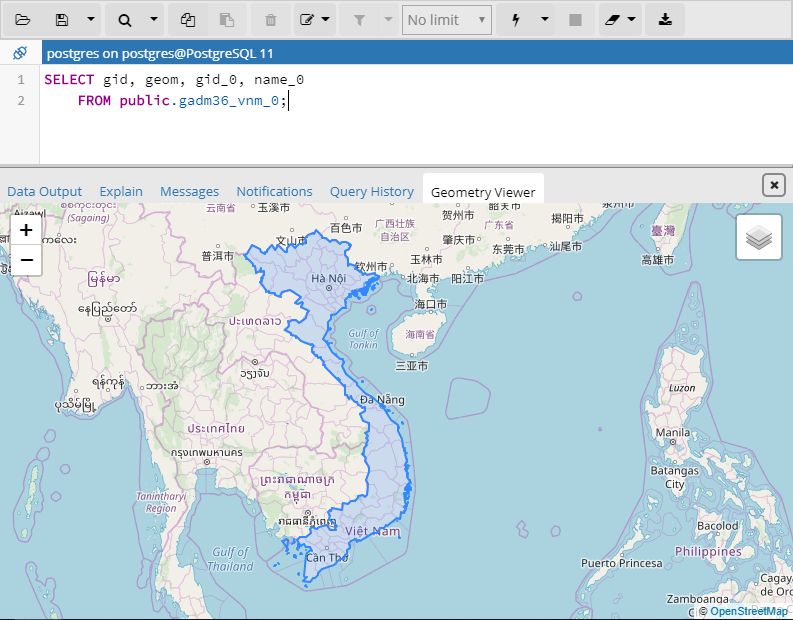


Hình 1.1: Một trang ví dụ trên OpenLayers

1.1.4. Cơ sở dữ liệu không gian

1.1.4.1. Giới thiệu

Cơ sở dữ liệu không gian là một hệ thống cơ sở dữ liệu cung cấp các kiểu dữ liệu không gian trong mô hình dữ liệu và các ngôn ngữ truy vấn. Các kiểu dữ liệu không gian như: Point, Line, Polygon... Cơ sở dữ liệu cung cấp mô hình trừ tượng cơ bản cho cấu trúc của thực thể hình học trong không gian cũng như mối quan hệ giữa chúng như: giao nhau, thuộc nhau... Thông qua các phần mềm mô phỏng, dữ liệu không gian chứa trong cơ sở dữ liệu không gian sẽ được đồ họa thành bản đồ.



Hình 1.2: Thực thi lệnh truy vấn trên PostgreSQL

1.1.4.2. Lựa chọn cơ sở dữ liệu không gian

Qua quá trình phân tích, so sánh, sau cũng cơ sở dữ liệu PostgreSQL được lựa chọn làm cơ sở dữ liệu không gian, lưu trữ dữ liệu bản đồ sử dụng trong phạm vi đồ án này. PostgreSQL có hỗ trợ PostGIS - một chương trình phần mềm nguồn mở bổ sung hỗ trợ cho các đối tượng địa lý vào cơ sở dữ liệu quan hệ đối tượng. PostgreSQL được đánh giá là hiệu quả trong khi chi phí triển khai thấp (có thể là không tốn kém), cách lưu trữ đơn giản, bảo mật (nhờ tính mã nguồn mở), tương thích cao.

Bên cạnh đó, dữ liệu bản đồ rất lớn, cho cơ sở dữ liệu không gian phải hỗ trợ kích thước dữ liệu phải lớn. Cơ sở dữ liệu PostGreSQL hỗ trợ dữ liệu lên đến kích thước Terabyte, phù hợp để lưu dữ liệu bản đồ.

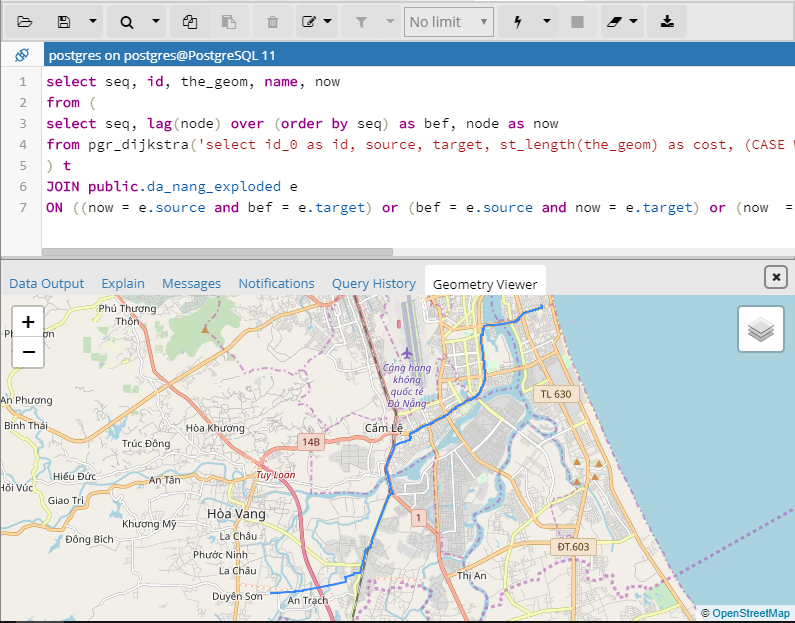
Bảng 1.1: Các giới hạn lưu trữ của PostgreSQL

|  |  |
| --- | --- |
| Khả năng | Giá trị |
| Kích thước của CSDL | Không giới hạn |
| Kích thước của 1 bảng dữ liệu | 32 TB |
| Kích thước của 1 dòng dữ liệu | 1.6 TB |
| Kích thước của 1 trường dữ liệu | 1 GB |
| Số lượng dòng trong 1 bảng | Không giới hạn |
| Số lượng cột trong 1 bảng | 250 - 1600 tùy theo kiểu dữ liệu của các cột |

Một số ưu điểm của PostgreSQL mà ta có thể kể đến như:

* Hoạt động được trên nhiều hệ điều hành như: Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), and Windows
* Lưu trữ (dạng nhị phân) các đối tượng có dữ liệu lớn như hình ảnh, âm thanh, hoặc video.
* Hỗ trợ các ngôn ngữ như C/C++, Java, .Net, Perl, Python, Ruby, Tcl, JS,…
* Tạo index giúp quá trình truy vấn đạt hiệu quả cao hơn
* Hỗ trợ các chức năng tìm kiếm tổng quát hóa có thể “gắn” các chỉ mục vào giúp quá trình tìm kiếm được thực hiện trên phạm vi rộng hơn như: compound, unique, partial, and functional indexes.

Ngoài ra PostGIS còn hỗ trợ nhiều tính năng, đặc biệt có tính năng tìm đường ngắn nhất sử dụng thuật toán Dijkstra, cho kết quả nhanh và chính xác.



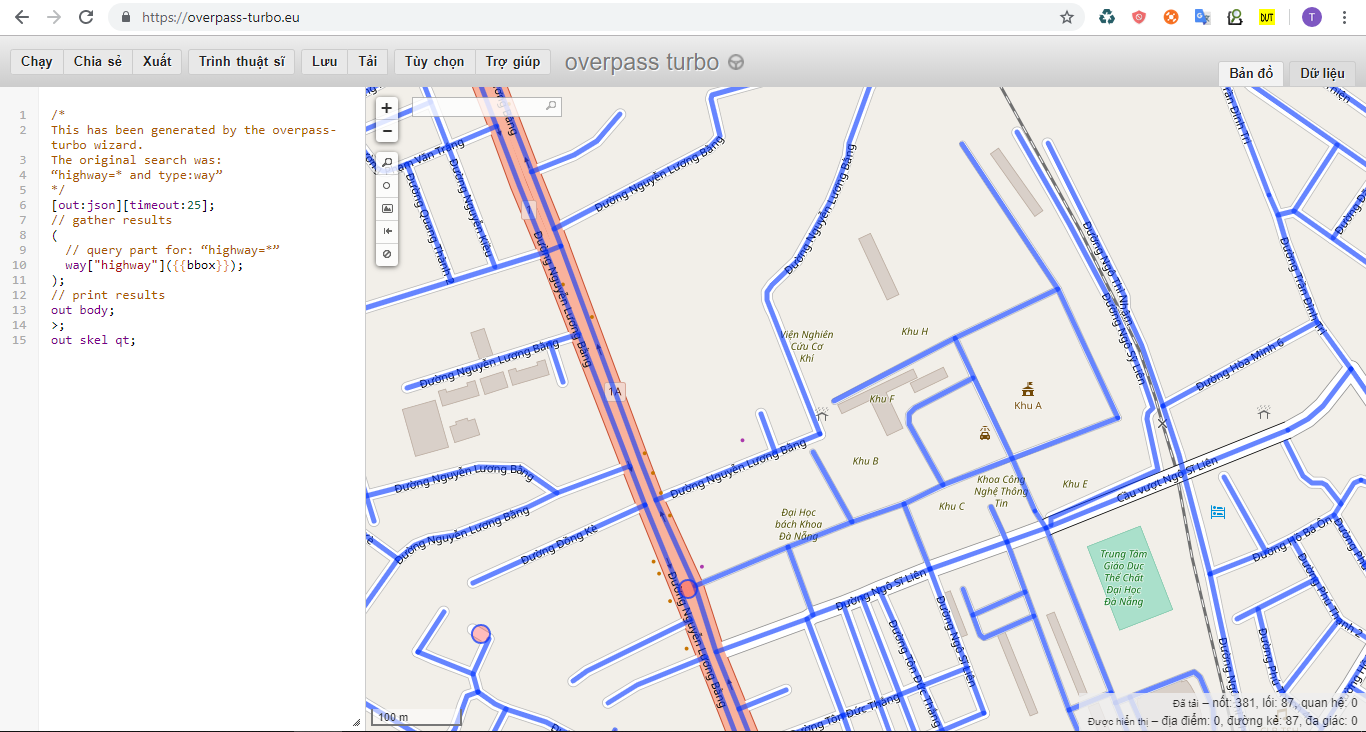
Hình 1.3: Ví dụ truy vấn đường đi ngắn nhất giữa 2 điểm trên bản đồ

