

MỤC LỤC

CHƯƠNG 1. TỔNG QUAN VỀ WEBGIS VÀ TÌNH HÌNH PHÁT TRIỂN WEBGIS TRÊN THẾ GIỚI VÀ Ở VIỆT NAM	1
1.1 Tổng quan về webgis.....	1
1.1.1 Công nghệ thành lập WebGIS.....	1
1.1.2 Kiến trúc WebGIS và các bước xử lý.	1
1.1.3 Các chuẩn trao đổi WebGIS hiện nay:.....	8
1.2 Tình hình phát triển công nghệ WEB GIS trên thế giới	10
1.3 Tình hình phát triển công nghệ WebGISở Việt Nam.....	13
CHƯƠNG 2 THIẾT KẾT CẤU TRÚC VÀ XÂY DỰNG NỘI DUNG CỦA HỆ THỐNG WEBGIS.....	14
2.1 Phân tích nền tảng công nghệ	14
2.1.1 Ngôn ngữ lập trình Java	14
2.1.2 Thư viện mã nguồn mở OpenCV.....	14
2.1.3 Cơ sở dữ liệu đồ họa PostGis.....	15
2.1.4 Máy chủ Geoserver	18
2.1.5 JavaScript, JQuery và thư viện mã nguồn mở OpenLayers.....	20
2.2 Phương pháp thiết kế hệ thống.....	21
2.2.1 Trước khi UML ra đời:.....	23
2.2.2 Sự ra đời của UML:.....	24
2.2.3 UML (Unified Modeling Language):	24
2.2.4 Phương pháp và các ngôn ngữ mô hình hoá:	25
CHƯƠNG 3. ĐÁNH GIÁ, LỰA CHỌN CÔNG NGHỆ VÀ PHÁT TRIỂN HỆ THỐNG	26
3.1 Khái quát chung	26
3.2 Phương pháp nghiên cứu và cách tiếp cận	26
3.2.1 Công nghệ Nền là gì?	26
3.2.2 Căn cứ kiến trúc hệ thống	27

3.2.3 Căn cứ các giải pháp thu thập, cập nhật dữ liệu dữ liệu	27
3.2.4 Căn cứ các giải pháp lưu trữ dữ liệu không gian và dữ liệu thuộc tính	27
3.2.5 Căn cứ các giải pháp kỹ thuật phân phối dữ liệu	28
3.2.6 Căn cứ các yêu cầu về tính an ninh, bảo mật của hệ thống, cơ sở hạ tầng phục vụ hệ thống và quy trình công nghệ phát triển hệ thống.....	28
3.3 Kết quả đánh giá, lựa chọn và phát triển hệ thống.....	29
3.3.1 Công nghệ nền cho quản trị cơ sở dữ liệu.....	29
3.3.2 Công nghệ bản đồ.....	33
3.3.3 Công cụ phát triển phần mềm	34
3.3.4 Mô hình chức năng của hệ thống	43
3.3.5 Các chức năng của hệ thống.....	44
CHƯƠNG 4 TÍCH HỢP, KIỂM THỬ, ĐÁNH GIÁ VÀ HIỆU CHỈNH	47
HỆ THỐNG	47
4.1 Kiểm thử hệ thống	47
4.1.1 Khái niệm kiểm thử hệ thống.....	47
4.1.2. Các nguyên tắc kiểm thử	48
4.1.3. Độ tin cậy của phần mềm.....	48
4.1.4. Quá trình kiểm thử	49
4.1.5. Chiến lược kiểm thử phần mềm	54
4.2 Tích hợp và kiểm thử hệ thống webgi sông Cả	57
4.2.1. Tích hợp các hợp phần	57
4.2.2. Kiểm thử hệ thống.....	60
4.2.3. Kiểm tra khả năng chịu tải	61
4.3 Đánh giá và hiệu chỉnh hệ thống.....	82
4.3.1 Các tiêu chí cơ bản	82
4.3.2 Hiệu suất xử lý	84
4.3.3 Tính dễ hiểu.....	84
KẾT LUẬN VÀ KIẾN NGHỊ	88

TÀI LIỆU THAM KHẢO	90
PHỤ LỤC	92
<i>Hướng dẫn sử dụng công cụ trên WebGIS quản lý và sử dụng nguồn nước sông</i>	
<i>Cả</i>	<i>92</i>

DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

CNTT Công nghệ thông tin

TV Tivi

ISO International Standard Organization (tổ chức tiêu chuẩn quốc tế)

IEC International Electrotechnical Commission (Ủy ban kỹ thuật điện quốc tế)

DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

Hình 1.1 Cấu trúc WEBGIS	2
Hình 1.2 Các dạng yêu cầu từ phía client	4
Hình 1.3 Cấu hình Server-side	6
Hình 1.4 Cấu hình Client Side	7
Hình 1.5 Khả năng xử lý GIS.....	8
Hình 1.6 Các chức năng của WEBGIS	9
Hình 1.7 Các thông số trong chuỗi URL thực hiện chức năng GetMap.....	10
Hình 2.1 Vị trí của PostGIS trong PostgreSQL	16
Hình 3.1 Vị trí của PostGIS trong PostgreSQL	30
Hình 3.2 Mô hình chức năng của hệ thống web	43
Hình 3.3 Mô hình kiến trúc công nghệ của hệ thống.....	44
Hình 3.4 Mô hình chức năng xuất nhập dữ liệu.....	45
Hình 3.5 Chức năng bản đồ đối với quản trị viên tầng ứng dụng.....	45
Hình 3.6 Chức năng bản đồ đối với người dùng.....	46
Hình 4.1: Quy trình kiểm thử	49
Hình 4.2: hệ quản trị CSDL PostgresSQL	58
Hình 4.3: Cấu trúc file GeoJSON	59
Hình 4.4: Các chức năng cơ bản của webgis sông Cả	60
Hình 4.5: kết quả kiểm tra hiệu năng	61
Hình 4.6: Chất lượng trong vòng đời sản phẩm.....	66
Hình 4.7: Mô hình chất lượng cho chất lượng trong và ngoài.....	69
Hình 4.8: Mô hình chất lượng sử dụng	72
Hình 4.9: Mối liên hệ giữa tiêu chuẩn ISO 9126 và ISO 14598.....	74
Hình 4.10: Khung tham số chất lượng phần mềm	76
Hình 4.11: Quy trình đánh giá sản phẩm phần mềm	81
Hình 4.12: các nhóm Bản đồ của hệ thống webGIS sông Cả	83
Hình 4.13: Các chức năng cơ bản của webGIS	83

Hình 4.14: Kiến trúc áp dụng trong webGIS sông Cả.....	84
Hình 4.16: giao diện hệ quản trị CSDL PostgreSQL.....	86
Hình 4.17: Cấu trúc file GeoJSON sử dụng trong hệ thống webGIS sông Cả...	87

CHƯƠNG 1. TỔNG QUAN VỀ WEBGIS VÀ TÌNH HÌNH PHÁT TRIỂN WEBGIS TRÊN THẾ GIỚI VÀ Ở VIỆT NAM

1.1 Tổng quan về webgis

Ứng dụng web-GIS quản lý và sử dụng nguồn nước lưu vực sông Cả là một ứng dụng cho phép nhập dữ liệu sản phẩm của đề tài như bản đồ, số liệu, bảng biểu ..., hiển thị dữ liệu bằng hình ảnh lên trang web bản đồ.

Trên giao diện bản đồ web, người dùng sẽ nhìn thấy các kết quả thực hiện của đề tài (bản đồ hệ thống thủy lợi, các bản đồ chuyên đề, hệ thống bảng biểu thống kê...), đồng thời có các chức năng thống kê, thao tác phân tích xử lý dữ liệu thông qua giao diện web.

1.1.1 Công nghệ thành lập WebGIS

✓ WebGIS là hệ thống thông tin địa lý phân tán trên một mạng các máy tính để tích hợp, trao đổi các thông tin địa lý trên World Wide Web (Edward, 2000, URL). Trong cách thực hiện nhiệm vụ phân tích GIS, dịch vụ này gần giống như là kiến trúc Client-Server của Web. Xử lý thông tin địa lý được chia ra thành các nhiệm vụ ở phía server và phía client.

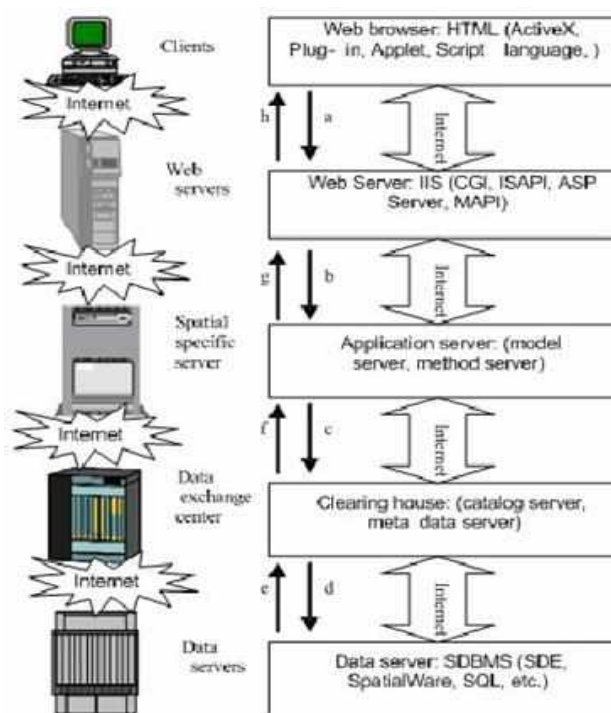
Một client tiêu biểu là trình duyệt web và server-side bao gồm một Web server có cung cấp một chương trình phần mềm WebGIS. Client thường yêu cầu một ảnh bản đồ hay vài xử lý thông tin địa lý qua Web đến server ở xa. Server chuyển đổi yêu cầu thành mã nội bộ và gọi những chức năng về GIS bằng cách chuyển tiếp yêu cầu tới phần mềm WebGIS. Phần mềm này trả về kết quả, sau đó kết quả này được định dạng lại cho việc trình bày bởi trình duyệt hay những hàm từ các plug-in hoặc Java applet. Server sau đó trả về kết quả cho client để hiển thị, hoặc gửi dữ liệu và các công cụ phân tích đến client để dùng ở phía client.

1.1.2 Kiến trúc WebGIS và các bước xử lý.

✓ Kiến trúc webGis:

Kiến trúc xuất bản web của hệ thống tin dữ liệu không gian cũng gần giống như kiến trúc dành cho một hệ thống tin web cơ bản khác, ngoại trừ có ứng dụng

GIS sử dụng các kỹ thuật khác. Có nhiều dạng của việc xuất bản web cho thông tin không gian.

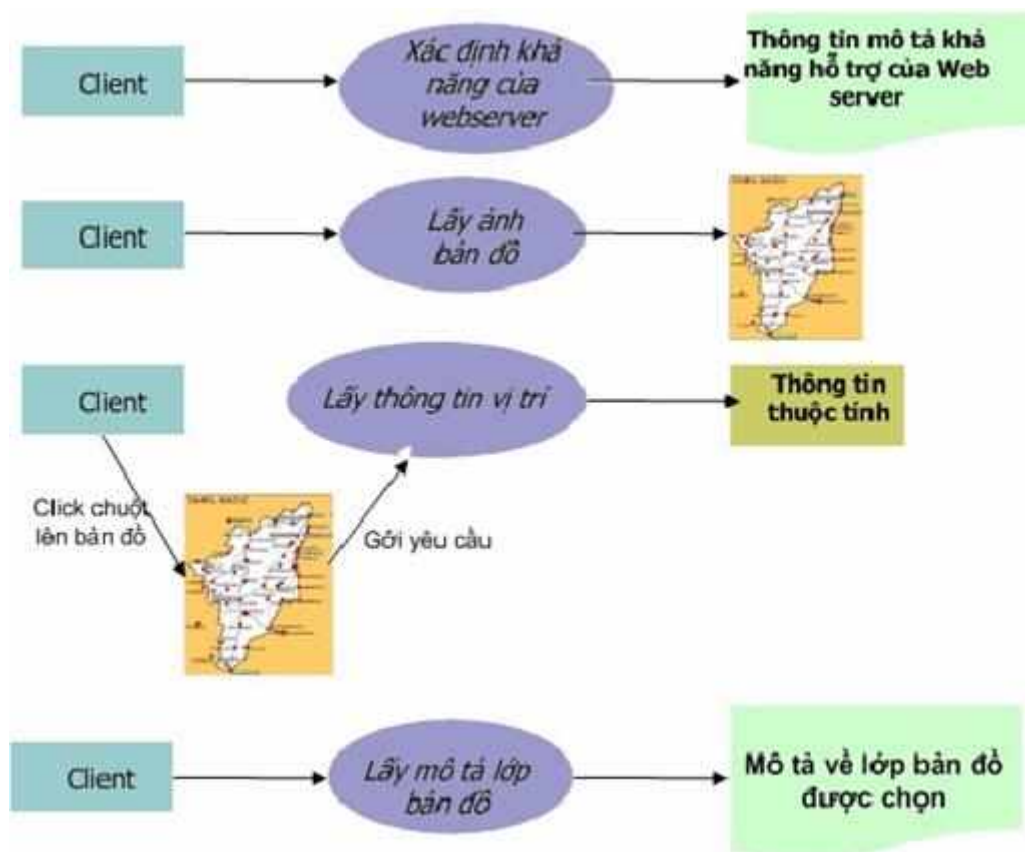


Hình 1.1 Cấu trúc WEBGIS

Cơ sở dữ liệu không gian sẽ được dùng để quản lý và truy xuất dữ liệu không gian, được đặt trên data server. Nhà kho hay nơi lưu trữ (clearing house) được dùng để lưu trữ và duy trì những siêu dữ liệu (dữ liệu về dữ liệu - metadata) về dữ liệu không gian tại những data server khác nhau. Dựa trên những thành phần quản lý dữ liệu, ứng dụng server và mô hình server được dùng cho ứng dụng hệ thống để tính toán thông tin không gian thông qua các hàm cụ thể. Tất cả kết quả tính toán của ứng dụng server sẽ được gửi đến web server để thêm vào các gói HTML, gửi cho phía client và hiển thị nơi trình duyệt web. Xem hình minh họa dưới đây. Lưu ý là tất cả các thành phần đều được kết nối nhau thông qua mạng Internet.

- ✓ Các bước xử lý: Quá trình làm việc với hệ thống web xử lý thông tin không gian được minh họa như hình vẽ trên.
- a. Client gửi yêu cầu của người sử dụng thông qua giao thức HTTP đến webserver.

- b. Web server nhận yêu cầu của người dùng gửi đến từ phía client, xử lý và chuyển tiếp yêu cầu đến ứng dụng trên server có liên quan.
- c. Application server (chính là các ứng dụng GIS) nhận các yêu cầu cụ thể đối với ứng dụng và gọi các hàm có liên quan để tính toán xử lý. Nếu có yêu cầu dữ liệu nó sẽ gửi yêu cầu dữ liệu đến data exchange server (server trao đổi dữ liệu).
- d. Data exchange server nhận yêu cầu dữ liệu và tìm kiếm vị trí của những dữ liệu này sau đó gửi yêu cầu dữ liệu đến server chứa dữ liệu (data server) tương ứng cần tìm.
- e. Data server dữ liệu tiến hành truy vấn lấy ra dữ liệu cần thiết và trả dữ liệu này về cho data exchange server.
- f. Data exchange server nhận dữ liệu từ nhiều nguồn data server khác nhau nằm rải rác trên mạng. Sắp xếp dữ liệu lại theo logic của yêu cầu dữ liệu, sau đó gửi trả dữ liệu về cho application server.
- g. Application server nhận dữ liệu trả về từ các data exchange server và đưa chúng đến các hàm cần sử dụng, xử lý chúng tại đây và kết quả được trả về cho web server.
- h. Web server nhận về kết quả xử lý, thêm vào các ngữ cảnh web (HTML, PHP...) để có thể hiển thị được trên trình duyệt và cuối cùng gửi trả kết quả về cho trình duyệt dưới dạng các trang web.



Hình 1.2 Các dạng yêu cầu từ phía client

✓ Trong mô hình hoạt động của WebGIS được chia ra 2 phần : các hoạt động ở phía client (client side) và các hoạt động xử lý ở phía server (server side).

▪ Client side:

Client side được dùng để hiển thị kết quả đến cho người dùng, nhận các điều khiển trực tiếp từ người dùng và tương tác với web server thông qua trình duyệt web.

Các trình duyệt web sử dụng chủ yếu HTML để định dạng trang web. Thêm vào đó một vài plug-in, ActiveX và các mã Applet được nhúng vào trình duyệt để tăng tính tương tác với người dùng.

▪ Server side: Gồm có Web server, Application server, Data server và Clearinghouse.. Server side có nhiệm vụ lưu trữ dữ liệu không gian, xử lý tính toán và trả về kết quả (dưới dạng hiển thị được) cho client side.

Web server: Web server được dùng để phục vụ cho các ứng dụng web, web server sử dụng nghi thức HTTP để giao tiếp với trình duyệt web ở phía client. cầu

từ phía client đối với ứng dụng web đều được web server nhận và thông dịch và sau đó gọi các chức năng của ứng dụng thông qua các giao tiếp mạng như MAPI, Winsock, named pipe...

Application server: Đây là phần chương trình gọi các hàm xử lý GIS, gửi yêu cầu lấy dữ liệu đến clearinghouse.

Data server: Data server là phần cơ bản của hầu hết các hệ thống thông tin với nhiệm vụ quản lý và điều khiển truy cập dữ liệu. Ban đầu, đa số GIS sử dụng File System để quản lý dữ liệu không gian và DBMS (Database Management System) để quản lý dữ liệu thuộc tính. Ngày nay có nhiều sản phẩm và giải pháp phần mềm thay thế để quản lý dữ liệu không gian và thuộc tính một cách chung nhất. Ví dụ: SDE của ESRI (1998), SpatialWare của MapInfo (1998)...

Clearing house: Clearinghouse được sử dụng để chứa dữ liệu về dữ liệu không gian được quản lý bởi các data server. Clearinghouse đóng vai trò như một cuốn catalog, clearinghouse tìm kiếm trong catalog này để tìm dữ liệu cần.

✓ Có 2 chiến thuật lựa chọn, tương ứng với 2 kiểu triển khai, kiểu thứ nhất tập trung công việc chủ yếu cho phía server, kiểu kia ngược lại tập trung công việc cho phía client.

Chiến thuật Server- side: Những chiến thuật này tập trung vào việc cung cấp dữ liệu GIS và phân tích “theo yêu cầu” bởi một server đủ mạnh, server này sẽ truy cập dữ liệu và phần mềm cần thiết để xử lý dữ liệu.