1.1.5. Dữ liệu bản đồ

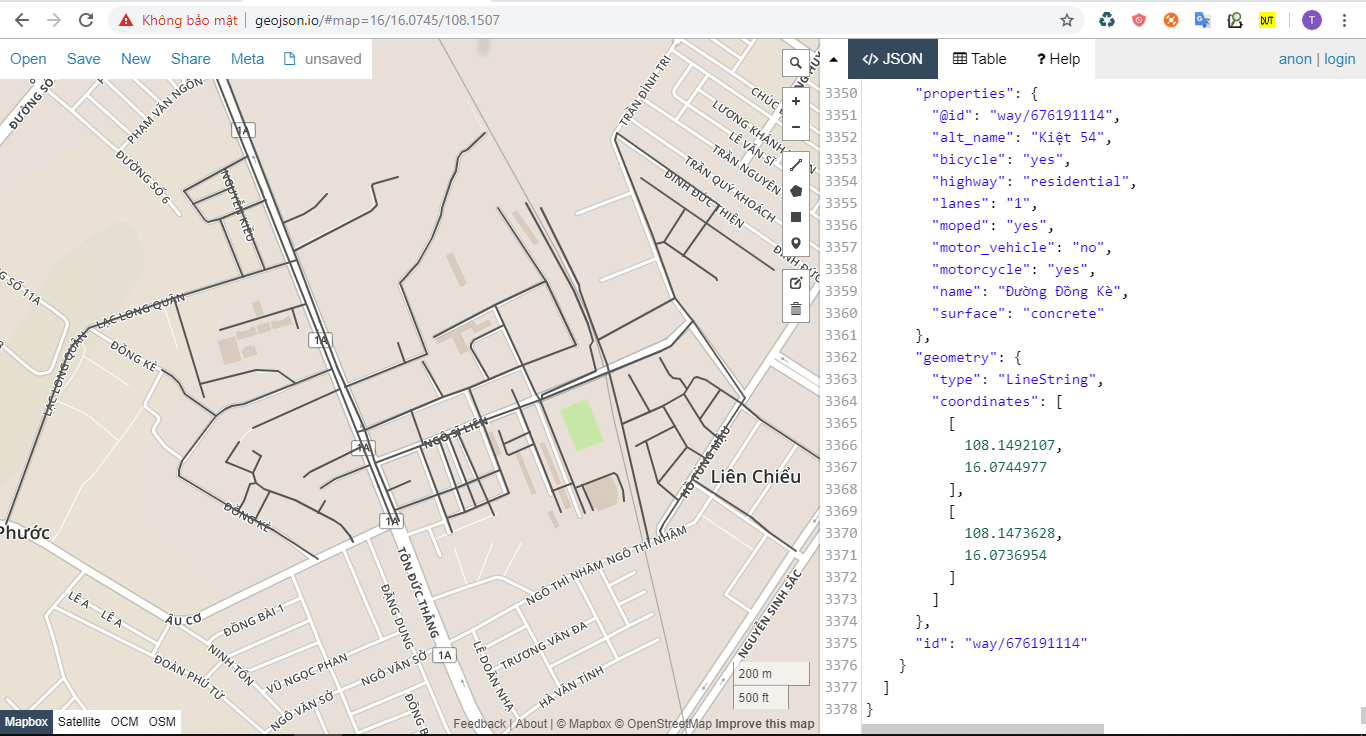
Trong phạm vi dự án, dữ liệu bản đồ có 2 nguồn:

* Website Global Administrative Areas [4]: đây là trang web hỗ trợ tải miễn phí dữ liệu bản đồ hành chính các quốc gia.
* OpenStreetMap [5]: đây là trang web hỗ trợ tải miễn phí dữ liệu bản đồ.

Dữ liệu từ Global Administrative Areas được xuất ở đinh dạng shapefile, có thể nhập vào cơ sở dữ liệu và dùng ngay. Riêng dữ liệu từ OpenStreetMap được thu thập qua Overpass-Turbo [6]. Dữ liệu bản đồ được chỉnh sửa trên GeoJSON.io [7]. Sau đó xuất ra tệp định dạng shapefile(\*.shp).

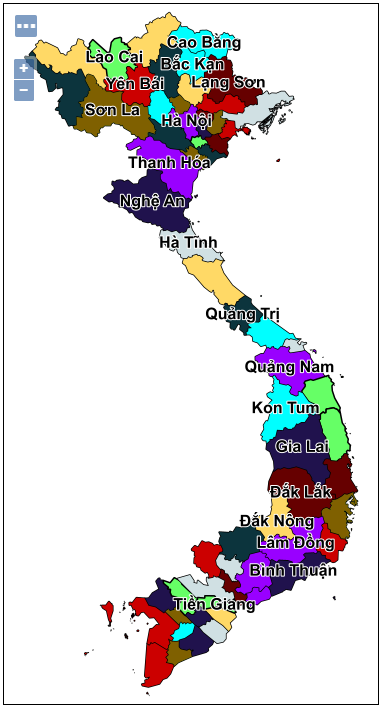


Hình 1.4: Giao diện trích dữ liệu từ OpenStreetMap trên Overpass-Turbo



Hình 1.5: Giao diện xem dữ liệu vừa trích xuất từ Overpass-turbo trên GeoJSON.io

Tất cả dữ liệu dạng shapefile được nhập vào cơ sở dữ liệu PostGreSQL qua công cụ PostGIS 2.0 Shapefile and DBF Loader Exporter.

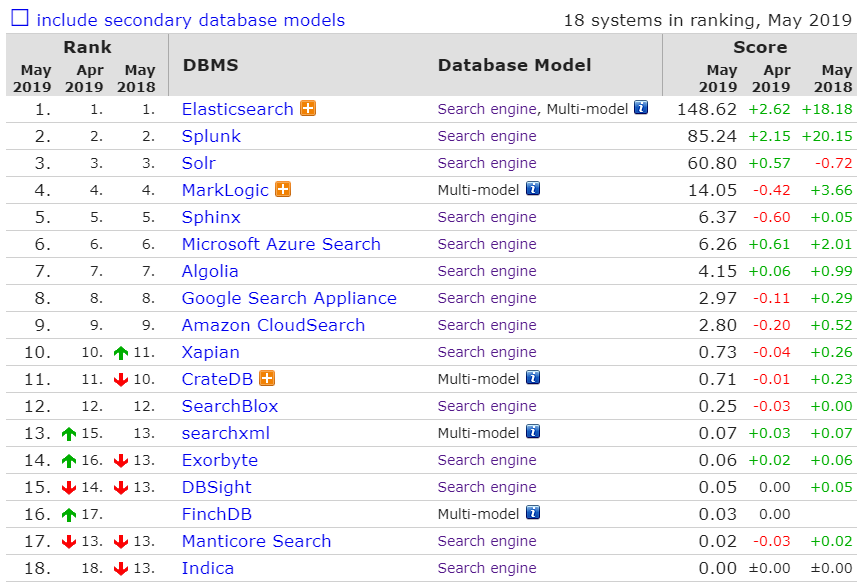


Hình 1.6: Hình ảnh bản đồ khi kết hợp GeoServer, OpenLayers và PostgreSQL

## 1.2. ElasticSearch

1.2.1. Giới thiệu

Elasticsearch (ES) [8] là một công cụ tìm kiếm dựa trên phần mềm Lucene. Nó cung cấp một bộ máy tìm kiếm dạng phân tán, có đầy đủ công cụ với một giao diện web HTTP có hỗ trợ dữ liệu JSON. ES được phát triển bằng Java và được phát hành dạng nguồn mở theo giấy phép Apache. ES là một công cụ tìm kiếm phổ biến nhất [9]



Hình 1.7: Xếp hạng các hệ thống quản lý cơ sở dữ liệu tìm kiếm theo mức độ phổ biến của chúng

1.2.2. Đặc điểm

ES có một số đặc điểm như sau:

* ES là một search engine.
* ES được kế thừa từ Lucene Apache
* ES thực chất hoạt động như 1 web server, có khả năng tìm kiếm nhanh chóng (near realtime) thông qua giao thức RESTful
* ES có khả năng phân tích và thống kê dữ liệu
* ES chạy trên server riêng và đồng thời giao tiếp thông qua RESTful

Các thành phần bên trong của ES:

* Node là trung tâm hoạt động của ES. Lưu trữ toàn bộ dữ liệu để có thể thực hiện công việc lưu trữ và tìm kiếm.
* Cluster: tập hợp các nodes hoạt động cùng với nhau, chia sẽ cùng thuộc tính cluster.name.
* Index: tương tự như database của SQL hay MongoDB thì Index là một tập các type dữ liệu.
* Type: cũng như SQL có table và MongoDB có collection thì ES định nghĩa nó là type, đại diện cho tập dữ liệu có chung tính chất, đặc trưng cho đối tượng nó lưu trữ.
* Shard là đối tượng của Lucene, là tập con các documents của 1 Index. Một Index có thể được chia thành nhiều shard. Shard hỗ trợ trong việc tìm kiếm, nhiều shard giúp tìm kiếm nhanh hơn đồng thời nhiều shard thì dữ liệu sẽ được an toàn hơn.

1.2.3. Nhập dữ liệu

Dữ liệu tìm kiếm được xem là phần quan trọng nhất của hoạt động tìm kiếm, cũng như bản đồ có dữ liệu bản đồ là quần không thể thiếu vậy. Cũng vì chính lý do đó, dữ liệu tìm kiếm bản đồ, chính là dữ liệu bản đồ. Hoạt động tìm kiếm ở đây chính là tìm đường, tìm các địa điểm, tìm phường/xã/thị trấn, tìm quận/huyện, tìm tỉnh/thành phố.

Dữ liệu bản đồ và dữ liêu tìm kiếm bản đồ được xem là gần như nhau. Do đó, đữ liệu tìm kiếm được nhập trực tiếp từ dữ liệu bản đồ. Công cụ được lựa chọn để thực hiện việc nhập dữ liệu này là Logstash.

1.3. ESP8266

1.3.1. Giới thiệu

ESP8266 [10] là một mạch vi điều khiển có thể giúp chúng ta điều khiển các thiết bị điện tử.Thêm vào đó nó được tích hợp wi-fi 2.4GHz có thể dùng cho lập trình. Có thể lập trình bằng LUA, Python hoặc C (qua Arduino IDE).

Cấu hình ESP8266 V12 :

* WiFi: 2.4 GHz hỗ trợ chuẩn 802.11 b/g/n
* Điện áp hoạt động: 3.3V
* Điện áp vào: 5V thông qua cổng USB
* Số chân I/O: 11 (tất cả các chân I/O đều có Interrupt/PWM/I2C/One-wire, trừ chân D0)
* Bộ nhớ Flash: 4MB
* Hỗ trợ bảo mật: WPA/WPA2, tích hợp giao thức TCP/IP

1.3.2. Ứng dụng

ESP8266 dần trở nên phổ biến khi có thể lập trình bằng C qua Arduino IDE, được dùng để giảng dạy, nghiên cứu từ độ tuổi nhỏ, đến ứng dụng xây dựng các thiết bị IoT nhờ kết nối wifi đơn giản. Như dự báo thời tiết, giám sát nhiệt độ độ ẩm, thậm chí là máy chơi game mini hay điều khiển thiết bị điện qua internet.



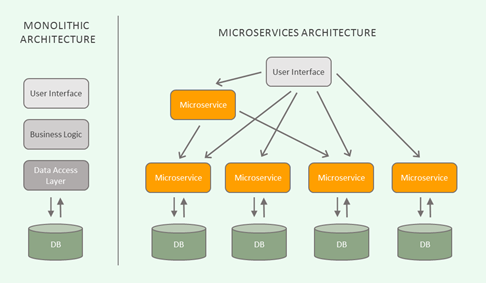
Hình 1.8: Một ứng dụng sở dụng vi điều khiển ESP8266 [11]

1.4. Kiến trúc dịch vụ siêu nhỏ Microservice

1.4.1. Giới thiệu

Microservice là kiến trúc ứng dụng - xây dựng một ứng dụng mà ứng dụng này là tổng hợp của nhiều services nhỏ và độc lập có thể chạy riêng biệt, phát triển và triển khai độc lập. Các service nhỏ có thể được phát triển riêng, độc lập với nhau.

1.4.2. Đặc điểm



Hình 1.9: Sự tương quan giữa kiến trúc monolithic và microservices

Qua hình ảnh ta thấy được sự khác nhau giữa kiến trúc monolithic truyền thống và kiến trúc microservice hiện nay. Nhờ sự khác biệt này nên microservice có một số ưu nhược điểm như sau.

Ưu điểm:

* Giảm sự gia tăng các liên kết/ràng buộc rối rắm trong quá trính phát triển/mở rộng hệ thống
* Chia nhỏ ứng dụng lớn thành các dịch vụ nhỏ, quản lý, bảo trì, nâng cấp, dễ thay đổi công nghệ
* Tạo tách bạch rõ ràng các chức năng thành từng khối, dễ quản lý chức năng lẫn quản lý dữ liệu
* Dễ phát triển, nhanh, dễ kiểm thử

Nhược điểm:

* Dễ tạo ra các dịch vụ vụn vặt, khó kiểm soát, dữ liệu phân tán quá mức cần thiết.
* Sinh ra sự cố kết nối chậm, vì các dịch vụ hoạt động riêng rẻ
* Dữ liệu khó được đồng bộ
* Sinh ra các chuỗi dịch vụ phụ thuộc nhau. A gọi B, B gọi C,...
* Tăng sự kết nối giữa các dịch vụ trong hệ thống, khó kiểm soát các kết nối
* Cần có cơ chế tự cập nhật địa chỉ, cổng của dịch vụ, mô tả phiên bản của mỗi dịch vụ khi số dịch vụ nhiều, hay thay đổi hoặc mở rộng.

1.4.3. Ứng dụng

Mỗi dịch vụ sẽ được phát triển riêng biệt, các dữ liệu cần đồng bộ sẽ được giao tiếp qua các API. Như vậy, khi phát triển sẽ phát triển mỗi dịch vụ riêng biệt, khi hoạt động sẽ hoạt động các dịch vụ riêng biệt. Một dịch vụ bị mất kết nối thì toàn bộ dịch vụ sẽ hoạt động không đúng nhưng toàn dịch vụ vẫn hoạt động, không như dịch vụ theo kiến trúc đơn khối, khi không hoạt động sẽ làm dịch vụ dừng hoàn toàn.

Việc bảo trì các dịch vụ cũng trở nên đơn giản hơn, vì khi thông qua kiến trúc microservice, các dịch vụ có thể hoạt động nhiều instance (dịch vụ hoạt động cùng lúc nhiều phiên bản). Khi một phiên bản dừng hoạt động, API Gateway sẽ định tuyến sang phiên bản khác. Như vậy, các dịch vụ dùng hoạt động có thể được sữa lỗi và khởi động lại trong khi hệ thống vẫn hoạt động bình thường.

Khi mở rộng, hệ thống API Gateway sẽ hỗ trợ định tuyến, nếu một dịch vụ bình thường bị quá tải, thì dịch vụ sẽ hoạt động không ổn định. Ngược lại, trong kiến trúc các dịch vụ siêu nhỏ, API Gateway sẽ cân bằng tải, đảm bảo các dịch vụ hoạt động đồng đều và ổn định.