Trong WebGIS đôi khi thuật ngữ “map server” được dùng để chỉ ra rằng chiến thuật áp dụng là server- side. Mà trong đó khi người dùng gửi yêu cầu cần “map” để hiển thị, thì sẽ được “phục vụ” bởi server. Chiến thuật server-side dựa trên khả năng trình duyệt web của người dùng có thể gửi các yêu cầu đến các phần mềm GIS trên server thông qua Internet.

Các chương trình được dùng để nhận và xử lý yêu cầu người dùng có thể được viết bằng các ngôn ngữ như : Perl, Visual Basic, C++... Các chương trình

này cũng có thể mua từ các nhà sản xuất để tạo khả năng kết nối tốt hơn đến các hệ xử lý GIS đã tồn tại.

Để có thể giao tiếp với các ứng dụng WebGIS đặt trên server, web server có thể sử dụng các chuẩn giao tiếp phổ biến như CGI (Common Gateway Interface), Java, ISAPI (Internet Server Application Programming Interface), and NSAPI (Netscape Server Application Programming Interface)...

Nhìn chung, chiến thuật này áp dụng tốt nhất cho các ứng dụng WebGIS thương mại hay cộng đồng với số lượng lớn người dùng mà không quan tâm đến khả năng xử lý GIS trên các máy người dùng.

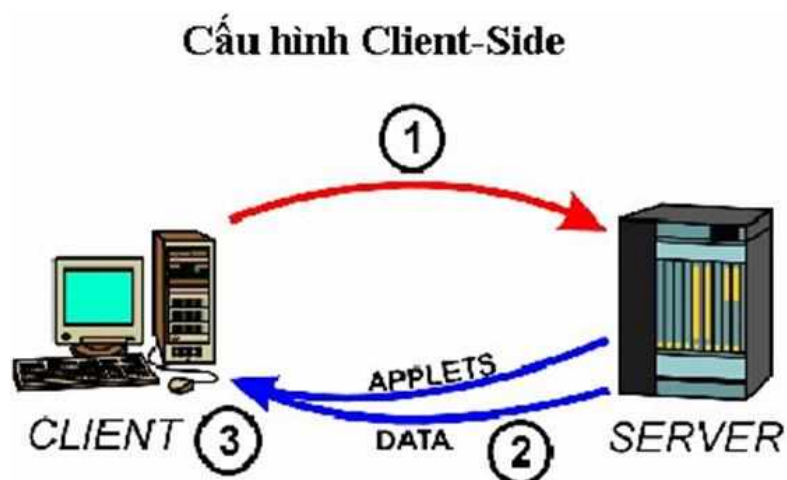


Hình 1.3 Cấu hình Server-side

▪ Chiến thuật Client-side: Thay vì để server làm quá nhiều việc, một số chức năng xử lý GIS sẽ được đưa về phía máy người dùng, và tại đây sẽ có một phần dữ liệu được xử lý. Có 2 dạng triển khai chiến thuật client side như sau:

GIS Applet được phân phối đến Client khi có yêu cầu: Trong cách triển khai chiến thuật này các xử lý GIS sẽ được server cung cấp cho phía client dưới dạng các chương trình thực thi nhỏ hoặc là các applet để có thể chạy được ở phía client. Những applet như vậy được phân phối đến client khi client cần nó để xử lý. Các applet có thể được viết bằng Java, JavaScript hoặc ở dạng các

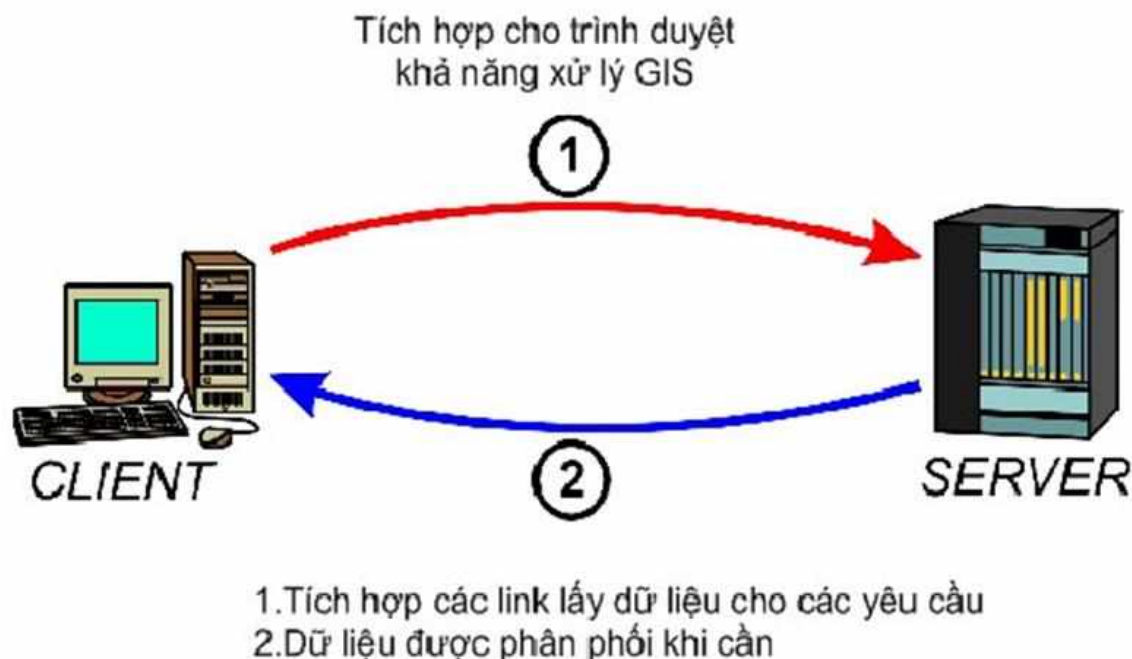
ActiveX. Như vậy trình duyệt cần được tích hợp các compiler để xử lý các applet này.



Hình 1.4 Cấu hình Client Side

GIS Applet và Plug-in cố định ở Client: Các triển khai trên cần thêm các chức năng xử lý GIS vào trình duyệt. Tuy nhiên việc chuyển dữ liệu và các applet cần thiết qua mạng Internet mất nhiều thời gian nhất là khi mà ứng dụng ít được dùng đến. Giải pháp cho vấn đề này là cách triển khai sau đây:

- + Chuyển các GIS applet đến máy tính client một cách cố định, và không phải chuyển đến mỗi khi cần nữa.
- + Download và cài đặt cố định các plug-in vào trình duyệt web của client.
- + Xây dựng một trình duyệt web có tích hợp sẵn phần mềm xử lý GIS để chạy trên client.
- + Tích hợp các link đến nguồn tài nguyên dữ liệu khác trên mạng trong mỗi gói dữ liệu tải về.
- + Server chỉ được gọi khi client khi cần dữ liệu mới, hoặc dữ liệu cho một ứng dụng mới.
- + Người dùng được toàn quyền thao tác trên dữ liệu họ dùng và phân tích chúng.



Hình 1.5 Khả năng xử lý GIS

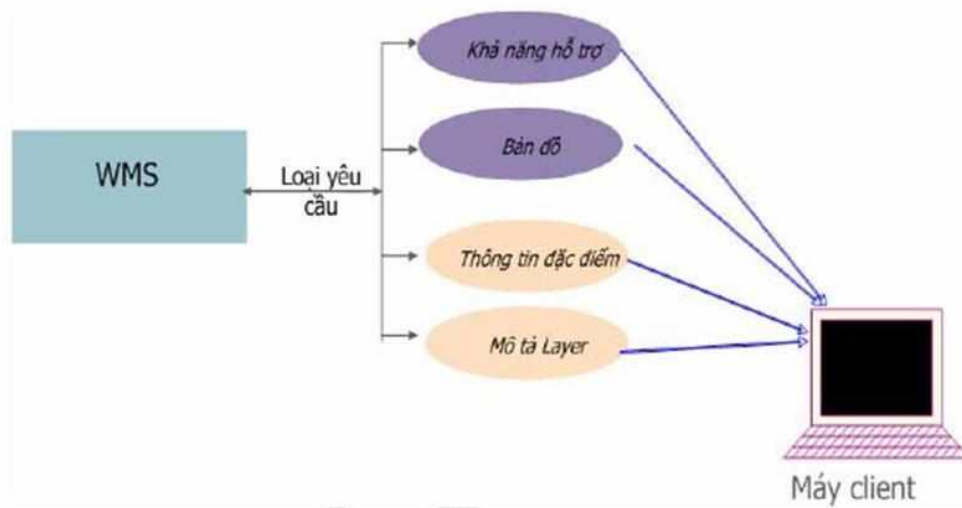
1.1.3 Các chuẩn trao đổi WebGIS hiện nay:

✓ Web Map Service / Server (WMS):

Đây là một chuẩn do tổ chức OGC đưa ra. Trong đó web server sẽ trở thành web map server có service phục vụ cho chia sẻ dữ liệu. Các hoạt động mà client có thể thực hiện thông qua service này gồm : nhận về mô tả các bản đồ, nhận về bản đồ, và các thông tin truy vấn các đặc điểm được thể hiện trên bản đồ.

Chuẩn này không áp dụng cho việc nhận về dữ liệu thô (dữ liệu chưa xử lý dạng thuộc tính hay không gian) mà thường nhận về một ảnh bản đồ dạng đồ họa. Những bản đồ này thường được tạo ra với các định dạng như PNG, GIF, JPEG hoặc cũng có thể là dưới dạng các yếu tố đồ họa như SVG (dạng XML) hoặc là định dạng WebCGM (Web Computer Graphics Metafile).

Sau đây là các chức năng của một web map service:



Hình 1.6 Các chức năng của WEBGIS

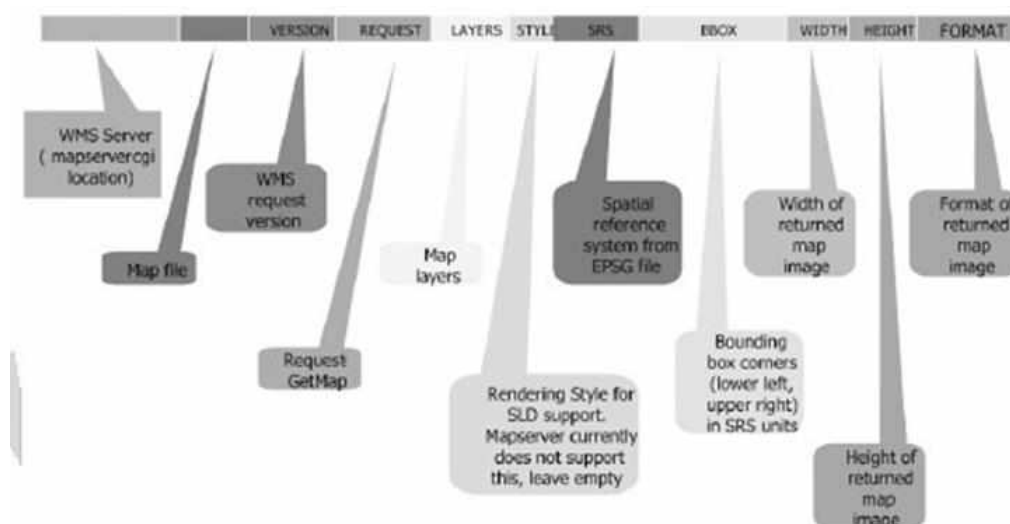
GetCapabilities - Khả năng hỗ trợ (bắt buộc): Client nhận về một mô tả thông tin về WMS, các tham số được chấp nhận và hỗ trợ, bảng mô tả, thường dưới dạng file XML.

GetMap - Lấy bản đồ (bắt buộc) : Client nhận về một ảnh bản đồ phù hợp với tham số mà client gửi lên server.

GetFeatureInfo - Lấy thông tin đặc điểm (không bắt buộc) : Client hỏi thông tin về đặc điểm nào đó (đối tượng) xuất hiện trên bản đồ.

Trình duyệt web phía client có thể thông qua Web Map Service thực hiện các chức năng này bằng cách gửi các yêu cầu dưới dạng một URL. Nội dung của chuỗi URL này phụ thuộc vào công việc được yêu cầu (chỉ ra bởi tham số Request Type).

Sau đây là các thông số trong chuỗi URL thực hiện chức năng GetMap



Hình 1.7 Các thông số trong chuỗi URL thực hiện chức năng GetMap

✓ Web Feature Service / Server (WFS):

Đây cũng là chuẩn do OGC đưa ra. Trong đó web server giờ được gọi là Web Feature server có service phục vụ việc chia sẻ dữ liệu. Tuy nhiên thay vì trả về một ảnh bản đồ dạng đồ họa thì Web Feature Server sẽ gửi trả về thông tin không gian và thông tin thuộc tính có liên quan dưới dạng file GML (Geographic Markup Language) một dạng XML và sau đó client sử dụng file XML này làm dữ liệu để tạo ra ảnh bản đồ.

Web Feature Service hoàn toàn tương tự như Web Map Service về các chức năng, cách làm việc chỉ khác nhau dạng dữ liệu trao đổi.

1.2 Tình hình phát triển công nghệ WEB GIS trên thế giới

Khái niệm Theo định nghĩa do tổ chức bản đồ thế giới (Cartophy) đưa ra thì: “Web GIS được xem như là một hệ thống thông tin địa lý được phân bố qua môi trường mạng máy tính để tích hợp, phân phối và truyền tải thông tin địa lý trực tuyến trên Internet

- Có khả năng phân phối thông tin địa lý rộng rãi trên toàn cầu.
- Người dùng Internet có thể truy cập đến các ứng dụng GIS mà không phải mua phần mềm.
- Đối với phần lớn người dùng không có kinh nghiệm về GIS thì việc sử dụng Web - GIS sẽ đơn giản hơn việc sử dụng các ứng dụng GIS loại khác.

Các phương thức phát triển Web - GIS

Có nhiều phương thức dùng để thêm các chức năng của GIS trên Web:

- Server side: cho phép người dùng gửi yêu cầu lấy dữ liệu và phân tích trên máy chủ. Máy chủ sẽ thực hiện các yêu cầu và gửi trả dữ liệu hoặc kết quả cho người dùng.

- Client side: cho phép người dùng thực hiện vài thao tác phân tích trên dữ liệu tại chính máy người dùng.

- Server và client: kết hợp hai phương thức server side và client side để phục vụ nhu cầu của người dùng. Cụ thể:

- Các tác vụ đòi hỏi sử dụng cơ sở dữ liệu (CSDL) hoặc phân tích phức tạp sẽ được gán trên máy chủ.

- Các tác vụ nhỏ sẽ được gán ở máy khách. Trong trường hợp này, cả máy chủ và máy khách cùng chia sẻ thông tin với nhau về sức mạnh và khả năng của chúng. Hiện nay công nghệ Web – GIS được xây dựng trên hai loại phần mềm là phần mềm mã nguồn mở và phần mềm thương mại (hay phần mềm mã nguồn đóng). Các phần mềm mã nguồn mở có những bước tiến mạnh mẽ trong những năm gần đây và từng bước trở thành những công cụ thiết thực và hỗ trợ tốt cho người sử dụng. Các phần mềm loại này cho GIS cũng có cùng xu hướng đó và ngày nay đã có thể đáp ứng được nhu cầu của dân chuyên nghiệp trên toàn thế giới. Với việc sử dụng công cụ Web - GIS mã nguồn mở, người sử dụng không phải mua và cài đặt các chương trình GIS trên máy tính mà có thể tiến hành phân tích, tính toán, lưu giữ, chia sẻ thông tin môi trường trực tiếp qua chương trình được xây dựng chỉ với một máy tính có nối mạng Internet và một trình duyệt Web bất kỳ. Ngoài ra, người dùng còn có thể xem xét cách thức hoạt động của phần mềm để có thể hiệu chỉnh hay phát triển theo ý mình, có thể sử dụng dưới nhiều mục đích, dưới nhiều máy khác nhau hay phân phối lại tùy ý. Được xây dựng trên các hệ điều hành mã nguồn mở, dựa trên các cơ sở dữ liệu, dịch vụ Web và các kỹ thuật phát triển phần mềm, ngày nay chúng ta tìm thấy những hệ thống mã

nguồn mở tồn tại vững chắc trong lĩnh vực ứng dụng GIS. Những hệ thống này có thể là những hệ quản trị cơ sở dữ liệu không gian (như PostGIS), môi trường phân tích dữ liệu (như GRASS), kỹ thuật máy chủ cho Web (như MapServer, Geoserver, Deegree), những công cụ cho máy trạm (như MapBuilder, MapBender), những công cụ GIS chuyên nghiệp cho máy desktop thông thường (như gvSIG).

Phần lớn những phần mềm mã nguồn mở về GIS đáp ứng cả nhu cầu của cộng đồng mã nguồn mở lẫn các công ty tư nhân để phát triển, tích hợp, hỗ trợ kỹ thuật và đào tạo. Nhờ vào tính mở của các phần mềm loại này, các công ty loại nhỏ và trung bình có thể dễ dàng cung cấp các giải pháp và dịch vụ của họ sau khi phát triển thêm từ các mã nguồn mở. Để có thể phát triển các ứng dụng thông tin địa lý trên các phần mềm mã nguồn mở đòi hỏi các lập trình viên có kỹ thuật cao và nhiều kinh nghiệm. Đòi hỏi sự hiểu biết sâu về công nghệ Web, ngôn ngữ Web cũng như các hệ thống dữ liệu địa lý. Các phần mềm GIS thương mại hiện nay được đa phần người dùng GIS chuyên nghiệp sử dụng. Trong đó, đáng chú ý là các sản phẩm của hãng ESRI, như:

- Những phần mềm ArcGIS Desktop cho phép tạo, phân tích, vẽ bản đồ, quản lý, chia sẻ và xuất bản thông tin địa lý.
- Những module mở rộng của ArcGIS như Spatial Analyst, 3D Analyst, Network Analyst ...
- ArcSDE là phần mềm cho máy chủ cho phép nhiều người dùng truy cập vào hệ cơ sở dữ liệu không gian địa lý được lưu trữ trong hệ thống quản lý cơ sở dữ liệu quan hệ.
- ArcIMS là giải pháp cung cấp những bản đồ động, dữ liệu GIS và dịch vụ qua Web.
- MapObject, ArcEngine, ArcGIS Server là những công cụ để phát triển các ứng dụng GIS. Sản phẩm của ESRI với ưu điểm là xây dựng hệ thống Web – GIS nhanh, có thể kết hợp với các công nghệ nền khác, với các hợp phần do người lập

trình tự phát triển và bản thân những phần mềm của hãng ESRI có những công cụ mà các phần mềm GIS mã nguồn mở chưa phát triển được. Tuy nhiên, chi phí chuyển giao công nghệ còn cao và chỉ phù hợp với các dự án lớn

WebGIS được ứng dụng ngày càng nhiều ở các nước phát triển như: Hoa Kỳ, Nhật Bản,... các ứng dụng WebGIS đưa vào thực tế mang lại nhiều hiệu quả về kinh tế - xã hội – môi trường cho nhiều cường quốc, trong đó có hệ thống WebGIS hỗ trợ phát triển cộng đồng tại tỉnh Shimane - Nhật Bản, được phát triển bởi Trung tâm nghiên cứu vùng núi (Mountainous Region Research Center – MRRC).