1.5. NodeJS

1.5.1. Giới thiệu

Node.js [12] là một hệ thống phần mềm được thiết kế để viết các ứng dụng internet có khả năng mở rộng, đặc biệt là máy chủ web. Chương trình được viết bằng JavaScript, sử dụng kỹ thuật điều khiển theo sự kiện, nhập/xuất không đồng bộ để tối tiểu tổng chi phí và tối đại khả năng mở rộng. Node.js bao gồm có V8 JavaScript engine của Google, libUV, và vài thư viện khác.

Node.js được tạo bởi Ryan Dahl từ năm 2009, và phát triển dưới sự bảo trợ của Joyent.

Trong quy mô dự án này, NodeJS dùng để xây dựng máy chủ web, có chức năng render trang web bản đồ, các chức năng giám sát và cảnh báo của hệ thống.

1.5.2. NodeJS và MongoDB

MongoDB là một NoSQL, hiểu suất cao rất nổi tiếng, nó được xây dựng với ý tưởng dựa trên cấu trúc document.

Trong MongoDB, dữ liệu được lưu trữ như một document, một tập của các cặp key-value. Bạn có thể định nghĩa nhiều database trong MongoDB và mỗi database có nhiều collections, những collections này đơn giản là tập của các documents được lưu trữ dạng cặp key-value. Cấu trúc dữ liệu định nghĩa document được gọi là BSON(Binary JSON). BSON là một dạng nhị phân của JSON và cũng hỗ trợ những kiểu dữ liệu. Chính vì cấu trúc này tạo nên sự linh hoạt, cho nên khi kết hợp với NodeJS sự linh hoạt đó được đẩy lên cao, kết hợp với hiệu suất cao cho nên MongoDB được chọn làm cơ sở dữ liệu quản lý các thiệt bị cảm biến.

1.5.3. NodeJS và Firebase

Ngoài MongoDB, Firebase cũng là một cơ sở dữ liệu, được gọi là cơ sở dữ liệu thời gian thực, là một nền tảng di động giúp bạn nhanh chóng phát triển các ứng dụng chất lượng cao, phát triển ứng dụng cho người dùng quy mô lớn và kiếm được nhiều tiền hơn. Trong phạm vi đồ án, Firebase dùng để làm cơ sở dữ liệu chứa dữ liệu của các cảm biến.

Firebase dễ dàng kết nối cũng như truyền dữ liệu từ ESP8266 cũng như dễ dàng sử dụng với NodeJS. Nhưng khi dùng Firebase, sẽ tăng chi phí dịch vụ khi phạm vi sản phẩm mở rộng. Sự phụ thuộc lớn với Google cho nên Firebase không được chọn làm cơ sở dữ liệu chính cho hệ thống. Giới hạn ở mức lưu trữ dữ liệu thời gian thực từ các cảm biến.

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

2.1. Xác định yêu cầu

Hệ thống cung cấp giao diện web, người dùng có thể dễ dàng tiếp cận, tìm kiếm các cảm biến có trên hệ thống, giám sát mực nước và đưa ra lich trình làm việc phì hợp trong điều kiện mực nước trên các cung đường không an toàn.

2.1.1. Yêu cầu chức năng phía người dùng

Các chức năng cơ bản phía người dùng yêu cầu như sau:

* Hiển thị danh sách các cảm biến một cách trực quan, dễ sử dụng.
* Tìm kiếm dễ dàng và nhanh các địa điểm hoặc các vị trí có cảm biến.
* Biết được các mực nước tại các địa điểm quan tâm.
* Nhận cảnh báo khi mực nước tại các điểm quan tâm vượt quá ngưỡng an toàn.
* Đưa ra đường đi tốt nhất tránh các điểm có mực nước vượt ngưỡng an toàn.
* Các cảm biến hoạt động ổn định, kết quả chính xác, dễ dàng lắp đặt, giá thành rẻ, tiết kiệm năng lượng.

2.1.2. Yêu cầu chức năng phía phát triển và quản trị hệ thống

Một số yêu cầu từ phía phát triển và quản lý hệ thống

* Dễ dàng phát triển.
* Dễ bảo trì.
* Dễ mở rộng quy mô.

2.1.3. Yêu cầu phi chức năng

2.1.3.1. Hiệu suất

Hệ thống phải đạt độ ổn định tương đối, đáp ứng tốc độ tải trang nhất định với lượng truy cập không lớn. Hệ thống đảm bảo được sự tối ưu trong việc tải các thành phần của trang, không gây lãng phí tài nguyên.

2.1.3.2. Yêu cầu an toàn

Hệ thống hoạt động độc lập, không phụ thuộc vào bất kỳ phần mềm nào, không ảnh hưởng đến các phần mềm khác.

2.1.3.3. Yêu cầu bảo mật

Hệ thống đảm bảo các thông tin người dùng phải được bảo mật nhất, không sử dụng các thư viện Javascript không rõ nguồn gốc.

Hệ thống có khả năng tích hợp các biện pháp bảo mật một cách dễ dàng.

2.1.3.4. Yêu cầu chất lượng

Hệ thống phải đảm bảo chất lượng tốt, hoạt động tốt, ít các lỗi cơ bản. Tiết kiệm được dung lượng web, giảm băng thông. Chỉ sử dụng các thư viện cần thiết và sử dụng đúng chỗ.

2.2. Phân tích và đặc tả yêu cầu

### 2.1.1. Danh sách tác nhân

Bảng dưới đây sẽ mô tả tất cả các tác nhân tham gia vào “Ứng dụng IoT xây dựng hệ thống giám sát và cảnh báo ngập lụt tại thành phố Đà Nẵng”. Mỗi tác nhân sẽ có một vai trò khác nhau trong hệ thống, hiện tại chúng ta có các tác nhân : Admin, Client

Bảng 2.1: Danh sách tác nhân

|  |  |
| --- | --- |
| Actor | Description |
| Admin | Đăng ký thêm thiết bị mới. |
| Client | Xem bản đồ  Xem mực nước các cảm biến  Đăng ký nhận thông báo  Tìm đường ngắn nhất trên bản đồ |

2.1.2. Biểu đồ của kiến trúc microservice

Hệ thống sẽ gồm các dịch vụ sau:

* IoT service: Dịch vụ chịu trách nhiệm giám sát và cảnh báo.
* Map service: Dịch vụ chịu trách nhiệm kết xuất bản đồ web.
* Search service: Dịch vụ chịu trách nhiệm hỗ trợ tìm kiếm nhanh.
* Web service: Dịch vụ kết xuất front end về browser.
* API gateway: Dịch vụ hỗ trợ định tuyến đến các dịch vụ khác.
* Config service: Dịch vụ cung cấp cáu hình cho các dịch vụ khác.
* Discovery service: Dịch vụ quản lý các phiên bản của các dịch vụ có đăng ký tại nó.



Hình 2.1: Kiến trúc microservice của hệ thống

2.1.3. Biểu đồ ca sử dụng tổng quát

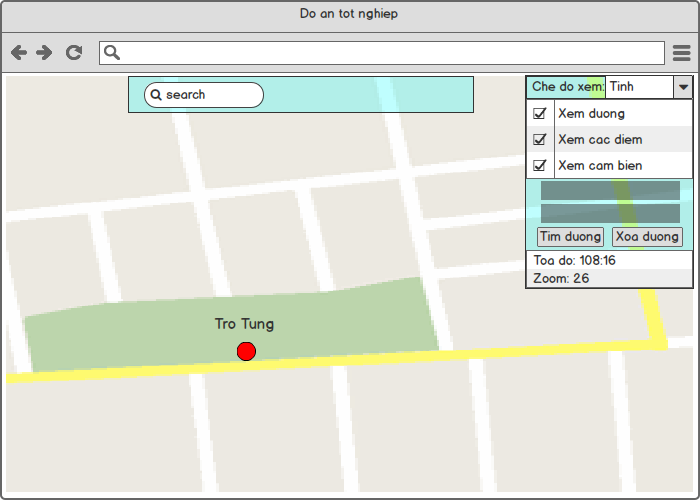


Hình 2.2: Biểu đồ ca sử dụng tổng quát

2.3. Đặc tả chức năng

2.3.1. Client

2.3.1.1. Xem bản đồ

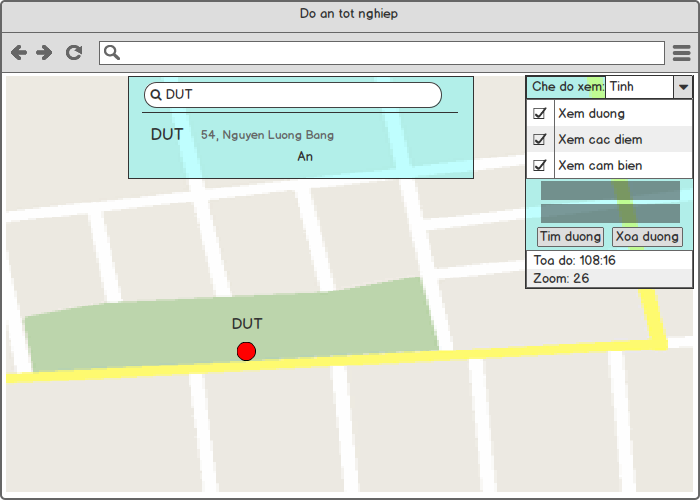


Hình 2.3: Màn hình chức năng xem bản đồ

Bảng 2.2: Đặc tả màn hình chức năng xem bản đồ

|  |  |  |  |
| --- | --- | --- | --- |
| **Screen** | Xem bản đồ | | |
| **Description** | Cho phép admin/client xem bản đồ Đà Nẵng | | |
| **Screen Access** | Admin/client truy cập đường dẫn đến ứng dụng | | |
| **Screen Content** | | | |
| **Item** | **Type** | **Data** | **Description** |
| Search | Textbox – String |  | Trường dành cho client nhập từ cần tìm |
| Chế độ xem | Dropdown | 0 – Xem bản đồ toàn quốc  1 – Xem bản đồ theo tỉnh | Trường dành cho client chọn chế độ xem bản đồ |
| Xem đường, xem các điểm, xem cảm biến | Checkbox |  | Tùy chọn các đối tượng cần xem |
| Tìm đường | Button |  | Bắt đầu tìm đường |
| Xóa đường | Button |  | Xóa tọa độ cần tìm đường |
| Tọa độ | Label |  | Tọa độ bản đồ hiện tại |
| Zoom | Label |  | Mức phòng đại hiện tại |
| **Screen Actions** | | | |
| **Action Name** | **Description** | **Success** | **Failure** |
| Di chuyển bản đồ | Khi client kéo chuột trên bản đồ | Bản đồ di chuyển theo chuột |  |
| Thu/phóng bản đồ | Lăn chuột trên bản đồ | Bản đồ sẽ phòng to hoặc thu nhỏ theo chuột |  |
| Thay đổi chế độ xem | Thay đổi đối tượng chế độ xem | Tải các bản đồ đúng theo chế độ xem |  |
| Thay đổi đối tượng cần xem | Chọn/bỏ chọn các đối tượng xem đường/xem các điểm/xem cảm biến | Tải các đối tượng cần xem |  |

2.3.1.2. Tìm kiếm địa điểm

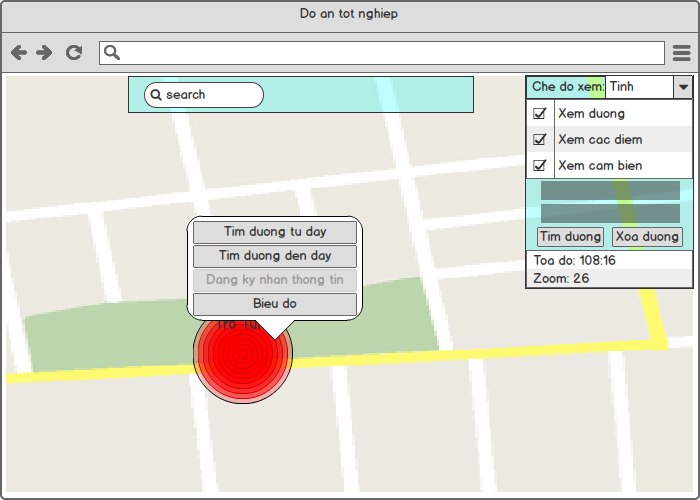


Hình 2.4:Màn hình chức năng tìm kiếm

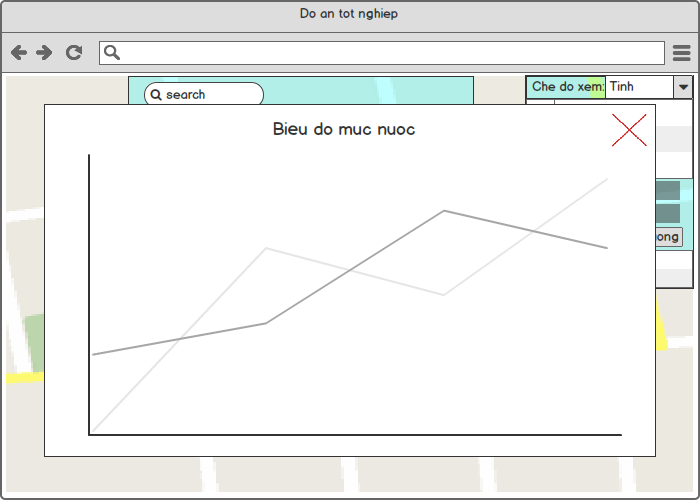
Bảng 2.3: Đặc tả màn hình chức năng tìm kiếm

|  |  |  |  |
| --- | --- | --- | --- |
| **Screen** | Tìm kiếm địa điểm | | |
| **Description** | Cho phép client tìm kiếm các địa điểm trên bản đồ | | |
| **Screen Access** | Admin/client truy cập xem bản đồ | | |
| **Screen Content** | | | |
| **Item** | **Type** | **Data** | **Description** |
| Search | Textbox – String |  | Trường dành cho client nhập từ cần tìm |
| Danh sách kết quả | Dropdown |  | Kết quả tìm kiếm |
| Ẩn | Button |  | Chọn ẩn kết qủa tìm kiếm |
| **Screen Actions** | | | |
| **Action Name** | **Description** | **Success** | **Failure** |
| Tìm kiếm | Nhập dữ liệu vào ô tìm kiếm và gửi đi | Hiển thị danh sách các địa điểm kết quả | Hiện cảnh báo ‘Error’ |
| Ẩn | Ẩn kết quả tìm kiếm | Ẩn kết quả tìm kiếm vừa hiển thị |  |
| Chọn đối tượng tìm kiếm | Chọn vào đối tượng trong danh sách kết quả tìm kiếm | Nhắm bản đồ đến đối tượng được chọn |  |

2.3.1.3. Xem tình hình mực nước



Hình 2.5: Màn hình chức năng xem tình hình mực nước

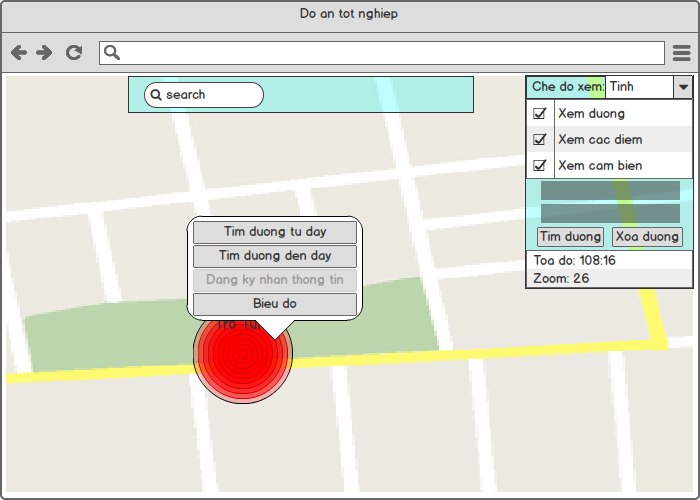


Hình 2.6: Màn hình chức năng xem tình hình mực nước

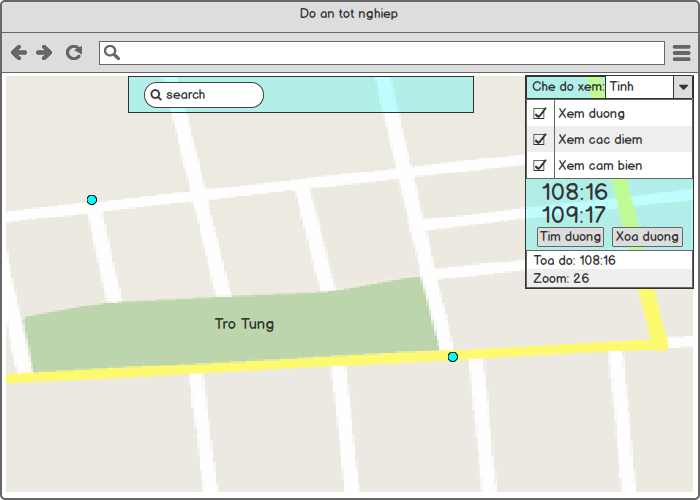
Bảng 2.4: Đặc tả màn hình chức năng xem tình hình mực nước