1.3 Tình hình phát triển công nghệ WebGIS ở Việt Nam

Tại Việt Nam tình hình nghiên cứu và ứng dụng WebGIS trong những năm gần đây luôn được quan tâm, các đề tài có giá trị như: Ứng dụng công nghệ WebGIS để xây dựng cơ sở dữ liệu phục vụ khai thác tiềm năng du lịch bền vững tỉnh Quảng Trị, Ứng dụng công nghệ mã nguồn mở xây dựng WebGIS phục vụ công tác tư vấn địa điểm thi đại học – cao đẳng trên địa bàn Thành phố Hồ Chí Minh,... WebGIS phát triển mạnh ở Việt Nam đang được nhiều người sử dụng có thể kể đến là Việt bản đồ www.vietbando.vn.

Các ứng dụng phổ biến của WEBGIS tại Việt Nam có thể kể đến như: Trong giao thông, quản lý đô thị, quản lý bán hàng... Tuy nhiên rất ít các trang WEBGIS hỗ trợ người dùng thực hiện thao tác Geoprocessing ngay trên môi trường web giống như hệ thống của đề tài.

CHƯƠNG 2 THIẾT KẾT CẤU TRÚC VÀ XÂY DỰNG NỘI DUNG CỦA HỆ THỐNG WEBGIS

2.1 Phân tích nền tảng công nghệ

2.1.1 Ngôn ngữ lập trình Java

Java là một ngôn ngữ lập trình dạng lập trình hướng đối tượng (OOP). Khác với phần lớn ngôn ngữ lập trình thông thường, thay vì biên dịch mã nguồn thành mã máy hoặc thông dịch mã nguồn khi chạy, Java được thiết kế để biên dịch mã nguồn thành bytecode, bytecode sau đó sẽ được môi trường thực thi (runtime environment) chạy.

Trước đây, Java chạy chậm hơn những ngôn ngữ dịch thẳng ra mã máy như C và C++, nhưng sau này nhờ công nghệ "biên dịch tại chỗ" - Just in time compilation, khoảng cách này đã được thu hẹp, và trong một số trường hợp đặc biệt Java có thể chạy nhanh hơn. Java chạy nhanh hơn những ngôn ngữ thông dịch như Python, Perl, PHP gấp nhiều lần. Java chạy tương đương so với C#, một ngôn ngữ khá tương đồng về mặt cú pháp và quá trình dịch/chạy

Cú pháp Java được vay mượn nhiều từ C & C++ nhưng có cú pháp hướng đối tượng đơn giản hơn và ít tính năng xử lý cấp thấp hơn. Do đó việc viết một chương trình bằng Java dễ hơn, đơn giản hơn, đỡ tốn công sửa lỗi hơn. Lập trình C rất hay xảy ra lỗi và khó sửa.

Trong Java, hiện tượng dò rỉ bộ nhớ hầu như không xảy ra do bộ nhớ được quản lý bởi Java Virtual Machine (JVM) bằng cách tự động "dọn dẹp rác" - Garbage collection. Người lập trình không phải quan tâm đến việc cấp phát và xóa bộ nhớ như C, C++. Tuy nhiên khi sử dụng những tài nguyên mạng, file IO, database (nằm ngoài kiểm soát của JVM) mà người lập trình không đóng (close) các streams thì memory leak vẫn có thể xảy ra.

2.1.2 Thư viện mã nguồn mở OpenCV

OpenCV là một thư viện mã nguồn mở phục vụ cho việc nghiên cứu hay phát triển về thị giác máy tính. Tối ưu hóa và xử lý các ứng dụng trong thời gian

thực. Giúp cho việc xây dựng các ứng dụng xử lý ảnh, thị giác máy tính, ... một cách nhanh hơn. OpenCV có hơn 500 hàm khác nhau, được chia làm nhiều phần phục vụ các công việc như: xử lý hình ảnh vệ tinh, y tế, an ninh, camera quan sát, nhận diện, robots, ...

OpenCV là mã nguồn mở và chúng ta không phải trả bất phí. Tất cả chúng ta đều có thể sử dụng OpenCV tùy theo việc chúng ta sử dụng nó như thế nào vào việc gì.

2.1.3 Cơ sở dữ liệu đồ họa PostGis

PostGIS là một cơ sở dữ liệu đặc biệt mở rộng của PostgreSQL. Nó hỗ trợ tất cả các function và các object định nghĩa trong OpenGIS “Simple Feature For SQL”. Giống như Oracle, DB2 và SQL server, PostGIS có khả năng tạo một cơ sở dữ liệu quan hệ. PostGIS có thể đổi lại tên như “PostgreSQL Spatial”. Nó được mở rộng hơn với những khả năng:

Mở rộng lưu trữ kiểu dữ liệu hình học cho các kiểu dữ liệu thường dùng (varchar, char, integer, date...).

Thêm vào các functions cho kiểu hình học (geometry) và cung cấp các thông tin hữu ích .

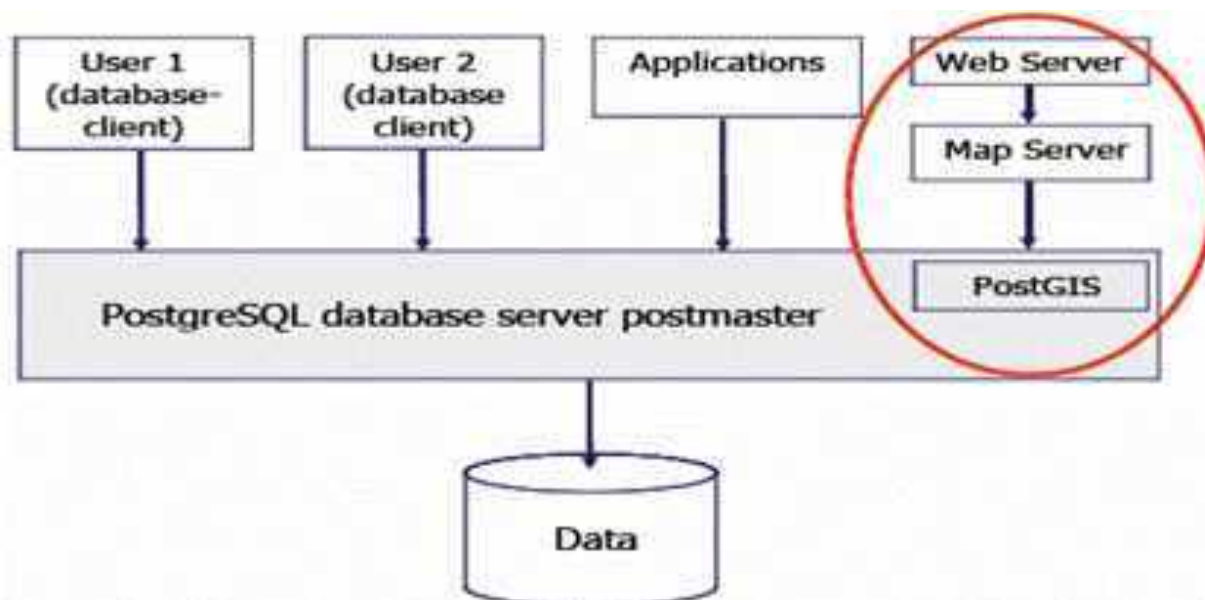
ST_Distance(geometry,geometry), ST_Area(geometry), ST_Length(geometry), ST_Intersects(geometry, geometry...)

Cơ chế đánh chỉ số cho các câu truy vấn với không gian giới hạn, trả về các record nhanh chóng từ các bảng dữ liệu khổng lồ.

Những chức năng cơ bản của cơ sở dữ liệu không gian dễ dàng được liệt kê như: kiểu, hàm, chỉ mục. Điều quan trọng nhất là khối lượng của việc xử lý không gian bên trong cơ sở dữ liệu khi những khả năng được giới thiệu: chiếu lại (re_projection), những phân tích lớn từ bản dữ liệu, tìm kiếm xấp xỉ, lọc thuộc tính, và nhiều hơn nữa.

Trên thực tế, PostGIS là một hợp phần quản lý thông tin không gian của PostgreSQL, nó cho phép quản trị CSDL không gian dùng trong hệ thống thông tin

địa lý (GIS), giống như SDE của ESRI hoặc Oracle's Spatial extension. PostGIS hỗ trợ cho OpenGIS vì "tính năng kỹ thuật SQL đơn giản" và đã được chứng nhận phù hợp với "Kiêu và Chức năng" của OpenGIS.



Hình 2.1 Vị trí của PostGIS trong PostgreSQL

Một số ưu điểm khi sử dụng CSDL PostGIS

PostGIS là một CSDL không gian có những ưu điểm để xử lý các thông tin về hình dạng không gian như trả lời các truy vấn về những đối tượng ở gần một vị trí nào đó, những đối tượng nằm trong phạm vi hoặc ở vùng phụ cận của một đối tượng khác, phạm vi của một vùng nơi mà có một hoạt động nào đó đang xảy ra là gì, những đối tượng nào nằm bên trong một đối tượng khác.

PostGIS có khả năng lưu trữ và thao tác với dữ liệu rất tốt. Nó cung cấp những khả năng xử lý thông tin địa lý bên trong một môi trường cơ sở dữ liệu. Những hàm SQL bao gồm buffer, intersection, within, distance... Những hàm này lấy dữ liệu hình học từ các cột trong bảng PostGIS và trả về những hình học mới hoặc những thông tin khác. Ví dụ như hàm distance sẽ tính toán khoảng cách giữa các đặc điểm (feature) không gian, và hàm sẽ trả về một hình mới là một đa giác được làm vùng đệm tại một khoảng cách nào đó từ feature nguồn.

PostGIS cho phép dễ dàng khi kết nối dữ liệu không gian với dữ liệu phi không gian trong một môi trường dữ liệu không gian và cung cấp đầy đủ sức mạnh

của ngôn ngữ truy vấn cấu trúc (SQL) để thực hiện những phân tích khác chẳng hạn như tính chi phí trung bình cho một vùng địa lý?, ...

PostGIS còn hỗ trợ thêm GiST-based and R-Tree spatial indexes, đây là điểm riêng biệt nhất của PostGIS so với các hệ quản trị CSDL khác giúp PostGIS dễ dàng quản lý, lưu trữ, khai thác và tìm kiếm thông tin không gian.

Một số đối tượng không gian theo chuẩn OGC trong PostGIS

PostGIS hỗ trợ các đối tượng và các hàm được định nghĩa chi tiết trong OGC "Simple Features for SQL". Với PostGIS ta có thể lưu trữ các điểm, đường, vùng, tập hợp điểm, tập hợp đường, tập hợp vùng và các thông tin hình học khác. PostGIS mở rộng chuẩn với sự hỗ trợ cho các tọa độ dạng 3DZ, 3DM và 4D.

OpenGIS định nghĩa 2 chuẩn để diễn tả các đối tượng không gian: Well-known Text (WKT) và Well-Known Binary (WKB). Cả Well-known Text và Well-Known Binary đều bao gồm thông tin về kiểu của đối tượng và tọa độ của nó.

PostGIS được hiện thực theo bản đặc tả "OGC Simple Feature Specifications" cho chuẩn SQL. Bản đặc tả này định nghĩa những thuật toán và lược đồ SQL để thêm (insert), truy vấn (query), thao tác (manipulate) và xóa (delete) những đối tượng không gian. Những tọa độ của các đối tượng không gian được lưu trữ trong các bảng Feature. Mỗi một bảng Feature có thể chứa một kiểu đối tượng geometry là điểm (point, multipoint), đường (linestring, multilinestring), đa giác (polygon, multipolygon và geometry collection). Những tọa độ của mỗi đối tượng được lưu trữ trong một trường với một kiểu riêng biệt.

Kiểu mô tả một tập hợp tọa độ không gian của đối tượng là WKT (Well Known Text). Khi những đối tượng không gian được đưa vào cơ sở dữ liệu thì PostGIS sẽ chuyển chúng từ WKT sang WKB (Well Known Binary) để việc lưu trữ được tốt hơn.

Ví dụ về miêu tả WKT:

POINT(0 0 0) -- XYZ

SRID=32632;POINT(0 0) -- XY with SRID

POINTM(0 0 0) -- XYM

POINT(0 0 0 0) -- XYZM

SRID=4326;MULTIPOINTM(0 0 0,1 2 1) -- XYM with SRID

MULTILINESTRING(((0 0 0,1 1 0,1 2 1),(2 3 1,3 2 1,5 4 1))

POLYGON(((0 0 0,4 0 0,4 4 0,0 4 0,0 0 0),(1 1 0,2 1 0,2 2 0,1 2 0,1 1 0))

MULTIPOLYGON((((0 0 0,4 0 0,4 4 0,0 4 0,0 0 0),(1 1 0,2 1 0,2 2 0,1 2 0,1 1 0)),((-1 -1 0,-1 -2 0,-2 -2 0,-2 -1 0,-1 -1 0)))

GEOMETRYCOLLECTIONM(POINTM(2 3 9), LINESTRINGM(2 3 4, 3 4 5))

Siêu dữ liệu từ mỗi bảng Feature được tập hợp lại và lưu trữ trong bảng `geometry_columns` để tổ chức kiểu và hệ thống tọa độ của dạng geometry được chứa trong đó.

2.1.4 Máy chủ Geoserver

GeoServer 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. Được kỳ vọng sẽ trở thành một phương thức đơn giản để kết nối những nguồn thông tin có sẵn từ Google Earth, NASA World Wind nhằm tạo ra các dịch vụ Webmap như Google Maps, Windows Live Local và Yahoo Maps.

GeoServer được viết bằng ngôn ngữ Java, cho phép người sử dụng chia sẻ và chỉnh sửa dữ liệu không gian địa lý (geospatial data).

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 là thành phần nền tảng của Geospatial Web.

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.

Các đặc trưng của Geoserver:

GeoServer cho phép xuất dữ liệu linh hoạt dựa vào việc hỗ trợ các chuẩn KML, GML, Shapefile, GeoRSS, Portable Document Format, GeoJSON, JPEG, GIF, SVG, PNG...

GeoServer có thể đọc được nhiều định dạng dữ liệu, bao gồm PostGIS, Oracle Spatial, ArcSDE, DB2, MySQL, Shapefiles, GeoTIFF, GTOPO30 và nhiều loại khác. Bên cạnh đó, GeoServer còn có thể chỉnh sửa dữ liệu nhờ những thành phần xử lý của Chuẩn Web Feature Server.

GeoServer được xây dựng trong bộ GeoTools, được viết bởi ngôn ngữ Java.

MapServer cũng được coi là một sản phẩm mã nguồn mở, có nhiều chức năng tương tự. Sự khác biệt giữa hai sản phẩm là Mapserver được phát triển trong môi trường cũ, được viết bởi ngôn ngữ C và hoạt động nhờ CGI, còn GeoServer được viết bằng ngôn ngữ Java. GeoServer phân biệt với MapServer bởi nó có giao diện đồ họa, giúp đơn giản hơn trong việc cấu hình, và thực thi chức năng sửa đổi dựa vào Web Feature Server, cho phép chỉnh sửa thông tin không gian cả trên Web cũng như trên máy trạm Desktop. Ưu điểm lớn nhất của MapServer là tốc độ thực thi nhanh hơn GeoServer, nhưng từ phiên bản 1.6 trở đi của GeoServer thì tốc độ của hai sản phẩm đã tương đương.

GeoServer hỗ trợ việc chia sẻ dữ liệu không gian địa lý lên Google Earth thông qua đặc tính 'network link' sử dụng KML.

2.1.5 JavaScript, JQuery và thư viện mã nguồn mở OpenLayers

JavaScript, theo phiên bản hiện hành, là một ngôn ngữ lập trình kịch bản dựa trên đối tượng được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ này được dùng rộng rãi cho các trang web, nhưng cũng được dùng để tạo khả năng viết script sử dụng các đối tượng nằm sẵn trong các ứng dụng. Nó vốn được phát triển bởi Brendan Eich tại Hãng truyền thông Netscape với cái tên đầu tiên Mocha, rồi sau đó đổi tên thành LiveScript, và cuối cùng thành JavaScript. Giống Java, JavaScript có cú pháp tương tự C, nhưng nó gần với Self hơn Java. .js là phần mở rộng thường được dùng cho tập tin mã nguồn JavaScript.

Phiên bản mới nhất của JavaScript là phiên bản 1.5, tương ứng với ECMA-262 bản 3. ECMAScript là phiên bản chuẩn hóa của JavaScript. Trình duyệt Mozilla phiên bản 1.8 beta 1 có hỗ trợ không đầy đủ cho E4X - phần mở rộng cho JavaScript hỗ trợ làm việc với XML, được chuẩn hóa trong ECMA-357.

JQuery thư viện JavaScript đa trình duyệt được thiết kế để đơn giản hóa lập trình phía máy người dùng của HTML, phát hành vào tháng 1 năm 2006 tại BarCamp NYC bởi John Resig. Được sử dụng bởi hơn 52% trong 10.000 truy cập nhiều nhất các trang web, jQuery là thư viện JavaScript phổ biến nhất được sử dụng ngày nay

jQuery miễn phí, mã nguồn mở phần mềm, kép cấp phép theo MIT Giấy phép GNU General Public License, phiên bản 2 jQuery của được để làm cho nó dễ dàng hơn để di chuyển một tài liệu, chọn DOM các yếu tố, tạo ra hoạt hình s, xử lý Sự kiện, và phát triển Ajax ứng dụng. jQuery cũng cung cấp khả năng cho các nhà phát triển để tạo ra plug-in s trên đầu trang của thư viện JavaScript. Điều này cho phép các nhà phát triển để tạo ra trừu tượng hóa ở mức độ thấp tương tác và hình ảnh động, hiệu ứng tiên tiến và vật dụng cao cấp, chủ đề có thể. Cách tiếp cận mô-đun để thư viện jQuery cho phép tạo ra các công cụ mạnh mẽ và năng động web và các ứng dụng web.

OpenLayers là một dự án của tổ chức OGC. OpenLayers là một thư viện điện tử mã nguồn mở JavaScript 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 phát triển và hỗ trợ bởi rất nhiều tổ chức trên thế giới. OpenLayers cho phép hiển thị bản đồ và các điểm đánh dấu (marker) lên 1 trang web từ nhiều nguồn khác nhau.