|  |  |  |  |
| --- | --- | --- | --- |
| **Screen** | Xem tình hình mực nước | | |
| **Description** | Cho phép admin/client xem mức nước hiện tại và trong quá khứ | | |
| **Screen Access** | Admin/client truy cập xem bản đồ | | |
| **Screen Content** | | | |
| **Item** | **Type** | **Data** | **Description** |
| Tìm đường từ đây | Button |  | Chọn điểm bắt đầu |
| Tìm đường đến đây | Button |  | Chọn điểm kết thúc |
| Đăng ký nhận thông tin | Button |  | Đăng ký nhận thông báo khi mược nước vượt ngưỡng an toàn |
| Biểu đồ | Button |  | Chọn xem biểu đồ mực nước |
| Biểu đồ mực nước | Iframe |  | Biểu đồ mực nước trong 7 ngày gần nhất |
| Close | Button |  | Tắt biểu đồ |
| **Screen Actions** | | | |
| **Action Name** | **Description** | **Success** | **Failure** |
| Click phải tại điểm có cảm biến trên bản đồ | Tùy chọn điểm cảm biến | Hiển thị bảng tùy chọn |  |
| Chọn xem biểu đồ mực nước | Chọn xem biểu đồ mực nước | Hiển thị biểu đồ mực nước | Hiển thị lỗi |
| Đóng biểu đồ | Đóng biểu đồ | Đóng biểu đồ |  |

2.3.1.4. Tìm đường



Hình 2.7: Màn hình chức năng tìm đường

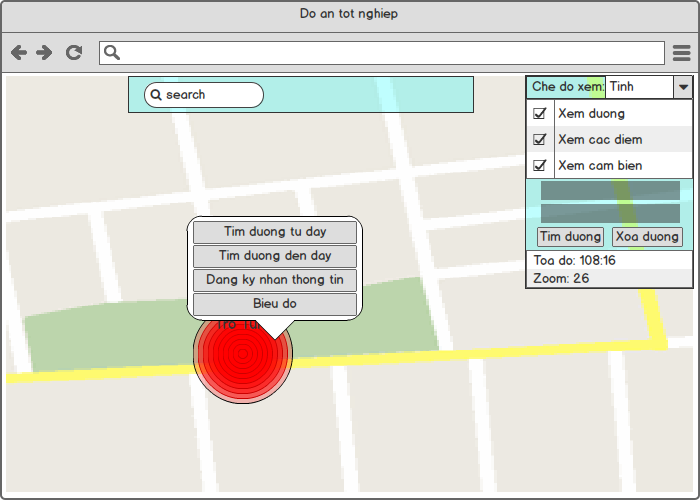


Hình 2.8: Màn hình chức năng tìm đường

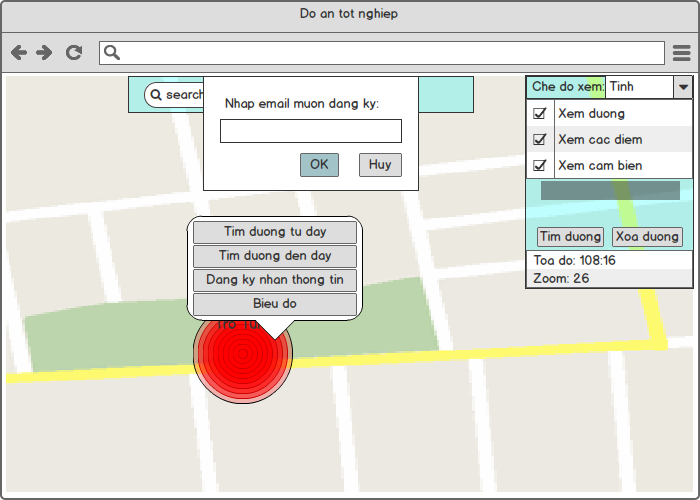
Bảng 2.5: Đặc tả màn hình chức năng tìm đường

|  |  |  |  |
| --- | --- | --- | --- |
| **Screen** | Tìm đường đi ngắn nhất | | |
| **Description** | Cho phép admin/client tìm đường đi ngắn nhất giữa 2 điểm | | |
| **Screen Access** | Admin/client truy cập xem bản đồ | | |
| **Screen Content** | | | |
| **Item** | **Type** | **Data** | **Description** |
| Tìm đường từ đây | Button |  | Chọn điểm bắt đầu |
| Tìm đường đến đây | Button |  | Chọn điểm kết thúc |
| Tìm đường | Button |  | Bắt đầu tìm đường |
| Xóa đường | Button |  | Xóa tọa độ cần tìm đường |
| **Screen Actions** | | | |
| **Action Name** | **Description** | **Success** | **Failure** |
| Chọn điểm bắt đầu | Click phải điểm cần chọn, chọn ‘tìm đường từ đây’ | Hiển thị bảng tùy chọn |  |
| Chọn điểm bắt đầu | Click phải điểm cần chọn, chọn ‘tìm đường từ đây’ | Hiển thị bảng tùy chọn |  |
| Chọn tìm đường | Click chọn tìm đường | Hiển thị đường ngắn nhất trên bản đồ |  |
| Chọn xóa đường | Click chọn xóa đường | Xóa đường vừa tìm được |  |

2.3.1.5. Đăng ký nhận thông báo



Hình 2.9: : Màn hình chức năng đăng ký nhận thông báo

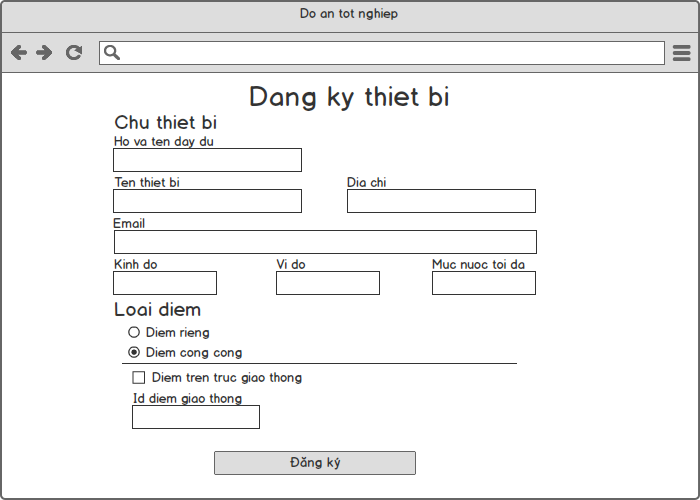


Hình 2.10: : Màn hình chức năng đăng ký nhận thông báo

Bảng 2.6: Đặc tả màn hình chức năng đăng ký nhận thông báo

|  |  |  |  |
| --- | --- | --- | --- |
| **Screen** | Đăng ký nhận thông báo | | |
| **Description** | Cho phép client đăng ký nhận thông báo khi mực nước quá mức cho phép | | |
| **Screen Access** | Client truy cập xem bản đồ | | |
| **Screen Content** | | | |
| **Item** | **Type** | **Data** | **Description** |
| Đăng ký nhận thông tin | Button |  | Mở form đăng ký nhận thông tin |
| Email | Input |  | Ô nhập email |
| OK | Button |  | Đăng ký |
| Hủy | Button |  | Hủy đăng ký |
| **Screen Actions** | | | |
| **Action Name** | **Description** | **Success** | **Failure** |
| Chọn đăng ký nhận thông tin | Click phải điểm cần chọn, chọn ‘tìm đường từ đây’ | Hiển thị bảng tùy chọn để nhập email |  |
| Đăng ký | Nhập email và nhấn OK | Tắt bảng nhập email | Nếu email không hợp lệ thì hiện thông báo ‘Email không hợp lệ’  Nếu đăng ký lỗi, hiện thông báo lỗi tương ứng. |
| Hủy | Chọn Hủy | Tắt bảng nhập email |  |

2.3.2. Admin



Hình 2.11: Màn hình chức năng đăng ký thiết bị

Bảng 2.7: Đặc tả màn hình chức năng đăng ký thiết bị

|  |  |  |  |
| --- | --- | --- | --- |
| **Screen** | Cài đặt địa điểm mới | | |
| **Description** | Cho phép admin đăng ký thêm thiết bị | | |
| **Screen Access** |  | | |
| **Screen Content** | | | |
| **Item** | **Type** | **Data** | **Description** |
| Họ và tên | Input |  | Ô nhập họ và tên chủ thiết bị |
| Tên thiết bị | Input |  | Ô nhập tên thiết bị |
| Địa chỉ | Input |  | Ô nhập địa chỉ thiết bị |
| Email | Input (email) |  | Ô nhập email |
| Kinh độ | Number | >-180, <=180 | Ô nhập kinh độ |
| Vĩ độ | Number | >-90, <=90 | Ô nhập vĩ độ |
| Mực nước tối đa | Number |  | Ô nhập mực nước tối đa |
| Loại điểm | Radio button | Điểm riêng - private  Điểm công cộng - public | Chọn loại điểm, nếu là điểm công cộng sẽ hiển thị checkbox điểm trên trục giao thông |
| Điểm trên trục giao thông | Checkbox |  | Nếu là điểm trên trục giao thông thì chọn, nếu chọn thì hiển thị input id điểm giao thông |
| Id điểm giao thông | Number |  | Ô nhập id điểm giao thông, id xem cơ sở dữ liệu điểm giao thông |
| Đăng ký | Button |  | Đăng ký |
| **Screen Actions** | | | |
| **Action Name** | **Description** | **Success** | **Failure** |
| Đăng ký | Nhập đủ thông tin và nhấn OK | Hiển thị thành công | Nếu các trường nhập không đầy đủ, hiển thị thông báo chưa nhập bên cạnh ô nhập.  Có lỗi trong quá trình xử lý thì hiển thị theo lỗi của máy chủ trả về. |

2.4. Biều đồ tuần tự mốt số chức năng chính

2.4.1. Cập nhật dữ liệu cảm biến



Hình 2.12: Biều đồ tuần tự chức năng cập nhật dữ liệu cảm biến

2.4.2. Đăng ký thêm thiết bị mới



Hình 2.13: Biều đồ tuần tự chức năng thêm mới thiết bị

2.4.3. Yêu cầu biểu đồ mực nước



Hình 2.14: Biều đồ tuần tự chức năng yêu cầu biểu đồ mực nước

2.4.4. Đăng ký nhận email



Hình 2.15: Biều đồ tuần tự chức năng đăng ký nhận email thông báo

# TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

3.1. Triển khai hệ thống

3.1.1. Môi trường triển khai

Hệ thống có thể triển khai trên bất kỳ, trong đó có thể là Windows, MacOS, Linux,… Hoặc có thể triển khai trên các loại công nghệ đám mây phổ thông như Google Cloud platform, Amazon Web Services (AWS), Microsoft Azure.

Trong phạm vi đồ án, các dịch vụ sẽ được triển khai trên hệ điều hành Windows 10. Trình duyệt web được sử dụng là Google Chrome.

Các môi trường cần được cài đặt:

* PostgreSQL và PostGIS
* GeoServer
* ElasticSearch
* NodeJS
* Java và Java Spring Boot
* Arduino IDE

Ngoài các môi trường thực thi chương trình, cần thêm môi trường lập trình, có thể dùng bất kỳ công cụ nào, có thể là Sublime Text, Visial Code hoặc Notepad.

Ngoài những công cụ lập trình và môi trường thực thi, có thể cài đặt thêm một vài công cụ sau:

* PostGIS 2.0 Shapefile and DBF Loader Exporter: Công cụ dùng để nhâp dữ liệu bản đồ từ bên ngoài vào PostgreSQL.
* QGIS3: Công cụ hỗ trợ chỉnh sửa bản đồ trục tiếp trên cơ sở dữ liệu PostgreSQL.
* Logstash: Công cụ hỗ trợ nhập dữ liệu tìm kiếm từ PostgreSQL đến ElasticSearch.

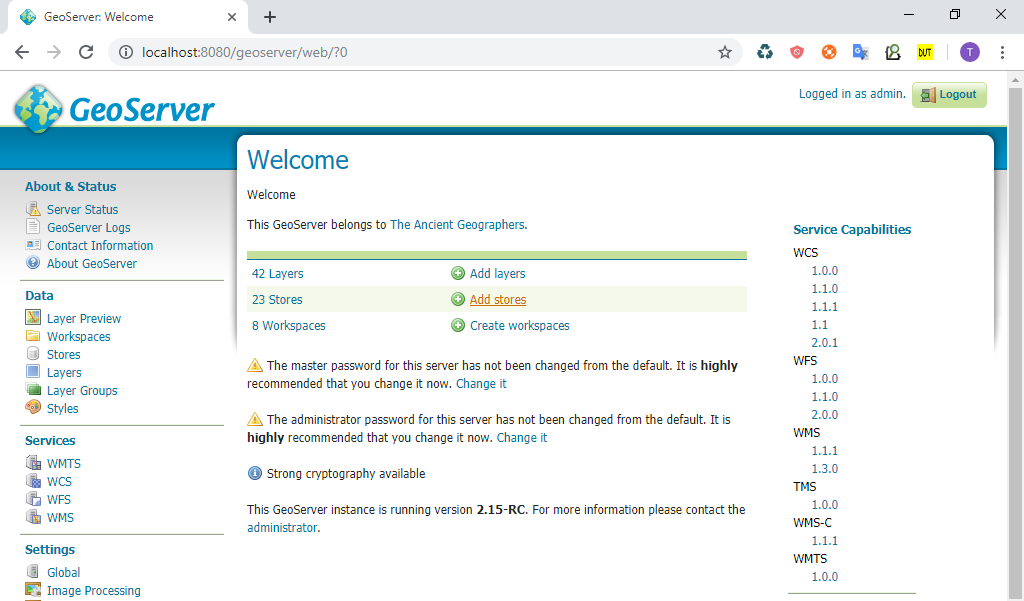
3.1.2. Cài đặt môi trường

3.1.2.1. Cài đặt GeoServer

Để cài đặt công cụ này, cần phải cài đặt Java trước trên máy, và cấu hình path JAVA-HOME ở môi trường máy tính. Java được nhiều công cụ sử dụng, nhưng rất phổ biến nên không đề cập cài đặt Java trong đồ án này.

Truy cập đường dẫn đến phần ‘download’ trên trang chủ của GeoServer và chọn các tùy chọn để tải về bộ cài đặt GeoServer. Cơ bản có 2 gói cài:

* Bộ cài đặt trực tiếp lên hệ điều hành, sau khi cài đặt khởi chạy bằng cháy kích hoạt tệp startup.bat trong thư mục /bin của thư mục cài đặt.
* Hoặc gói ứng dụng khởi chạy máy chủ qua Tomcat - một ứng dụng chủ (Application Server), khởi chạy qua cửa sổ quản lý ứng dụng của Tomcat.



Hình 3.1: Màn hình GeoServer trên trình duyệt khi sau khởi động

3.1.2.2. Cài đặt PostgreSQL

Truy câp trang <https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>, chọn phiên bản phù hợp để tải về.

Sau khi tải về, tiến hành cài đặt từ tệp thực thi. Sau khi hoàn tất, cài đặt được PostgreSQL và PgAdmin4 (theo phiên ban hiện tại). PgAdmin4 là công cụ để xem dữ liệu trực quan từ PostgreSQL.

Sau khi tạo cơ sở dữ liệu trên PostGreSQL thì đó chỉ là cơ sở dữ liệu bình thường, không phải là cơ sở dữ liệu không gian. Thực thi lệnh sau để cài đặt các thư viện PostGIS trên cơ sở dữ liệu vừa tạo. Bây giờ cơ sở dữ liệu có các phần mở rộng PostGIS.

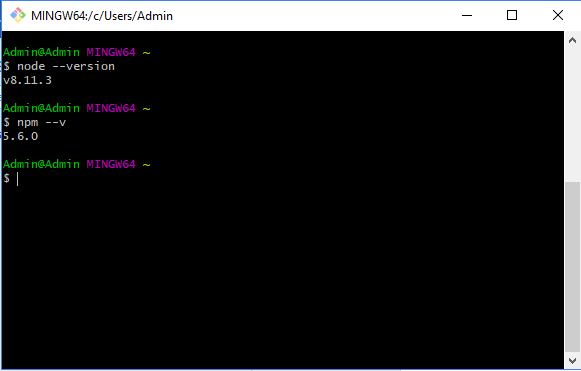
CREATE EXTENSION postgis;

3.1.2.3. Cài đặt NodeJS

Truy câp trang <https://nodejs.org/en/download/>, chọn phiên bản phù hợp để tải về.

Sau khi tải về, chỉ cần thực thi tệp cài đặt, cấu hình phù hợp sẽ cài đặt hoàn tất dễ dàng.