Hỗ trợ việc hiển thị dữ liệu bản đồ trên mọi trình duyệt hiện nay. Hỗ trợ việc xây dựng các ứng dụng địa lý trên web. Hỗ trợ các phương pháp chuẩn cho việc truy cập dữ liệu địa lý, ví dụ như 2 giao thức WMS và WFS của OGC

Viết hoàn toàn bằng JavaScript. OpenLayers được viết bằng JavaScript theo hướng đối tượng, sử dụng các thành phần từ Prototype.js và thư viện Rico. OpenLayers tách rời phần công cụ bản đồ và dữ liệu bản đồ. Nhờ đó mọi công cụ đều có thể hoạt động trên các nguồn dữ liệu khác nhau.

OpenLayers cho phép người dùng hiển thị nhiều layer khác nhau từ nhiều nguồn khác nhau cùng 1 lúc. Các nguồn này có thể là 1 WMS, WFS hay những dịch vụ bản đồ web mở khác như GeoRSS, OpenStreetMap, Google Maps/Earth hay các file dữ liệu như GML, KML....

OpenLayers là hoàn toàn miễn phí, mã nguồn mở JavaScript. Không phải trả phí và phức tạp như Google Maps API (Nếu sử dụng Google Maps API thì máy bạn phải kết nối Internet nếu không thì bạn phải trả phí để sử dụng).

2.2 Phương pháp thiết kế hệ thống

Tầm quan trọng của mô hình đã được lĩnh hội một cách thấu đáo trong hầu như tất cả các ngành khoa học kỹ thuật từ nhiều thế kỷ nay. Bất kỳ ở đâu, khi muốn xây dựng một vật thể nào đó, đầu tiên người ta đã tạo nên các bản vẽ để quyết định cả ngoại hình lẫn phương thức hoạt động của nó. Chẳng hạn các bản

vẽ kỹ thuật thường gặp là một dạng mô hình quen thuộc. Mô hình nhìn chung là một cách mô tả của một vật thể nào đó. Vật đó có thể tồn tại trong một số giai đoạn nhất định, dù đó là giai đoạn thiết kế hay giai đoạn xây dựng hoặc chỉ là một kế hoạch. Nhà thiết kế cần phải tạo ra các mô hình mô tả tất cả các khía cạnh khác nhau của sản phẩm. Ngoài ra, một mô hình có thể được chia thành nhiều hướng nhìn, mỗi hướng nhìn trong số chúng sẽ mô tả một khía cạnh riêng biệt của sản phẩm hay hệ thống cần được xây dựng. Một mô hình cũng có thể được xây dựng trong nhiều giai đoạn và ở mỗi giai đoạn, mô hình sẽ được bổ sung thêm một số chi tiết nhất định.

Mô hình thường được mô tả trong ngôn ngữ trực quan, điều đó có nghĩa là đa phần các thông tin được thể hiện bằng các ký hiệu đồ họa và các kết nối giữa chúng, chỉ khi cần thiết một số thông tin mới được biểu diễn ở dạng văn bản; Theo đúng như câu ngạn ngữ "Một bức tranh nói nhiều hơn cả ngàn từ". Tạo mô hình cho các hệ thống phần mềm trước khi thực sự xây dựng nên chúng, đã trở thành một chuẩn mực trong việc phát triển phần mềm và được chấp nhận trong cộng đồng làm phần mềm giống như trong bất kỳ một ngành khoa học kỹ thuật nào khác. Việc biểu diễn mô hình phải thỏa mãn các yếu tố sau:

- Chính xác (accurate): Mô tả đúng hệ thống cần xây dựng.
- Đồng nhất (consistent): Các view khác nhau không được mâu thuẫn với nhau.
- Có thể hiểu được (understandable): Cho những người xây dựng lần sử dụng
- Dễ thay đổi (changeable)
- Dễ dàng liên lạc với các mô hình khác.

Có thể nói thêm rằng mô hình là một sự đơn giản hoá hiện thực. Mô hình được xây dựng nên để chúng ta dễ dàng hiểu và hiểu tốt hơn hệ thống cần xây dựng. Tạo mô hình sẽ giúp cho chúng ta hiểu thấu đáo một hệ thống phức tạp trong sự toàn thể của nó.

Nói tóm lại, mô hình hóa một hệ thống nhằm mục đích:

- Hình dung một hệ thống theo thực tế hay theo mong muốn của chúng ta .
- Chỉ rõ cấu trúc hoặc ứng xử của hệ thống.
- Tạo một khuôn mẫu hướng dẫn nhà phát triển trong suốt quá trình xây dựng hệ thống.
- Ghi lại các quyết định của nhà phát triển để sử dụng sau này.

2.2.1 Trước khi UML ra đời:

Đầu những năm 1980, ngành công nghệ phần mềm chỉ có duy nhất một ngôn ngữ hướng đối tượng là Simula. Sang nửa sau của thập kỷ 1980, các ngôn ngữ hướng đối tượng như Smalltalk và C++ xuất hiện. Cùng với chúng, nảy sinh nhu cầu mô hình hoá các hệ thống phần mềm theo hướng đối tượng. Và một vài trong số những ngôn ngữ mô hình hoá xuất hiện những năm đầu thập kỷ 90 được nhiều người dùng là:

- Grady Booch's Booch Modeling Methodology
- James Rumbaugh's Object Modeling Technique – OMT
- Ivar Jacobson's OOSE Methodology
- Hewlett- Packard's Fusion
- Coad and Yordon's OOA and OOD

Mỗi phương pháp luận và ngôn ngữ trên đều có hệ thống ký hiệu riêng, phương pháp xử lý riêng và công cụ hỗ trợ riêng, khiến nảy ra cuộc tranh luận phương pháp nào là tốt nhất. Đây là cuộc tranh luận khó có câu trả lời, bởi tất cả các phương pháp trên đều có những điểm mạnh và điểm yếu riêng. Vì thế, các nhà phát triển phần mềm nhiều kinh nghiệm thường sử dụng phối hợp các điểm mạnh của mỗi phương pháp cho ứng dụng của mình. Trong thực tế, sự khác biệt giữa các phương pháp đó hầu như không đáng kể và theo cùng tiến trình thời gian, tất cả những phương pháp trên đã tiệm cận lại và bổ sung lẫn cho nhau. Chính hiện thực này đã được những người tiên phong trong lĩnh vực mô hình hoá hướng đối tượng nhận ra và họ quyết định ngồi lại cùng nhau để tích hợp những điểm

manh của mỗi phương pháp và đưa ra một mô hình thống nhất cho lĩnh vực công nghệ phần mềm.

2.2.2 Sự ra đời của UML:

Trong bối cảnh trên, người ta nhận thấy cần thiết phải cung cấp một phương pháp tiệm cận được chuẩn hoá và thống nhất cho việc mô hình hoá hướng đối tượng. Yêu cầu cụ thể là đưa ra một tập hợp chuẩn hoá các ký hiệu (Notation) và các biểu đồ (Diagram) để nắm bắt các quyết định về mặt thiết kế một cách rõ ràng, rành mạch. Đã có ba công trình tiên phong nhắm tới mục tiêu đó, chúng được thực hiện dưới sự lãnh đạo của James Rumbaugh, Grady Booch và Ivar Jacobson. Chính những cố gắng này dẫn đến kết quả là xây dựng được một Ngôn Ngữ Mô Hình Hoá Thống Nhất (Unified Modeling Language – UML).

UML là một ngôn ngữ mô hình hoá thống nhất có phần chính bao gồm những ký hiệu hình học, được các phương pháp hướng đối tượng sử dụng để thể hiện và miêu tả các thiết kế của một hệ thống. Nó là một ngôn ngữ để đặc tả, trực quan hoá, xây dựng và làm sơ liệu cho nhiều khía cạnh khác nhau của một hệ thống có nồng độ phần mềm cao. UML có thể được sử dụng làm công cụ giao tiếp giữa người dùng, nhà phân tích, nhà thiết kế và nhà phát triển phần mềm.

Trong quá trình phát triển có nhiều công ty đã hỗ trợ và khuyến khích phát triển UML có thể kể tới như : Hewlett Packard, Microsoft, Oracle, IBM, Unisys.

2.2.3 UML (Unified Modeling Language):

Ngôn ngữ mô hình hóa thống nhất (Unified Modeling Language – UML) là một ngôn ngữ để biểu diễn mô hình theo hướng đối tượng được xây dựng bởi ba tác giả trên với chủ đích là:

- Mô hình hoá các hệ thống sử dụng các khái niệm hướng đối tượng.
- Thiết lập một kết nối từ nhận thức của con người đến các sự kiện cần mô hình hoá.
- Giải quyết vấn đề về mức độ thừa kế trong các hệ thống phức tạp, có nhiều ràng buộc khác nhau.

- Tạo một ngôn ngữ mô hình hoá có thể sử dụng được bởi người và máy.

2.2.4 Phương pháp và các ngôn ngữ mô hình hoá:

Phương pháp hay phương thức (method) là một cách trực tiếp cấu trúc hoá sự suy nghĩ và hành động của con người. Phương pháp cho người sử dụng biết phải làm gì, làm như thế nào, khi nào và tại sao (mục đích của hành động). Phương pháp chứa các mô hình (model), các mô hình được dùng để mô tả những gì sử dụng cho việc truyền đạt kết quả trong quá trình sử dụng phương pháp. Điểm khác nhau chính giữa một phương pháp và một ngôn ngữ mô hình hoá (modeling language) là ngôn ngữ mô hình hoá không có một tiến trình (process) hay các câu lệnh (instruction) mô tả những công việc người sử dụng cần làm.

Một mô hình được biểu diễn theo một ngôn ngữ mô hình hoá. Ngôn ngữ mô hình hoá bao gồm các ký hiệu – những biểu tượng được dùng trong mô hình – và một tập các quy tắc chỉ cách sử dụng chúng. Các quy tắc này bao gồm:

- Syntactic (Cú pháp): cho biết hình dạng các biểu tượng và cách kết hợp chúng trong ngôn ngữ.
- Semantic (Ngữ nghĩa): cho biết ý nghĩa của mỗi biểu tượng, chúng được hiểu thế nào khi nằm trong hoặc không nằm trong ngữ cảnh của các biểu tượng khác.
- Pragmatic : định nghĩa ý nghĩa của biểu tượng để sao cho mục đích của mô hình được thể hiện và mọi người có thể hiểu được.

CHƯƠNG 3. ĐÁNH GIÁ, LỰA CHỌN CÔNG NGHỆ VÀ PHÁT TRIỂN HỆ THỐNG

3.1 Khái quát chung

Ứng dụng Web-GIS quản lý và sử dụng nguồn nước lưu vực sông Cả là một ứng dụng cho phép nhập dữ liệu ảnh vệ tinh và các dữ liệu chuyên đề khác vào trong cơ sở dữ liệu và tiến hành hiển thị kết quả tổng hợp bằng hình ảnh lên trang web bản đồ.

Trên giao diện bản đồ web, người dùng sẽ nhìn thấy các bản đồ chuyên đề cũng như các sản phẩm khác của đề tài được biểu diễn bằng những bản đồ, đồ thị, bảng biểu khác nhau, đồng thời có các chức năng thống kê, thao tác phân tích xử lý dữ liệu thông qua giao diện web.

- Mục đích

Xuất phát từ nhu cầu và mục tiêu quan trọng của Đề tài, mục đích của chuyên đề Nghiên cứu đánh giá lựa chọn công nghệ nền cần đánh giá các công nghệ nền hiện nay, đưa ra các ưu nhược điểm của từng Công nghệ để lựa chọn công nghệ phù hợp chung cho toàn Hệ thống.

- Yêu cầu

- Nghiên cứu các công nghệ nền hiện nay
- Đánh giá ưu nhược điểm của các công nghệ
- Lựa chọn công nghệ áp dụng cho xây dựng phần mềm

3.2 Phương pháp nghiên cứu và cách tiếp cận

Để trả lời các hỏi nghiên cứu, chuyên đề đã sử dụng một số phương pháp nghiên cứu và tiếp cận như sau:

3.2.1 Công nghệ Nền là gì?

Công nghệ Nền là một thuật ngữ công nghệ cho phép tạo ra các sản phẩm, quy trình hỗ trợ phát triển trong quá khứ, hiện tại và tương lai. Nó thiết lập các khả năng lâu dài cho việc nghiên cứu và phát triển. Nó có thể được định nghĩa là

một cấu trúc hoặc công nghệ mà các sản phẩm khác có thể hiển thị lên không đòi hỏi các chi phí khi xuất hiện quá trình/ công nghệ mới.

Trong nền tảng máy tính, ví dụ, phần cứng máy tính phục vụ như là nền tảng của một hệ điều hành do đó là một nền tảng cho các phần mềm hệ điều hành cũng như các phần mềm ứng dụng đó hoạt động được. Cơ sở hạ tầng giao thông vận tải phục vụ như là một nền tảng cho các phương tiện.

Một nền tảng công nghệ làm tăng sự dễ dàng trong sản xuất. Phần ít hơn các cụm chi tiết cần phải được thiết kế, thực hiện và lưu trữ, khi xây dựng không cần quá nhiều đào tạo.

3.2.2 Căn cứ kiến trúc hệ thống

Kiến trúc tổng thể của hệ thống có định hướng quản lý dữ liệu các ngành dạng phân tán. Do vậy khi chúng ta lựa chọn công nghệ nền áp dụng phải đảm bảo được yêu cầu này.

3.2.3 Căn cứ các giải pháp thu thập, cập nhật dữ liệu dữ liệu

Thông qua giải pháp này, dễ nhận thấy dữ liệu được quản lý phân tán và những người dùng được khai thác là những người được phân quyền trong hệ thống. Những loại dữ liệu trong quá trình thu thập có thể rất lớn. Đó là một vấn đề cần được xem xét trong khi lựa chọn công nghệ.

3.2.4 Căn cứ các giải pháp lưu trữ dữ liệu không gian và dữ liệu thuộc tính

Cũng như những yêu cầu trên, lưu trữ dữ liệu lớn mang tính chất phân tán phải đòi hỏi cơ sở hạ tầng thực sự mạnh. Hơn thế nữa giúp định hướng việc sử dụng hệ quản trị cơ sở dữ liệu nào có thể đáp ứng được những vấn đề trên. Hiện nay có các loại hệ quản trị cơ sở dữ liệu lớn như: SQL Server của Microsoft, DB2 của IBM và Oracle của Oracle là 3 hệ quản trị CSDL lớn điển hình. Oracle là hệ quản trị mà theo như nhận xét hiện nay cho phép lưu trữ được dữ liệu không gian và phi không gian rất lớn và quản lý dữ liệu phân tán có độ an toàn bảo mật cao. Bên cạnh đó cũng có nhiều Hệ quản trị CSDL mã nguồn mở như MySQL,

PostGres là lựa chọn tốt cho công tác nghiên cứu nhờ tiết kiệm chi phí và khả năng tùy chỉnh tốt.

3.2.5 Căn cứ các giải pháp kỹ thuật phân phối dữ liệu

Phân phối dữ liệu là một trong những thành phần không thể thiếu trong hệ thống cơ sở dữ liệu. Dịch vụ này hỗ trợ và chia sẻ dữ liệu giữa các ngành với nhau. Trong thực tế, có nhiều kỹ thuật phân phối dữ liệu tùy theo loại dữ liệu đó là gì. Nếu dữ liệu có kích thước lớn, chúng ta có thể cung cấp dưới dạng các tệp tin có cấu trúc (như dạng mdb, gdb hoặc cấu trúc tệp tin của định dạng shape...), cho phép người dùng tải dữ liệu, ngược lại nếu kích thước dữ liệu nhỏ thì chúng ta có thể cung cấp dưới dạng các dịch vụ thông tin, hoặc cũng có thể là các tệp tin thuộc tính có kích thước nhỏ.

3.2.6 Căn cứ các yêu cầu về tính an ninh, bảo mật của hệ thống, cơ sở hạ tầng phục vụ hệ thống và quy trình công nghệ phát triển hệ thống

Cơ sở dữ liệu đã được kiểm tra, đánh giá, xử lý, tích hợp và được lưu trữ một cách có hệ thống, có tổ chức dưới dạng tệp dữ liệu lưu trên các hệ thống tin học, các thiết bị lưu trữ và các vật mang tin như các loại ổ cứng máy tính, băng từ, đĩa CD, DVD... hoặc văn bản, tài liệu được xây dựng, cập nhật và duy trì phục vụ quản lý nhà nước và các hoạt động kinh tế, xã hội, quốc phòng, an ninh, nghiên cứu khoa học, giáo dục đào tạo và nâng cao dân trí trên địa bàn cả nước.

Các chức năng của hệ quản trị CSDL có ảnh hưởng đến công tác an ninh, bảo mật của hệ thống bao gồm:

- **Triggers:** các trigger được sử dụng để lấy dữ liệu đến từ các câu lệnh INSERT, UPDATE (để mã hóa).
- **Views:** các view được sử dụng để lấy dữ liệu đến từ các câu lệnh SELECT (để giải mã)
- **Extended Stored Procedures:** được gọi từ các Triggers hoặc Views dùng để kích hoạt các dịch vụ được cung cấp bởi module DBPEM từ trong môi trường của hệ quản trị CSDL.

- **DBPEM (Database Policy Enforcing Module):** cung cấp các dịch vụ mã hóa/giải mã dữ liệu gửi đến từ các Extended Stored Procedures và thực hiện việc kiểm tra quyền truy xuất của người dùng (dựa trên các chính sách bảo mật được lưu trữ trong CSDL về quyền bảo mật).
- **Security Database:** lưu trữ các chính sách bảo mật (policy) (quy định người dùng nào trong hoàn cảnh nào (thời gian, sử dụng chương trình gì, sử dụng máy tính nào, ...) có thể “thấy” được trường (field) nào trong CSDL) và các khóa giải mã. Xu hướng ngày nay thường là lưu trữ CSDL về bảo mật này trong Active Directory (một CSDL dạng thư mục để lưu trữ tất cả thông tin về hệ thống mạng) để tiện tích hợp vào cơ sở hạ tầng của các doanh nghiệp.
- **Security Services:** chủ yếu thực hiện việc bảo vệ các khóa giải mã được lưu trong CSDL bảo mật.
- **Management Console:** Dùng để cập nhật thông tin lưu trong CSDL bảo mật (chủ yếu là soạn thảo các chính sách bảo mật) và thực hiện thao tác bảo vệ một trường (field) nào đó trong CSDL của doanh nghiệp.

3.3 Kết quả đánh giá, lựa chọn và phát triển hệ thống

3.3.1 Công nghệ nền cho quản trị cơ sở dữ liệu

PostGIS là một cơ sở dữ liệu đặc biệt mở rộng của PostgreSQL. Nó hỗ trợ tất cả các function và các object định nghĩa trong OpenGIS “Simple Feature For SQL”. Giống như Oracle, DB2 và SQL server, PostGIS có khả năng tạo một cơ sở dữ liệu quan hệ. PostGIS có thể đổi lại tên như “PostgreSQL Spatial”. Nó được mở rộng hơn với những khả năng:

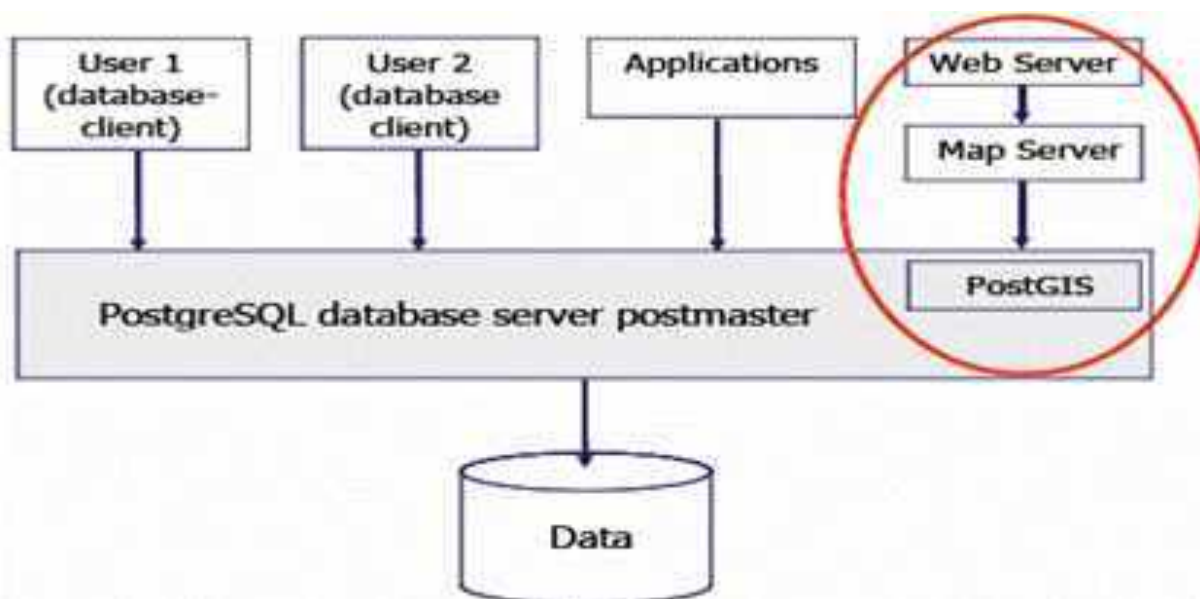
- Mở rộng lưu trữ kiểu dữ liệu hình học cho các kiểu dữ liệu thường dùng (varchar, char, integer, date...).
- Thêm vào các functions cho kiểu hình học (geometry) và cung cấp các thông tin hữu ích .

ST_Distance(geometry,geometry), ST_Area(geometry), ST_Length(geometry), ST_Intersects(geometry, geometry)...

- Cơ chế đánh chỉ số cho các câu truy vấn với không gian giới hạn, trả về các record nhanh chóng từ các bảng dữ liệu khổng lồ.

Những chức năng cơ bản của cơ sở dữ liệu không gian dễ dàng được liệt kê như: kiểu, hàm, chỉ mục. Điều quan trọng nhất là khối lượng của việc xử lý không gian bên trong cơ sở dữ liệu khi những khả năng được giới thiệu: chiếu lại (re_projection), những phân tích lớn từ bản dữ liệu, tìm kiếm xấp xỉ, lọc thuộc tính, và nhiều hơn nữa.

Trên thực tế, PostGIS là một hợp phần quản lý thông tin không gian của PostgreSQL, nó cho phép quản trị CSDL không gian dùng trong hệ thống thông tin địa lý (GIS), giống như SDE của ESRI hoặc Oracle's Spatial extension. PostGIS hỗ trợ cho OpenGIS vì "tính năng kỹ thuật SQL đơn giản" và đã được chứng nhận phù hợp với "Kiểu và Chức năng" của OpenGIS.



Hình 3.1 Vị trí của PostGIS trong PostgreSQL

Một số ưu điểm khi sử dụng CSDL PostGIS

PostGIS là một CSDL không gian có những ưu điểm để xử lý các thông tin về hình dạng không gian như trả lời các truy vấn về những đối tượng ở gần một vị trí nào đó, những đối tượng nằm trong phạm vi hoặc ở vùng phụ cận của một

đối tượng khác, phạm vi của một vùng nơi mà có một hoạt động nào đó đang xảy ra là gì, những đối tượng nào nằm bên trong một đối tượng khác.

PostGIS có khả năng lưu trữ và thao tác với dữ liệu rất tốt. Nó cung cấp những khả năng xử lý thông tin địa lý bên trong một môi trường cơ sở dữ liệu. Những hàm SQL bao gồm buffer, intersection, within, distance... Những hàm này lấy dữ liệu hình học từ các cột trong bảng PostGIS và trả về những hình học mới hoặc những thông tin khác. Ví dụ như hàm distance sẽ tính toán khoảng cách giữa các đặc điểm (feature) không gian, và hàm sẽ trả về một hình mới là một đa giác được làm vùng đệm tại một khoảng cách nào đó từ feature nguồn.

PostGIS cho phép dễ dàng khi kết nối dữ liệu không gian với dữ liệu phi không gian trong một môi trường dữ liệu không gian và cung cấp đầy đủ sức mạnh của ngôn ngữ truy vấn cấu trúc (SQL) để thực hiện những phân tích khác chẳng hạn như tính chi phí trung bình cho một vùng địa lý?, ...

PostGIS còn hỗ trợ thêm GiST-based and R-Tree spatial indexes, đây là điểm riêng biệt nhất của PostGIS so với các hệ quản trị CSDL khác giúp PostGIS dễ dàng quản lý, lưu trữ, khai thác và tìm kiếm thông tin không gian.

Một số đối tượng không gian theo chuẩn OGC trong PostGIS

PostGIS hỗ trợ các đối tượng và các hàm được định nghĩa chi tiết trong OGC "Simple Features for SQL". Với PostGIS ta có thể lưu trữ các điểm, đường, vùng, tập hợp điểm, tập hợp đường, tập hợp vùng và các thông tin hình học khác. PostGIS mở rộng chuẩn với sự hỗ trợ cho các tọa độ dạng 3DZ, 3DM và 4D.

OpenGIS định nghĩa 2 chuẩn để diễn tả các đối tượng không gian: Well-known Text (WKT) và Well-Known Binary (WKB). Cả Well-known Text và Well-Known Binary đều bao gồm thông tin về kiểu của đối tượng và tọa độ của nó.

PostGIS được hiện thực theo bản đặc tả "OGC Simple Feature Specifications" cho chuẩn SQL. Bản đặc tả này định nghĩa những thuật toán và lược đồ SQL để thêm (insert), truy vấn (query), thao tác (manipulate) và xóa

(delete) những đối tượng không gian. Những tọa độ của các đối tượng không gian được lưu trữ trong các bảng Feature. Mỗi một bảng Feature có thể chứa một kiểu đối tượng geometry là điểm (point, multipoint), đường (linestring, multilinestring), đa giác (polygon, multipolygon và geometry collection). Những tọa độ của mỗi đối tượng được lưu trữ trong một trường với một kiểu riêng biệt.

Kiểu mô tả một tập hợp tọa độ không gian của đối tượng là WKT (Well Known Text). Khi những đối tượng không gian được đưa vào cơ sở dữ liệu thì PostGIS sẽ chuyển chúng từ WKT sang WKB (Well Known Binary) để việc lưu trữ được tốt hơn.

Ví dụ về miêu tả WKT:

POINT(0 0 0) -- XYZ

SRID=32632;POINT(0 0) -- XY with SRID

POINTM(0 0 0) -- XYM

POINT(0 0 0 0) -- XYZM

SRID=4326;MULTIPOINTM(0 0 0,1 2 1) -- XYM with SRID

MULTILINESTRING((0 0 0,1 1 0,1 2 1),(2 3 1,3 2 1,5 4 1))

POLYGON((0 0 0,4 0 0,4 4 0,0 4 0,0 0 0),(1 1 0,2 1 0,2 2 0,1 2 0,1 1 0))

MULTIPOLYGON(((0 0 0,4 0 0,4 4 0,0 4 0,0 0 0),(1 1 0,2 1 0,2 2 0,1 2 0,1 1 0)),((-1 -1 0,-1 -2 0,-2 -2 0,-2 -1 0,-1 -1 0)))

GEOMETRYCOLLECTIONM(POINTM(2 3 9), LINESTRINGM(2 3 4, 3 4 5))

Siêu dữ liệu từ mỗi bảng Feature được tập hợp lại và lưu trữ trong bảng geometry_columns để tổ chức kiểu và hệ thống tọa độ của dạng geometry được chứa trong đó.

Để đảm bảo thống nhất về nền tảng công nghệ, đặc biệt quan trọng là hệ quản trị cơ sở dữ liệu lớn. Dự án thành phần “Xây dựng giải pháp kỹ thuật và đầu tư công nghệ” sẽ tiến hành mua sắm bằng hình thức đấu thầu theo quy định một giải pháp công nghệ cho hệ quản trị cơ sở dữ liệu.

Hệ quản trị cơ sở dữ liệu lớn phải đảm bảo các điều kiện sau :

- Là sản phẩm của một trong các hãng sản xuất hệ quản trị cơ sở dữ liệu hàng đầu trên thế giới bao gồm : Oracle, IBM, Informix và Microsoft
- Sản phẩm hỗ trợ vận hành trên nhóm các máy chủ, xử lý song song đối với các máy chủ nhiều CPU, cân bằng tải và tự động chuyển máy chủ khi có sự cố
- Hỗ trợ trực tuyến từ chính hãng sản xuất trong thời gian ít nhất 01 năm
- Các yêu cầu kỹ thuật cụ thể khác sẽ được thuyết minh trong báo cáo kinh tế kỹ thuật

Từ những phân tích các hệ quản trị cơ sở dữ liệu và những giải pháp kỹ thuật áp dụng xây dựng cơ sở dữ liệu quốc gia thì Hệ quản trị PostGres được đánh giá phù hợp với mô hình kiến trúc và điều kiện đề tài đặt ra.

3.3.2 Công nghệ bản đồ

GeoServer 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. Được kỳ vọng sẽ trở thành một phương thức đơn giản để kết nối những nguồn thông tin có sẵn từ Google Earth, NASA World Wind nhằm tạo ra các dịch vụ Webmap như Google Maps, Windows Live Local và Yahoo Maps.

GeoServer được viết bằng ngôn ngữ Java, cho phép người sử dụng chia sẻ và chỉnh sửa dữ liệu không gian địa lý (geospatial data).

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 là thành phần nền tảng của Geospatial Web.

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.

Các đặc trưng của Geoserver:

GeoServer cho phép xuất dữ liệu linh hoạt dựa vào việc hỗ trợ các chuẩn KML, GML, Shapefile, GeoRSS, Portable Document Format, GeoJSON, JPEG, GIF, SVG, PNG...

GeoServer có thể đọc được nhiều định dạng dữ liệu, bao gồm PostGIS, Oracle Spatial, ArcSDE, DB2, MySQL, Shapefiles, GeoTIFF, GTOPO30 và nhiều loại khác. Bên cạnh đó, GeoServer còn có thể chỉnh sửa dữ liệu nhờ những thành phần xử lý của Chuẩn Web Feature Server.

GeoServer được xây dựng trong bộ GeoTools, được viết bởi ngôn ngữ Java.

MapServer cũng được coi là một sản phẩm mã nguồn mở, có nhiều chức năng tương tự. Sự khác biệt giữa hai sản phẩm là Mapserver được phát triển trong môi trường cũ, được viết bởi ngôn ngữ C và hoạt động nhờ CGI, còn GeoServer được viết bằng ngôn ngữ Java. GeoServer phân biệt với MapServer bởi nó có giao diện đồ họa, giúp đơn giản hơn trong việc cấu hình, và thực thi chức năng sửa đổi dựa vào Web Feature Server, cho phép chỉnh sửa thông tin không gian cả trên Web cũng như trên máy trạm Desktop. Ưu điểm lớn nhất của MapServer là tốc độ thực thi nhanh hơn GeoServer, nhưng từ phiên bản 1.6 trở đi của GeoServer thì tốc độ của hai sản phẩm đã tương đương.

GeoServer hỗ trợ việc chia sẻ dữ liệu không gian địa lý lên Google Earth thông qua đặc tính 'network link' sử dụng KML.

3.3.3 Công cụ phát triển phần mềm

Xây dựng công cụ trên nền tảng .Net

ASP.NET là một nền tảng ứng dụng web (web application framework) mới nhất được phát triển và cung cấp bởi Microsoft tên mở rộng là .aspx, cho phép những người lập trình tạo ra những trang web động, những ứng dụng web và những dịch vụ web. Lần đầu tiên được đưa ra thị trường vào tháng 2 năm 2002 cùng với phiên bản 1.0 của .NET framework, là công nghệ nối tiếp của Microsoft Active Server Pages(ASP). ASP.NET được biên dịch dưới dạng Common Language Runtime (CLR), cho phép những người lập trình viết mã ASP.NET với bất kỳ ngôn ngữ nào được hỗ trợ bởi .NET language.

Trang ASP.NET được biên dịch trước thay vì phải đọc và biên dịch mỗi khi trang web nhận được yêu cầu ,Khác với trang sử dụng ngôn ngữ khác mỗi lần triệu gọi là mỗi lần trang web phải biên dịch lại tốn rất nhiều tài nguyên cho việc xử lý như thế,vấn đề này làm chậm tiến trình xử lý của hệ thống

- ASP.NET biên dịch những trang web động thành những tập tin DLL có thể thi hành nhanh chóng và hiệu quả

- ASP.NET hỗ trợ mạnh mẽ bộ thư viện phong phú và đa dạng của .NET Framework làm việc với XML, Web Service, truy cập CSDL qua ADO.Net

- ASP.NET sử dụng phong cách lập trình mới: Code behide. Tách code riêng, giao diện riêng nên dễ dàng cho việc nâng cấp và bảo trì

- ASP.NET giúp tối ưu hệ thống ,giải quyết hiện tượng thất cổ chai (nghẽn) khi có nhiều truy cập cùng lúc

- ASP.NET là ngôn ngữ lập trình hướng đối tượng (OOP-object-oriented programming). OOP được xem là giúp tăng năng suất, đơn giản hóa độ phức tạp khi bảo trì cũng như mở rộng hệ thống bằng cách cho phép lập trình viên tập trung vào các đối tượng phần mềm ở bậc cao hơn

- ASP.NET có đặc tính kế thừa cao (inheritance): Đặc tính này cho phép một đối tượng có thể có sẵn các đặc tính mà đối tượng khác đã có thông qua kế thừa. Điều này cho phép các đối tượng chia sẻ hay mở rộng các đặc tính sẵn có mà không phải tiến hành định nghĩa lại

- ASP.NET phù hợp với các ứng dụng web lớn, các CSDL lớn đòi hỏi, chiếm đến 55% thị phần về các giải pháp ứng dụng Web

- ASP.NET Khả năng mở rộng cao hơn.: có ý nghĩa là một ứng dụng có thể trải rộng tương tác trên nhiều server, khả năng giao tiếp giữa các server được tăng cường.

- ASP.NET cho phép sử dụng các điều khiển Login mới nhằm giúp tạo ra các trang bảo mật nhanh hơn bằng cách kéo và thả điều khiển Login vào trang .aspx trong môi trường IDE như Visual Studio .NET hoặc WebMatrix.

- ASP.NET hỗ trợ xác thực người dùng dựa vào form bao gồm quản lý cookie và tự động chuyển trang đối với những người dùng không hợp lệ.

Xây dựng công cụ trên nền tảng Java

Máy ảo Java (JVM - Java Virtual Machine)

Tất cả các chương trình muốn thực thi được thì phải được biên dịch ra mã máy. Mã máy của từng kiến trúc CPU của mỗi máy tính là khác nhau (tập lệnh mã máy của CPU Intel, CPU Solarix, CPU Macintosh ... là khác nhau), vì vậy trước đây một chương trình sau khi được biên dịch xong chỉ có thể chạy được trên một kiến trúc CPU cụ thể nào đó. Đối với CPU Intel chúng ta có thể chạy các hệ điều hành như Microsoft Windows, Unix, Linux, OS/2, ...

Chương trình thực thi được trên Windows được biên dịch dưới dạng file có đuôi .EXE còn trên Linux thì được biên dịch dưới dạng file có đuôi .ELF, vì vậy trước đây một chương trình chạy được trên Windows muốn chạy được trên hệ điều hành khác như Linux chẳng hạn thì phải chỉnh sửa và biên dịch lại.

Ngôn ngữ lập trình Java ra đời, nhờ vào máy ảo Java mà khó khăn nêu trên đã được khắc phục. Một chương trình viết bằng ngôn ngữ lập trình Java sẽ được biên dịch ra mã của máy ảo java (mã java bytecode). Sau đó máy ảo Java chịu trách nhiệm chuyển mã java bytecode thành mã máy tương ứng. Sun Microsystem chịu trách nhiệm phát triển các máy ảo Java chạy trên các hệ điều hành trên các kiến trúc CPU khác nhau.

Thông dịch:

Java là một ngôn ngữ lập trình vừa biên dịch vừa thông dịch. Chương trình nguồn viết bằng ngôn ngữ lập trình Java có đuôi *.java đầu tiên được biên dịch thành tập tin có đuôi *.class và sau đó sẽ được trình thông dịch thông dịch thành mã máy.

Độc lập nền:

Một chương trình viết bằng ngôn ngữ Java có thể chạy trên nhiều máy tính có hệ điều hành khác nhau (Windows, Unix, Linux, ...) với điều kiện ở đó có cài đặt máy ảo java (Java Virtual Machine).

Hướng đối tượng:

Hướng đối tượng trong Java tương tự như C++ nhưng Java là một ngôn ngữ lập trình hướng đối tượng hoàn toàn. Tất cả mọi thứ đề cập đến trong Java đều liên quan đến các đối tượng được định nghĩa trước, thậm chí hàm chính của một chương trình viết bằng Java (đó là hàm main) cũng phải đặt bên trong một lớp. Hướng đối tượng trong Java không có tính đa kế thừa (multi inheritance) như trong C++ mà thay vào đó Java đưa ra khái niệm interface để hỗ trợ tính đa kế thừa.

Đa nhiệm - đa luồng (MultiTasking - Multithreading):

Java hỗ trợ lập trình đa nhiệm, đa luồng cho phép nhiều tiến trình, tiểu trình có thể chạy song song cùng một thời điểm và tương tác với nhau.