Nếu máy tính thuộc mạng có cấu hình phức tạp, cần cấu hình proxy cho NodeJS. Vì NodeJS sử dụng công cụ quản lý các gói thư viện, thông quan mạng để có thể cài đặt các gói cần thiết. Dùng 2 lệnh sau để kiểm tra cài đặt: ‘node --version’ và ‘npm –v’



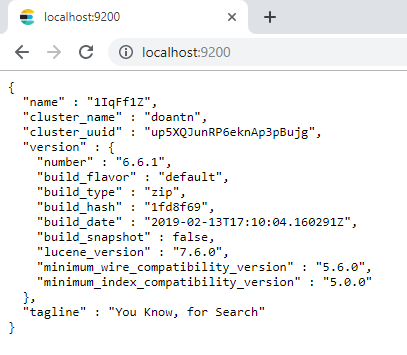
Hình 3.2: Kiểm tra cài đặt NodeJS

3.1.2.4. Cài đặt ElasticSearch

Truy cập đường dẫn <https://www.elastic.co/products/elasticsearch> để tải bộ công cụ ES, ngoài ra có thể cài thêm Kibana – là công cụ giúp xem cũng như truy vấn dữ liệu từ ES và Logstash - là công cụ hỗ trợ nhập dữ liệu từ các nguồn dữ liệu ngoài ES vào ES.

Để khởi chạy ES, cần chạy file elasticsearch.bat ở thư mục /bin của ES sau khi tải và giải nén. Nếu có thay đổi cấu hình của ES, truy cập và chỉnh sửa file /config/ elasticsearch.yml của ES và khởi động lại ES để cấu hình mới được cập nhật.

Gọi API đến địa chỉ localhost:9200 để kiểm tra ES. Như hình bên dưới là đã khởi chạy thành công.



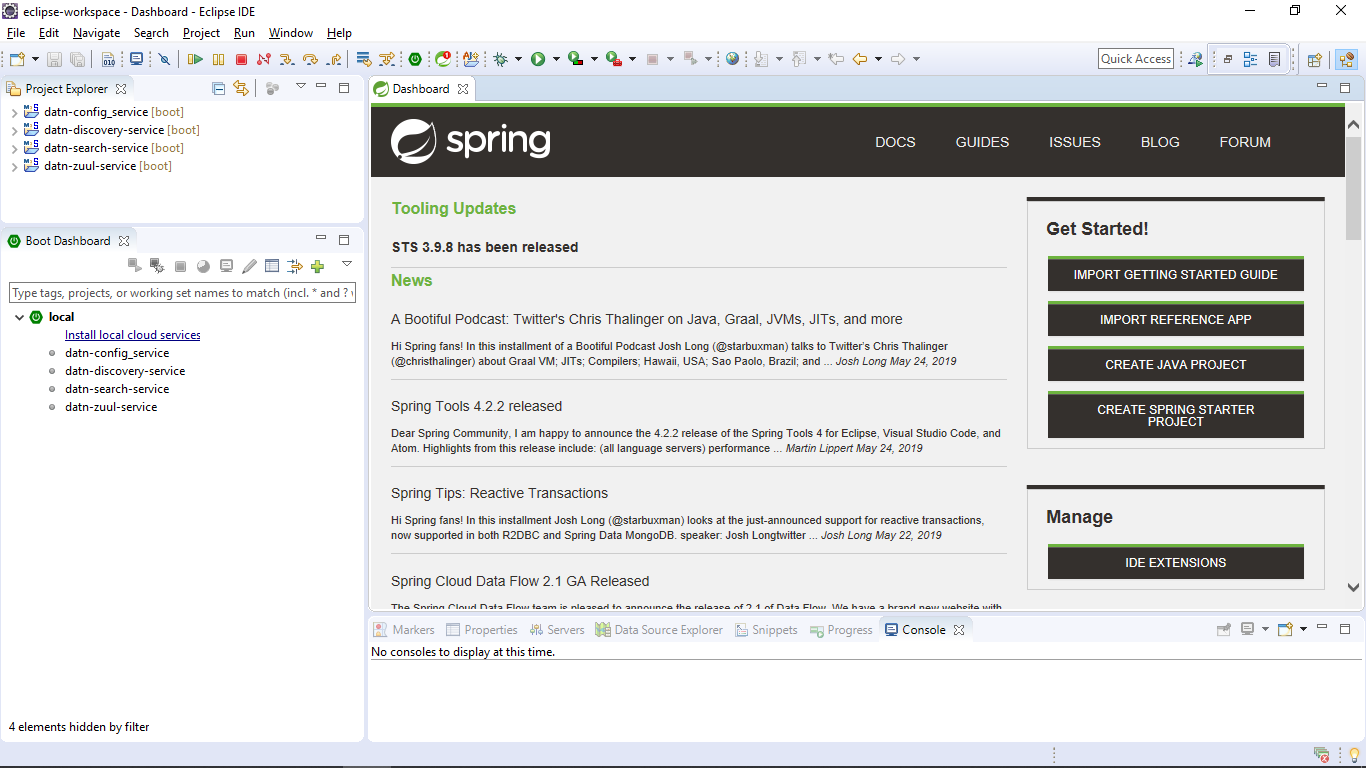
Hình 3.3: Kết quả kiểm tra cài đặt ElasticSearch thành công

3.1.2.5. Cài đặt Spring Boot

Để tiện cho lập trình lẫn gỡ lỗi, cần cài đặt Eclipse. Truy câp trang https://www.eclipse.org/downloads/, chọn phiên bản phù hợp để tải về.

Sau khi tải về, thưc thi tệp và chọn cấu hình phù hợp để lập trình mạng.

Khi hoàn thành cài đặt, vào phần Eclipse Marketplace trong mục Help. Tìm và cài đặt công cụ Spring Tool Suite để cài đặt Spring Boot. Sau khi cài đặt xong, khởi động lại Eclipse để kiểm tra cài đặt. Khi trên Eclipse có phần Boot Dashboard, khi đó cài đặt Spring Boot trên Eclipse đã hoàn thành.



Hình 3.4: Màn hình Eclipse sau khi cài đặt Spring Boot Suite

3.2. Kết quả thực nghiệm

3.2.1. Giao diện người dùng

3.2.1.1.

**Ghi chú về trình bày mỗi trang của đồ án**

* Page layout: cỡ giấy A4; lề trái: 3cm, lề phải: 2cm, lề trên: 2,5cm, lề dưới: 2,5cm; header và footer: from edge: 1,6cm;
* Đánh số trang: bắt đầu đánh số trang từ phần “MỞ ĐẦU”;
* Tiêu đề chương, mục, tiểu mục: xem chi tiết như minh họa ở trên;
* Mục Header: Tên đề tài (định dạng: font Time New Roman, Italic, size 10, căn lề: giữa);
* Mục Footer: Sinh viên thực hiện, giảng viên hướng dẫn, đánh số trang (định dạng: font Time New Roman, size 10);
* Chú dẫn bảng: nằm trên bảng, đánh số theo chương và số lũy tiến theo số thứ tự của bảng trong chương;
* Chú dẫn hình: nằm dưới hình, đánh số theo chương và số lũy tiến theo số thứ tự của hình trong chương;
* Đánh số công thức: bên phải công thức, đánh số theo chương và số lũy tiến theo số thứ tự của công thức trong chương;
* Nên sử dụng các chức năng về Bookmark, Caption, Cross-Reference, Format Heading,… của Microsoft Word hoặc các phần mềm soạn thảo tương tự; cần tổ chức theo dạng “Long Document”.

**KẾT LUẬN {size 14**}

{Để 2 dòng trống}

Nội dung kết luận {Font: Time New Roman; thường; cỡ chữ: 13; dãn dòng: 1,3; căn lề: justified}

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

**Ghi chú về phần Kết luận**

* Phần Kết luận cần phải nêu được những kết luận chung, khẳng định những kết quả đạt được, những đóng góp, đề xuất và kiến nghị (nếu có);
* Trong phần này, có thể định dạng các điểm/ mục kết luận theo dạng Outline hoặc Numbering hoặc Bullets.

**TÀI LIỆU THAM KHẢO**

{bold, size 14}

{Để 2 dòng trống}

{Font: Time New Roman; thường; cỡ chữ: 13; dãn dòng: 1,3; căn lề: justified}

https://vi.wikipedia.org/wiki/GeoServer

**Ghi chú:**

Sinh viên xem “Quy định về liêm chính học thuật” ban hành kèm theo Quyết định số 29/QĐ-ĐHBK ngày 09/01/2017 và “Hướng dẫn trích dẫn và lập Danh mục tài liệu tham khảo” được ban hành theo văn bản số 30/HD-ĐHBK ngày 09/01/2017 để thực hiện trích dẫn và lập Danh mục tài liệu tham khảo.

**PHỤ LỤC 1**

{bold, size 14}

{Font: Time New Roman; thường; cỡ chữ: 12; dãn dòng: 1,3; căn lề: justified}

**PHỤ LỤC 2**

{bold, size 14}

{Font: Time New Roman; thường; cỡ chữ: 12; dãn dòng: 1,3; căn lề: justified}