Khả chuyển (portable):

Chương trình ứng dụng viết bằng ngôn ngữ Java chỉ cần chạy được trên máy ảo Java là có thể chạy được trên bất kỳ máy tính, hệ điều hành nào có máy ảo Java. “Viết một lần, chạy mọi nơi” (Write Once, Run Anywhere).

Hỗ trợ mạnh cho việc phát triển ứng dụng:

Công nghệ Java phát triển mạnh mẽ nhờ vào “đại gia Sun Microsystem” cung cấp nhiều công cụ, thư viện lập trình phong phú hỗ trợ cho việc phát triển nhiều loại hình ứng dụng khác nhau cụ thể như:

- J2SE (Java 2 Standard Edition) hỗ trợ phát triển những ứng dụng đơn, ứng dụng client-server.
- J2EE (Java 2 Enterprise Edition) hỗ trợ phát triển các ứng dụng thương mại.
- J2ME (Java 2 Micro Edition) hỗ trợ phát triển các ứng dụng trên các thiết bị di động, không dây, ...

JavaScript, JQuery và thư viện mã nguồn mở OpenLayers

JavaScript, theo phiên bản hiện hành, là một ngôn ngữ lập trình kịch bản dựa trên đối tượng được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ này được dùng rộng rãi cho các trang web, nhưng cũng được dùng để tạo khả năng viết script sử dụng các đối tượng nằm sẵn trong các ứng dụng. Nó vốn được phát triển bởi Brendan Eich tại Hãng truyền thông Netscape với cái tên đầu tiên Mocha, rồi sau đó đổi tên thành LiveScript, và cuối cùng thành JavaScript. Giống Java, JavaScript có cú pháp tương tự C, nhưng nó gần với Self hơn Java. .js là phần mở rộng thường được dùng cho tập tin mã nguồn JavaScript.

Phiên bản mới nhất của JavaScript là phiên bản 1.5, tương ứng với ECMA-262 bản 3. ECMAScript là phiên bản chuẩn hóa của JavaScript. Trình duyệt Mozilla phiên bản 1.8 beta 1 có hỗ trợ không đầy đủ cho E4X - phần mở rộng cho JavaScript hỗ trợ làm việc với XML, được chuẩn hóa trong ECMA-357.

JQuery thư viện JavaScript đa trình duyệt được thiết kế để đơn giản hóa lập trình phía máy người dùng của HTML, phát hành vào tháng 1 năm 2006 tại BarCamp NYC bởi John Resig. Được sử dụng bởi hơn 52% trong 10.000 truy cập nhiều nhất các trang web, jQuery là thư viện JavaScript phổ biến nhất được sử dụng ngày nay

jQuery miễn phí, mã nguồn mở phần mềm, kép cấp phép theo MIT Giấy phép GNU General Public License, phiên bản 2 jQuery của được để làm cho nó dễ dàng hơn để di chuyển một tài liệu, chọn DOM các yếu tố, tạo ra hoạt hình s, xử lý Sự kiện, và phát triển Ajax ứng dụng. jQuery cũng cung cấp khả năng cho các nhà phát triển để tạo ra plug-in s trên đầu trang của thư viện JavaScript. Điều

này cho phép các nhà phát triển để tạo ra trừu tượng hóa ở mức độ thấp tương tác và hình ảnh động, hiệu ứng tiên tiến và vật dụng cao cấp, chủ đề có thể. Cách tiếp cận mô-đun để thư viện jQuery cho phép tạo ra các công cụ mạnh mẽ và năng động web và các ứng dụng web.

OpenLayers là một dự án của tổ chức OGC. OpenLayers là một thư viện điện tử mã nguồn mở JavaScript 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 phát triển và hỗ trợ bởi rất nhiều tổ chức trên thế giới. OpenLayers cho phép hiển thị bản đồ và các điểm đánh dấu (marker) lên 1 trang web từ nhiều nguồn khác nhau.

Hỗ trợ việc hiển thị dữ liệu bản đồ trên mọi trình duyệt hiện nay. Hỗ trợ việc xây dựng các ứng dụng địa lý trên web. Hỗ trợ các phương pháp chuẩn cho việc truy cập dữ liệu địa lý, ví dụ như 2 giao thức WMS và WFS của OGC

Viết hoàn toàn bằng JavaScript. OpenLayers được viết bằng JavaScript theo hướng đối tượng, sử dụng các thành phần từ Prototype.js và thư viện Rico. OpenLayers tách rời phần công cụ bản đồ và dữ liệu bản đồ. Nhờ đó mọi công cụ đều có thể hoạt động trên các nguồn dữ liệu khác nhau.

OpenLayers cho phép người dùng hiển thị nhiều layer khác nhau từ nhiều nguồn khác nhau cùng 1 lúc. Các nguồn này có thể là 1 WMS, WFS hay những dịch vụ bản đồ web mở khác như GeoRSS, OpenStreetMap, Google Maps/Earth hay các file dữ liệu như GML, KML....

OpenLayers là hoàn toàn miễn phí, mã nguồn mở JavaScript. Không phải trả phí và phức tạp như Google Maps API (Nếu sử dụng Google Maps API thì máy bạn phải kết nối Internet nếu không thì bạn phải trả phí để sử dụng).

Công cụ phát triển Oracle Application Development Framework (ADF)

a. Giới thiệu về Oracle ADF:

Oracle Application Development Framework (Oracle ADF) là một ‘end-to-end’ framework được xây dựng trên các công nghệ nền tảng java, Java EE cơ bản và mã nguồn mở để có thể dễ dàng và nhanh chóng thực thi các ứng dụng hướng dịch vụ (service – oriented).

Nếu phát triển những giải pháp cho doanh nghiệp như là tìm kiếm, hiển thị, tạo mới sửa đổi và xác nhận tính hợp lệ của dữ liệu thông qua web, wireless, desktop hoặc webservice interface, Oracle ADF có thể làm công việc này trở nên thật dễ dàng.

Oracle 11g và Oracle ADF cho ta một môi trường để có thể phát triển ứng dụng, với khả năng kéo thả data binding, màn hình thiết kế giao diện người dùng và tích hợp các tiện ích để có thể phát triển theo nhóm.

b. Cấu trúc nền tảng Oracle ADF:

Ứng dụng của bạn được xây dựng sẽ được sử dụng công nghệ cấu trúc Fusion web với các phân tách biệt về các lớp business logic, page navigation và giao diện người dùng trên nền tảng model-view-controller.

- Lớp layer mô tả giá trị của dữ liệu gắn liền với trang hiện tại.
- Lớp view chứa các trang giao diện người dùng để có thể xem và sửa dữ liệu.
- Lớp controller thực thi các lệnh của người dùng và điều hướng trang.
- Lớp business service điều khiển sự truy cập dữ liệu, chứa đựng business logic.

Mô-đun chính của framework là Oracle ADF Model, là nơi định nghĩa các databinding mà thi hành kỹ thuật JRS-227. Kỹ thuật này cung cấp API để truy cập data binding khởi tạo. Oracle ADF Model cho phép kết hợp bất cứ giao diện người dùng nào với bất cứ business service nào, mà không cần phải viết code. Các mô-dun khác tạo nên Fusion web framework như:

- Các thành phần Oracle ADF Business, để xây dựng các business service một cách đơn giản.
- Oracle ADF Faces, mang đến một thư viện phong phú cho phép sử dụng AJAX-enable UI cho ứng dụng web xây dựng trên JavaServer Faces (JSF).
- Oracle ADF controller, kết hợp JSF với Oracle ADF model. ADF Controller mở rộng từ JSF controller cơ bản bằng một số chức năng thêm vào, như là sự sử dụng lại các task flow không chỉ điều hướng giữa các trang JSF mà còn giữa các hoạt động khác, cho các thể hiện của việc gọi phương thức hoặc các task flow khác.

Oracle ADF cho phép các nhà phát triển lựa chọn công nghệ phù hợp để sử dụng trên mỗi lớp. Biểu đồ trên cho thấy các tùy chọn khác nhau có sẵn cho các nhà phát triển khi xây dựng các ứng dụng Oracle ADF.

Sự gắn kết giữa các thành phần khác nhau của ứng dụng Java EE làm cho Oracle ADF Model layer phát triển một cách linh hoạt. EJB, Web Services, JavaBeans, JPA/EclipseLink/TopLink... đều có thể sử dụng như Business Services cho Oracle ADF Model. View layers có thể bao gồm các giao diện Web dựa trên thực hiện với JSF, các ứng dụng Desktop Swing và MS Office front, cũng như các giao diện cho các thiết bị di động.

c. Business Services Layer

Lớp Business Services quản lý tương tác với một lớp dữ liệu. Nó cung cấp các dịch vụ như: đối tượng / bản đồ quan hệ, quản lý giao dịch, và các thực thi logic.

Business Services trong Oracle ADF có thể được thực hiện trong bất kỳ các tùy chọn sau:

Như các lớp của Java, EJB, dịch vụ Web, các đối tượng JPA, và Oracle ADF Business. Ngoài ra, dữ liệu có thể được thực hiện trực tiếp từ các tập tin (XML hoặc CSV) cũng như REST.

Business and Data Services:.. Mô hình dữ liệu back-end tương tác với nguồn dữ liệu có thể là một tập hợp các lớp Java hay WebServices, hoặc có thể là Enterprise JavaBeans (EJB), TopLink, hoặc ADF Business Components..

d. The controller Layer

Các lớp điều khiển quản lý luồng và xử lý các ứng dụng người dùng nhập vào. Ví dụ, khi bạn nhấp vào nút Tìm kiếm trên một trang, bộ điều khiển xác định những hành động để thực hiện (thực hiện tìm kiếm) và ở đâu để điều hướng đến (các trang kết quả). Có hai lựa chọn bộ điều khiển cho các ứng dụng dựa trên web trong JDeveloper: bộ điều khiển JSF tiêu chuẩn hoặc điều khiển ADF mà mở rộng các chức năng điều khiển JSF. Với ADF Controller bạn có thể ngắt các flows thành flows nhỏ hơn, tái sử dụng chúng..

e. The View Layer

The view layer đại diện cho giao diện người dùng của ứng dụng. Oracle ADF hỗ trợ truy cập đa kênh với các dịch vụ business services cho phép bạn sử dụng lại business services và truy cập chúng từ Web client, một client-server swing desktop, bảng tính Microsoft Excel, hoặc một thiết bị di động như điện thoại thông minh. Cho Web dựa trên giao diện Oracle ADF cung cấp một tập hợp phong phú của hơn 150 thành phần Ajax cho phép JSF mà đơn giản hóa việc tạo ra các giao diện người dùng năng động và hấp dẫn.

Business Services: ADF Business (ADF BC) thành phần cung cấp các Business Services đại diện cho bảng cơ sở dữ liệu và sự bền bỉ giá trị đầu vào người sử dụng cơ sở dữ liệu.

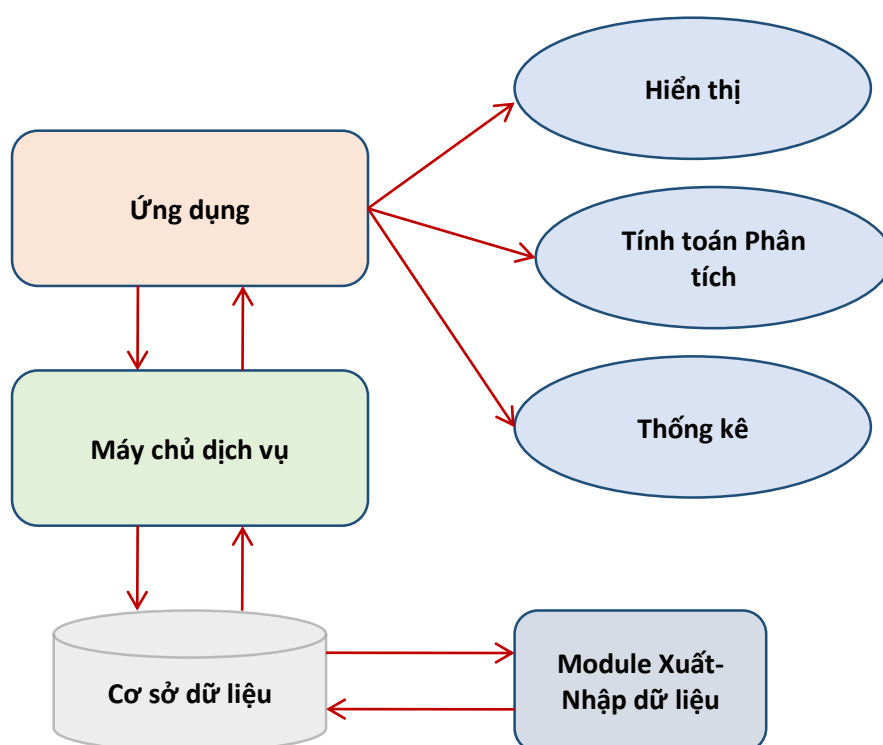
Model: ADF dữ liệu ràng buộc được thực hiện phù hợp với JSR-227, một dữ liệu chuẩn ràng buộc và cơ sở truy cập dữ liệu cho Java EE cung cấp một tiêu chuẩn cho các tương tác giữa các thành phần và phương pháp có sẵn trên các Business Services với dữ liệu tiêu chuẩn, hay Java. công nghệ này có thể liên kết với bất kỳ Business Services nào.

View: Java ServerFaces và ADF Faces, bộ dụng cụ này cho phép các thành phần giao diện người dùng được xác định bởi Java ServerFaces hoặc ADF Faces để được trả lại nội dung khác nhau trên các thiết bị khác nhau Ví dụ, giao diện người dùng có thể được trả lại trên trình duyệt của máy tính hoặc trên một thiết bị di động như một tế bào điện thoại hoặc PDA.

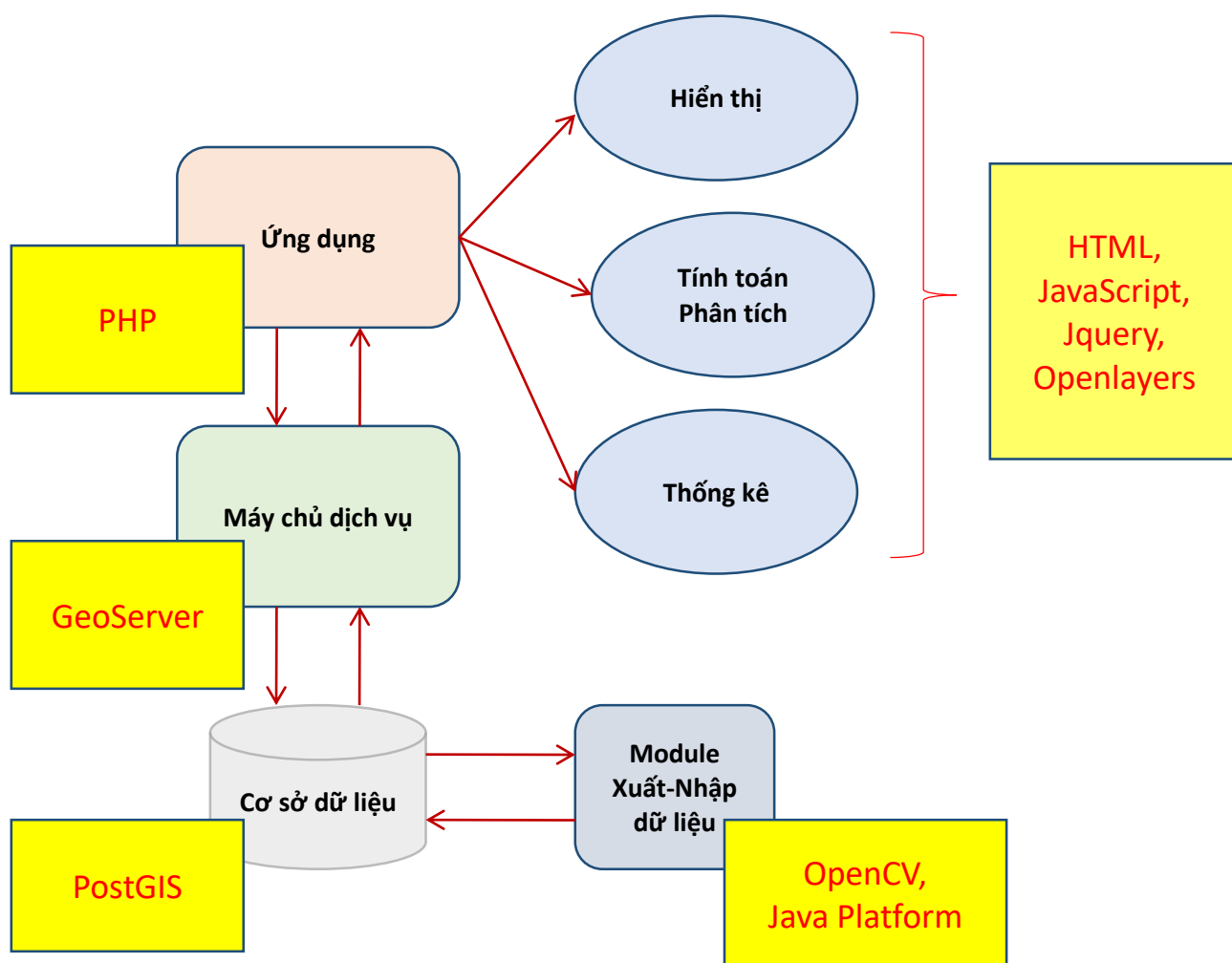
Metadata Services: các ứng dụng ADF BC cũng có thể sử dụng Metadata Services (MDS) cho tùy biến của hệ thống.

Java của Oracle và Công cụ phát triển Web Oracle JDeveloper11 là một công cụ phát triển toàn diện, cung cấp cơ sở để xây dựng các ứng dụng web sử dụng một bộ công cụ phát triển chung. Công cụ này đã phát triển để cung cấp một môi trường tích hợp cho phép một nhà phát triển để thiết kế, phát triển, khắc phục sự cố, tối ưu hóa, và các ứng dụng triển khai Java EE. JDeveloper là môi trường phát triển tích hợp (IDE) mà sử dụng các khung phát triển ứng dụng.

3.3.4 Mô hình chức năng của hệ thống



Hình 3.2 Mô hình chức năng của hệ thống web



Hình 3.3 Mô hình kiến trúc công nghệ của hệ thống

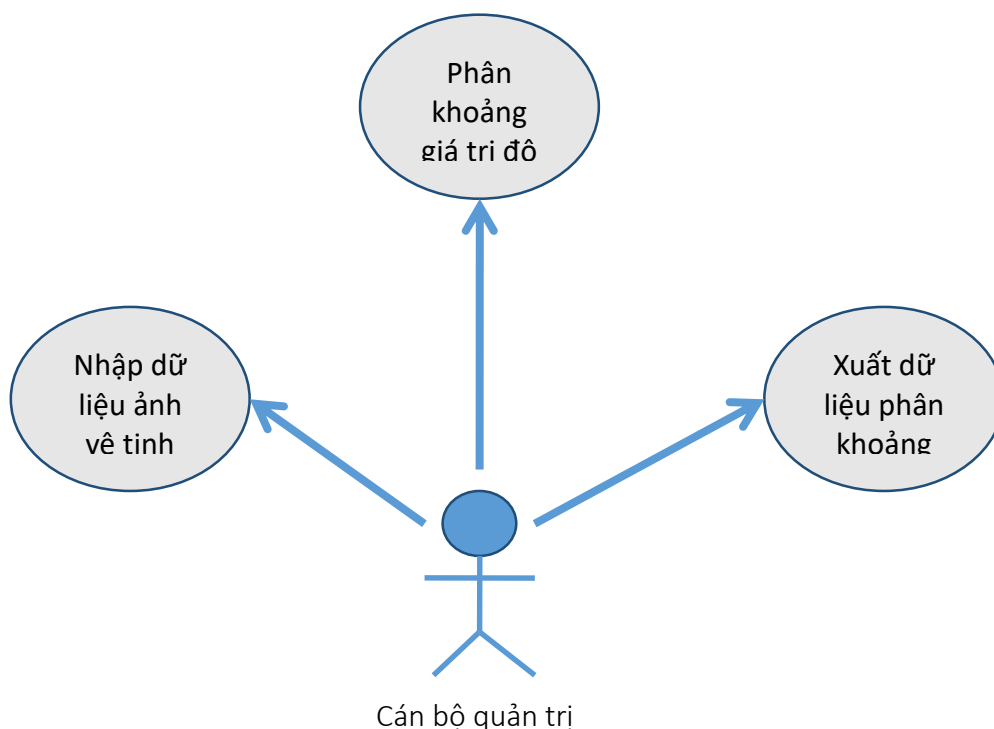
3.3.5 Các chức năng của hệ thống

Chức năng xuất nhập dữ liệu

Đây là phần chức năng chạy trên desktop, chỉ có cán bộ quản trị cấp server mới có thể sử dụng.

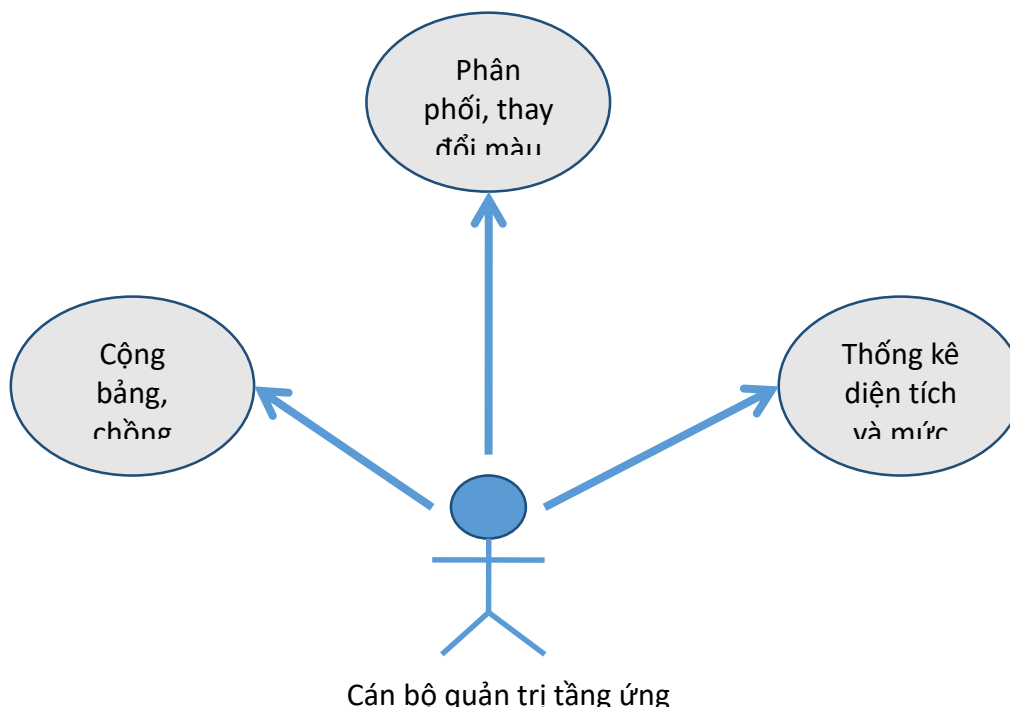
Nhiệm vụ chính là chuyển dữ liệu ảnh vệ tinh thành dạng bảng grid (dạng vector) để chuyển vào lưu trữ trong CSDL. Phân khoảng độ xám và gán giá trị tương ứng.

Cập nhật và quản lý dữ liệu theo thời gian.



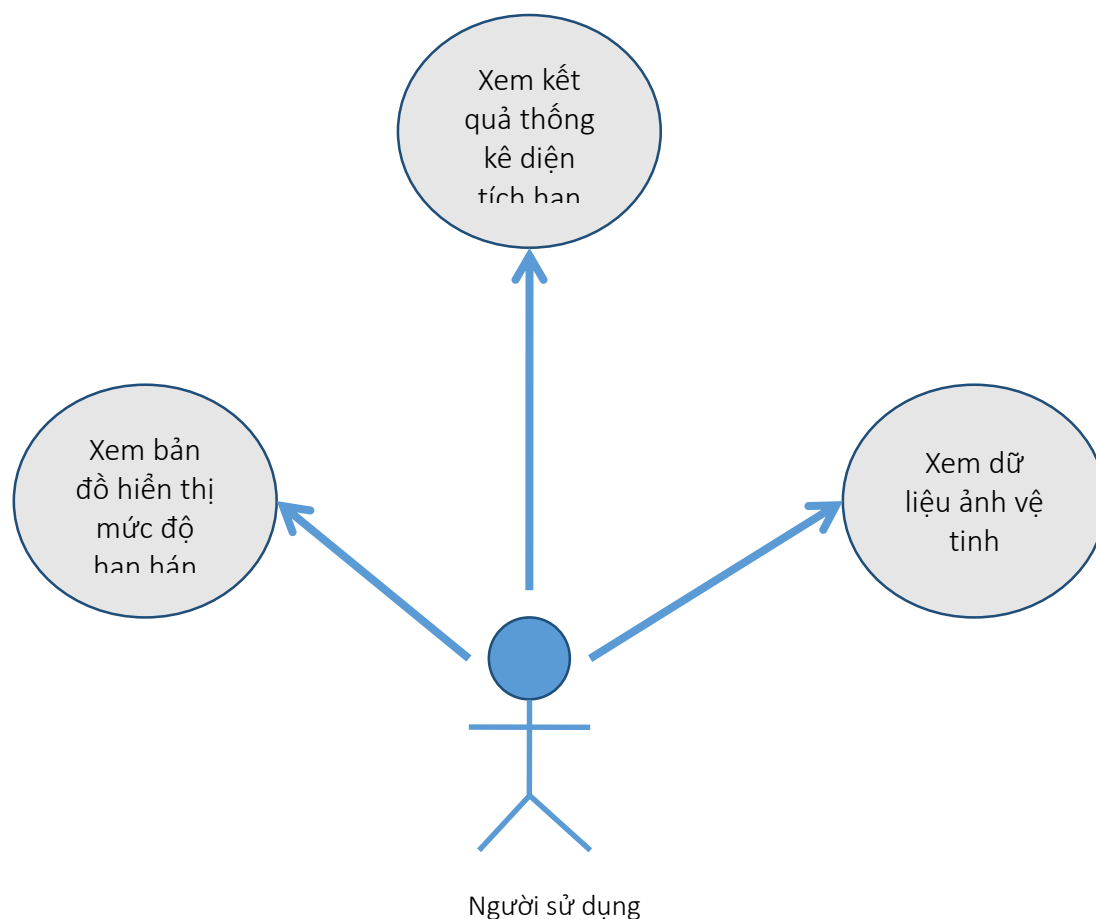
Hình 3.4 Mô hình chức năng xuất nhập dữ liệu

Các chức năng liên quan đến bản đồ



Hình 3.5 Chức năng bản đồ đối với quản trị viên tầng ứng dụng

Tầng ứng dụng có chức năng hiển thị bản đồ ô nhiễm trên nền bản đồ web và thực hiện các thao tác phân tích, phối màu, trình bày bản đồ chuyên đề. Người quản trị tầng ứng dụng có thể đăng nhập vào hệ thống và thực hiện các thao tác đối với dữ liệu và hiển thị dữ liệu theo ý mình.



Hình 3.6 Chức năng bản đồ đối với người dùng

Người dùng bình thường có thể truy cập vào trang web và xem các thông tin về các kết quả nghiên cứu của đề tài cũng như trạng thái thủy lợi của lưu vực sông Cả trên bản đồ nhưng không được phép thao tác với dữ liệu gốc.

CHƯƠNG 4 TÍCH HỢP, KIỂM THỬ, ĐÁNH GIÁ VÀ HIỆU CHỈNH HỆ THỐNG

4.1 Kiểm thử hệ thống

4.1.1 Khái niệm kiểm thử hệ thống

Kiểm thử sản phẩm phần mềm là xây dựng một cách có chủ đích những tập dữ liệu và dãy các thao tác nhằm đánh giá một số hoặc toàn bộ các tiêu chuẩn của phần mềm để đảm bảo rằng phần mềm thỏa mãn các yêu cầu của chúng và các yêu cầu đó thỏa mãn các yêu cầu của khách hàng.

Kiểm thử có hai mục tiêu là kiểm thử hợp lệ và kiểm thử khiếm khuyết.

- + Kiểm thử hợp lệ: để chứng minh cho đội ngũ phát triển và khách hàng thấy phần mềm đã thỏa mãn mọi yêu cầu hay chưa? Kiểm thử hợp lệ thành công khi hệ thống đã vận hành như mong đợi.
- + Kiểm thử khiếm khuyết: nhằm phát hiện các lỗi hoặc những khiếm khuyết của phần mềm và kiểm tra xem chức năng của nó có chính xác hoặc phù hợp với tài liệu đặc tả của nó hay không.

Ta đã biết kiểm thử là một quá trình đi tìm lỗi. Một kiểm thử thành công là kiểm thử tìm ra các lỗi mới, một kiểm thử tồi là kiểm thử mà không tìm được lỗi. Có ba kiểu lỗi trong ứng dụng phần mềm và các kiểm thử tốt sẽ phải xác định được cả ba loại lỗi đó, bao gồm:

- + Các chức năng của chương trình chạy bị lỗi hoặc ra kết quả không đúng như mong muốn.
- + Không làm những điều cần phải làm: đây là lỗi bỏ sót thường xuất hiện đối với những ứng dụng mới được phát triển.
- + Làm những điều không cần làm: đây là lỗi của các lệnh đã cư trú trước trong các ứng dụng bảo trì.

Các kiểm thử có các mức khác nhau và được tiến hành bởi các cá nhân khác nhau trong quá trình phát triển ứng dụng. Có kiểm thử được tiến hành bởi đội ngũ

dự án và có kiểm thử được tiến hành bởi các cơ quan bên ngoài để chấp nhận ứng dụng.

4.1.2. Các nguyên tắc kiểm thử

- + Nguyên tắc khách quan: người kiểm thử không phải là tác giả của phần mềm đang được kiểm thử.
- + Nguyên tắc ngẫu nhiên: dữ liệu và chức năng được chọn để kiểm thử tuy có chủ đích nhưng không phải xuất hiện theo một thứ tự nhất định.
- + Nguyên tắc "người sử dụng kém": hệ thống được một người sử dụng có trình độ thấp ở mức chấp nhận được dùng thử, người này có thể gây ra các sự cố không lường trước được của hệ thống.
- + Nguyên tắc "kẻ phá hoại": hệ thống rơi vào tay những người có trình độ nghiệp vụ cao, chủ ý để phá hoại. Trình độ ở đây thuộc lĩnh vực công nghệ thông tin hoặc lĩnh vực mà phần mềm đang xây dựng.

4.1.3. Độ tin cậy của phần mềm

Độ tin cậy của một hệ thống phần mềm là độ đo về mức độ chính xác của các chức năng mà hệ cung cấp. Độ tin cậy của các hệ thống phần mềm khác nhau là khác nhau, ví dụ hệ thống điều khiển máy bay có thể rất hiếm khi mắc lỗi nhưng chỉ cần chúng bị lỗi một lần thì đã bị coi là không đáng tin.

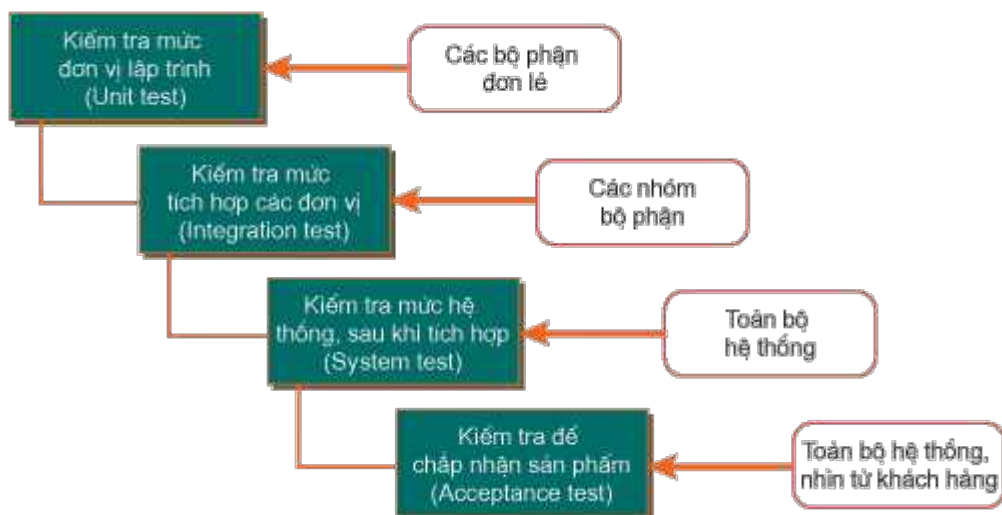
Độ tin cậy của phần mềm thường bị ảnh hưởng bởi một số các lỗi sau:

- + Lỗi chiến lược tức là ý đồ thiết kế sai.
- + Phân tích các yêu cầu không đầy đủ hoặc lệch lạc.
- + Hiểu sai về các chức năng của hệ thống.
- + Kết quả chạy của các chương trình con không đúng như ý đồ lập trình.
- + Lỗi lây lan là lỗi được truyền từ chương trình này sang chương trình khác.
- + Lỗi cú pháp là lỗi viết sai quy định của ngôn ngữ lập trình.
- + Hiệu ứng phụ là lỗi xảy ra khi một đơn vị chương trình làm thay đổi giá trị của một biến ngoài ý muốn của lập trình viên.

Khi phát triển phần mềm cần đảm bảo nguyên lý an toàn là mọi lỗi dù nhỏ hay lớn đều phải được phát hiện ở một bước nào đó của chương trình trước khi nó được thực hiện.

4.1.4. Quá trình kiểm thử

Trừ các hệ thống nhỏ, còn nói chung không nên kiểm thử nguyên cả khối chương trình mà chia thành các giai đoạn như sau:



Hình 4.1: Quy trình kiểm thử

- Kiểm thử đơn vị - Unit Test

Đơn vị - Unit là một thành phần phần mềm nhỏ nhất mà ta có thể kiểm tra được. Theo định nghĩa này các hàm, các thủ tục, các lớp, hoặc các phương thức đều có thể được xem là Unit.

Vì Unit được chọn để kiểm tra thường có kích thước nhỏ và chức năng hoạt động đơn giản nên không có khó khăn gì trong việc tổ chức, kiểm tra, ghi nhận và phân tích kết quả. Nếu phát hiện lỗi thì việc xác định nguyên nhân và khắc phục cũng tương đối dễ dàng vì chỉ khoanh vùng trong một đơn thể Unit đang được kiểm tra. Nguyên lý đúc kết từ thực tiễn cho thấy thời gian tốn cho kiểm thử đơn vị sẽ được đền bù bằng việc tiết kiệm được rất nhiều thời gian và chi phí cho việc kiểm tra, sửa lỗi ở các mức kiểm thử sau đó.

Unit Test thường do lập trình viên thực hiện, công đoạn này cần được thực hiện càng sớm càng tốt trong giai đoạn viết mã và trong suốt chu kỳ phát triển

phần mềm. Thông thường, Unit Test đòi hỏi kiểm thử viên phải có kiến thức về cả thiết kế và mã lệnh của chương trình.

Mục đích của Unit Test là bảo đảm thông tin vào và ra của Unit là chính xác trong mối quan hệ giữa các Unit khác, điều này đòi hỏi tất cả các nhánh bên trong của Unit đều phải được kiểm tra để phát hiện nhánh phát sinh lỗi.

Cũng như các mức kiểm thử khác, Unit Test cũng đòi hỏi phải chuẩn bị trước các tình huống (test case) và kịch bản (script), trong đó chỉ định rõ dữ liệu vào, các bước thực hiện và dữ liệu mong chờ sẽ xuất ra. Các test case và script này nên được giữ lại để tái sử dụng.

- Kiểm thử tích hợp – Integration Test

Kiểm thử tích hợp kết hợp các thành phần của một ứng dụng và kiểm tra như một ứng dụng đã hoàn thành. Trong khi kiểm thử đơn vị kiểm tra các thành phần riêng lẻ thì kiểm thử tích hợp kết hợp chúng lại với nhau và kiểm tra sự giao tiếp giữa chúng.

Kiểm thử tích hợp có 2 mục tiêu chính:

- + Phát hiện lỗi giao tiếp xảy ra giữa các Unit.
- + Tích hợp các Unit đơn lẻ thành các hệ thống nhỏ và cuối cùng là hệ thống hoàn chỉnh chuẩn bị cho kiểm thử ở mức hệ thống.

Trừ một số ít ngoại lệ, nói chung kiểm thử tích hợp chỉ nên thực hiện trên những Unit đã được kiểm tra cẩn thận trước đó bằng Unit Test. Một số ý kiến cho rằng Unit một khi đã qua giai đoạn Unit Test với các giao tiếp giả lập thì không cần phải thực hiện kiểm thử tích hợp nữa nhưng thực tế cho thấy việc tích hợp giữa các Unit dẫn đến những tình huống hoàn toàn khác.

Chiến lược thường được sử dụng trong kiểm thử tích hợp là tích hợp dần dần từng Unit vào nhóm các Unit khác đã được tích hợp và kiểm thử tích hợp trước đó. Lúc này, ta chỉ cần kiểm tra giao tiếp của Unit mới thêm vào với hệ thống các Unit đã tích hợp trước đó, điều này làm cho số lượng kiểm thử sẽ giảm đi rất nhiều và sai sót sẽ cũng bị hạn chế..

Có 4 loại kiểm tra trong kiểm thử tích hợp:

- + Kiểm tra cấu trúc: Kiểm tra nhằm bảo đảm rằng các thành phần bên trong của chương trình là chạy đúng, nó chỉ chú trọng đến hoạt động của các thành phần cấu trúc nội tại của chương trình chẳng hạn các lệnh và nhánh lựa chọn (kiểm thử hộp trắng).
- + Kiểm tra chức năng: Kiểm tra chú trọng đến chức năng của chương trình, khảo sát chúng theo yêu cầu kỹ thuật (kiểm thử hộp đen).
- + Kiểm tra hiệu năng: Kiểm tra việc vận hành của hệ thống.
- + Kiểm tra khả năng chịu tải: Kiểm tra các giới hạn của hệ thống.

- Kiểm thử hệ thống – System Test

Mục đích của kiểm thử hệ thống là kiểm tra toàn bộ hệ thống sau khi tích hợp có thỏa mãn các yêu cầu đã đặt ra hay không?

Kiểm thử hệ thống bắt đầu khi tất cả các bộ phận của phần mềm đã được tích hợp thành công. Thông thường loại kiểm thử này tốn rất nhiều công sức và thời gian, trong nhiều trường hợp việc kiểm tra đòi hỏi một số thiết bị phụ trợ, phần mềm hoặc phần cứng đặc thù, đặc biệt là các ứng dụng thời gian thực hoặc hệ thống nhúng.

Ở mức độ kiểm thử hệ thống, người kiểm thử cũng tìm kiếm các lỗi nhưng trọng tâm là đánh giá về hoạt động, thao tác, sự tin cậy và một số các yêu cầu khác liên quan đến chất lượng của toàn hệ thống.

Điểm khác nhau then chốt giữa kiểm thử tích hợp và kiểm thử hệ thống là kiểm thử hệ thống chú trọng các hành vi và lỗi trên toàn hệ thống, còn kiểm thử tích hợp chú trọng sự giao tiếp giữa các đơn thể hoặc đối tượng khi chúng làm việc cùng nhau. Thông thường ta phải thực hiện kiểm thử đơn vị và kiểm thử tích hợp để bảo đảm mọi Unit và sự tương tác giữa chúng hoạt động chính xác trước khi thực hiện kiểm thử hệ thống.

Sau khi hoàn thành kiểm thử tích hợp, một hệ thống phần mềm đã được hình thành cùng với các thành phần đã được kiểm tra đầy đủ. Tại thời điểm này,

lập trình viên hoặc kiểm tra viên bắt đầu kiểm tra phần mềm như một hệ thống hoàn chỉnh. Việc lập kế hoạch cho kiểm thử hệ thống nên bắt đầu từ giai đoạn hình thành và phân tích các yêu cầu.

Kiểm thử hệ thống kiểm tra cả các hành vi chức năng của phần mềm lẫn các yêu cầu về chất lượng như độ tin cậy, tính tiện lợi khi sử dụng, hiệu năng và bảo mật. Mức kiểm tra này đặc biệt thích hợp cho việc phát hiện lỗi giao tiếp với các phần mềm hoặc phần cứng bên ngoài, chẳng hạn các lỗi “tắc nghẽn” hoặc chiếm dụng bộ nhớ.

Đòi hỏi nhiều công sức, thời gian, tính chính xác và tính khách quan nên kiểm thử hệ thống thường được thực hiện bởi một nhóm kiểm tra viên hoàn toàn độc lập với nhóm phát triển dự án.

Kiểm thử hệ thống gồm nhiều loại kiểm tra khác nhau, phổ biến nhất gồm:

- + Kiểm tra chức năng: bảo đảm các hành vi của hệ thống thỏa mãn đúng yêu cầu thiết kế.
- + Kiểm tra khả năng vận hành: bảo đảm tối ưu việc phân bổ tài nguyên hệ thống, ví dụ phân bổ bộ nhớ nhằm đáp ứng các yêu cầu về thời gian xử lý hay thời gian truy vấn...
- + Kiểm tra khả năng chịu tải: bảo đảm hệ thống vận hành đúng dưới áp lực cao, ví dụ trường hợp nhiều người truy xuất cùng một lúc hoặc các tình huống bất thường...
- + Kiểm tra khả năng bảo mật: bảo đảm tính toàn vẹn, tính bảo mật của dữ liệu và của hệ thống.
- + Kiểm tra khả năng phục hồi: bảo đảm hệ thống có khả năng khôi phục trạng thái ổn định trước đó trong tình huống mất tài nguyên hoặc dữ liệu, điều này đặc biệt quan trọng đối với các hệ thống giao dịch như ngân hàng trực tuyến.

Nhìn từ quan điểm người dùng, các kiểm tra trên rất quan trọng nhằm bảo đảm rằng hệ thống đủ khả năng làm việc trong môi trường thực. Tuy nhiên không

nhất thiết phải thực hiện tất cả các loại kiểm tra trên mà tùy yêu cầu và đặc trưng của từng hệ thống, tùy khả năng và thời gian cho phép của dự án, khi lập kế hoạch, trưởng dự án sẽ quyết định áp dụng những loại kiểm tra nào.

Sau giai đoạn kiểm thử hệ thống, phần mềm được đưa cho khách hàng hoặc người dùng cuối kiểm tra để chấp nhận hoặc dùng thử (Alpha/Beta Test).

- Kiểm thử chấp nhận – Acceptance Test

Sau giai đoạn kiểm thử hệ thống là kiểm thử chấp nhận, loại kiểm thử này được khách hàng thực hiện hoặc ủy quyền cho một nhóm thứ ba thực hiện. Mục đích của kiểm thử chấp nhận là để chứng minh phần mềm thỏa mãn tất cả các yêu cầu của khách hàng và khách hàng chấp nhận sản phẩm.

Đối với những sản phẩm dành để bán rộng rãi trên thị trường cho nhiều người sử dụng, thông thường sẽ thông qua hai loại kiểm thử là Alpha và Beta. Với kiểm thử Alpha, người sử dụng tiềm năng kiểm tra phần mềm ngay tại nơi phát triển phần mềm, lập trình viên sẽ ghi nhận các lỗi và lên kế hoạch sửa chữa. Với kiểm thử Beta, phần mềm sẽ được gửi tới cho người sử dụng tiềm năng để kiểm tra ngay trong môi trường thực, lỗi và các phản hồi sẽ được gửi ngược lại cho lập trình viên để sửa chữa.

Thực tế cho thấy, nếu khách hàng không quan tâm và không tham gia vào quá trình phát triển phần mềm thì kết quả kiểm thử chấp nhận sẽ sai lệch rất lớn, mặc dù phần mềm đã trải qua tất cả các mức kiểm thử trước đó. Sự sai lệch này liên quan đến việc hiểu sai yêu cầu cũng như sự mong chờ của khách hàng. Ví dụ một phần mềm đã được nhóm phát triển dự án kiểm tra về chức năng, nhưng khách hàng khi kiểm tra sau cùng vẫn thất vọng vì bố cục màn hình nghèo nàn, thao tác không tự nhiên, không theo tập quán sử dụng của khách hàng v.v...

Gắn liền với giai đoạn kiểm thử chấp nhận là một nhóm những dịch vụ và tài liệu đi kèm như tài liệu hướng dẫn cài đặt, hướng dẫn sử dụng v.v... Tất cả tài liệu đi kèm phải được cập nhật và kiểm tra chặt chẽ.

- Kiểm thử hồi quy – Regression Test

Trước tiên cần khẳng định kiểm thử hồi quy không phải là một mức kiểm thử như các mức đã nói ở trên. Nó chỉ đơn thuần là kiểm tra lại phần mềm sau khi có một sự thay đổi nào đó xảy ra để bảo đảm sự thay đổi không gây ra lỗi mới trên những chức năng vốn đã làm việc tốt. Kiểm thử hồi quy có thể thực hiện tại mọi mức kiểm thử.

Ví dụ: một phần mềm đang phát triển khi kiểm thử cho thấy nó chạy tốt các chức năng A, B và C. Khi có thay đổi mã lệnh của chức năng C, nếu chỉ kiểm tra chức năng C thì chưa đủ, cần phải kiểm tra lại tất cả các chức năng khác liên quan đến chức năng C ví dụ A và B, lý do là khi C thay đổi, nó có thể sẽ làm A và B không còn làm việc đúng nữa.

Mặc dù không là một mức kiểm thử nhưng thực tế lại cho thấy kiểm thử hồi quy là một trong những loại kiểm thử tốn nhiều thời gian và công sức nhất. Tuy thế, việc bỏ qua kiểm thử hồi quy là không được phép vì có thể dẫn đến tình trạng phát sinh hoặc tái xuất hiện những lỗi nghiêm trọng, vì ta tưởng những lỗi đó hoặc không có hoặc đã được kiểm tra và sửa chữa rồi.

4.1.5. Chiến lược kiểm thử phần mềm

Ta có hai kiểu chiến lược kiểm thử phần mềm là kiểm thử logic và kiểm thử tích hợp:

- + Kiểm thử logic: gồm kiểm thử hộp trắng và kiểm thử hộp đen.
- + Kiểm thử tích hợp: xây dựng hệ thống từ những thành phần của nó và kiểm thử sự tương tác giữa các thành phần, gồm kiểm thử trên-xuống và kiểm thử dưới-lên.

Các chiến lược kiểm thử không loại trừ lẫn nhau, chúng có thể được sử dụng đơn lẻ hoặc đồng thời, tốt nhất là nên sử dụng nhiều kiểu chiến lược để phát hiện được hết các lỗi.

- Kiểm thử hộp đen – Black box

Còn gọi là kiểm thử chức năng, kiểm thử này chỉ quan tâm đến chức năng đã đề ra của chương trình mà không quan tâm đến các khâu thiết kế hoặc viết mã lệnh, do đó khi kiểm thử nó chỉ dựa vào bản mô tả chức năng của chương trình.

Kiểm thử hộp đen nhằm phát hiện các lỗi sau:

- + Thiếu chức năng.
- + Lỗi giao tiếp.
- + Lỗi trong cấu trúc dữ liệu.
- + Lỗi khi thực hiện chương trình.

Các kỹ thuật được áp dụng khi kiểm thử hộp đen gồm:

- + *Phân đoạn tương đương*: mục đích của phương pháp này là giảm số lượng kiểm thử bằng cách chọn các tập dữ liệu đại diện. Ta chia dữ liệu vào thành các đoạn, mỗi đoạn đại diện cho một số dữ liệu và việc kiểm thử chỉ thực hiện trên đại diện đó.
- + *Phân tích giá trị biên*: là trường hợp đặc biệt của phân đoạn tương đương, nó sử dụng các giá trị biên làm giá trị đại diện. Ví dụ nếu miền dữ liệu là tháng thì giá trị < 1 hoặc > 12 là không hợp lệ.
- + *Đoán lỗi*: dựa trên cơ sở trực giác và kinh nghiệm, các chuyên gia có thể dễ dàng kiểm tra các lỗi bằng cách đoán cái nào dễ xảy ra nhất. Ví dụ lỗi chia cho 0, nếu modul có phép chia thì phải kiểm thử lỗi này. Vì dựa trên trực giác, nên phép thử này khó tìm được hết các lỗi.

- Kiểm thử hộp trắng – White box

Là phương pháp kiểm thử dựa vào cấu trúc và mã lệnh chương trình. Phương pháp này kiểm nghiệm một phần của chương trình hay một phần của hệ thống để đảm bảo rằng chúng đáp ứng tốt tất cả các giá trị đầu vào bao gồm cả giá trị không đúng hay không theo dự định của chương trình.

Khi kiểm thử hộp trắng phải đạt được các yêu cầu sau:

- + *Bao phủ dòng lệnh*: mỗi dòng lệnh phải được thực thi ít nhất 1 lần, bởi vì nếu một câu lệnh không được thực hiện thì ta không thể biết có lỗi xảy ra trong câu lệnh đó hay không?
- + *Bao phủ nhánh*: mỗi nhánh của các quyết định logic đều phải được đi qua ít nhất 1 lần.
- + *Bao phủ lặp*: tiến hành trên tất cả các vòng lặp từ giá trị khởi tạo đến giá trị cuối cùng.

- Kiểm thử trên xuống – Top down

Kiểm thử trên xuống: xây dựng khung của hệ thống và đưa các thành phần vào trong nó. Chiến lược này kiểm tra mức cao của hệ thống trước khi kiểm tra các thành phần chi tiết, sau khi thành phần ở mức cao nhất được kiểm tra thì các thành phần con của nó mới được tiến hành kiểm tra và cứ tiếp tục như vậy cho đến khi thành phần ở mức đơn vị được kiểm tra.

- Ưu điểm: Các thành phần hệ thống được thử nghiệm ngay sau khi mã hóa vì vậy các lỗi có thể sớm được phát hiện do đó tiết kiệm được thời gian và chi phí. Ngoài ra ta có thể coi hệ thống tuy chưa hoàn chỉnh nhưng đã có những chức năng chính từ đó có thể minh họa tính khả thi của hệ thống.

- Nhược điểm: Cách thức xây dựng các nhánh và chọn chức năng nào để phát triển trước vẫn là bài toán chưa có nghiệm tối ưu, hơn nữa các mô hình ở mức cao vẫn chưa có đầu ra rõ ràng.

- Kiểm thử dưới lên – Bottom up

Bắt đầu kiểm thử các đơn thể cơ sở không gọi đến các đơn thể khác, sau đó bổ sung kiểm thử các đơn thể gọi đến các đơn thể đã được kiểm thử này. Nguyên tắc của kiểm thử dưới lên là kiểm thử mọi thay đổi tại đơn thể có thể ảnh hưởng tới chức năng của nó. Trong kiểm thử dưới lên tất cả các đơn thể được mã hóa và kiểm tra riêng rẽ.

Mỗi đơn thể phải được kiểm thử với ít nhất hai trường hợp: một trường hợp chạy tốt và một trường hợp không chạy. Trong trường hợp chạy sai hệ thống phải đưa ra được thông báo và quay lại được trạng thái ban đầu của giao dịch.

Nhược điểm của phương pháp là lỗi cấu trúc được phát hiện ở các bước cuối cùng, do đó chi phí xây dựng phần mềm sẽ bị tăng.

4.2 Tích hợp và kiểm thử hệ thống webgi sông Cả

4.2.1. Tích hợp các hợp phần

WebGIS sông Cả được xây dựng theo cấu trúc chuẩn bao gồm 3 tầng: tầng trình bày (front-end), tầng dữ liệu (database), và tầng giao dịch

- Tầng trình bày

Thông thường chỉ là các trình duyệt Internet Explorer, Mozilla Firefox, Chrome ... để mở các trang web theo URL đã được định sẵn là <http://www.websong.tk>. Ứng dụng client ở đây là một website được viết theo công nghệ chuẩn w3c. Các Client đôi khi cũng là một ứng dụng desktop tương tự như phần mềm MapInfo, ArcMap...

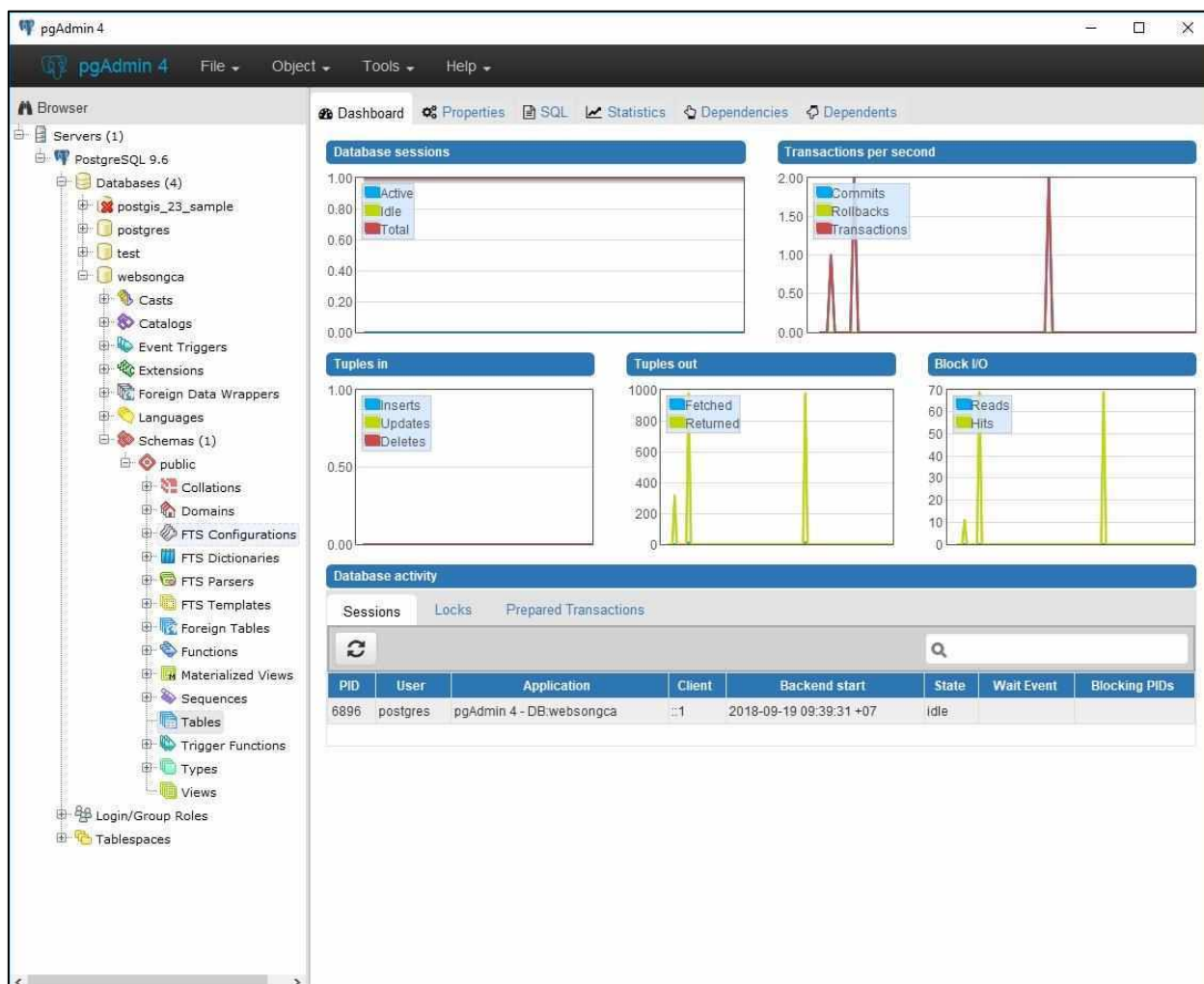
- Tầng giao dịch

Thông thường, tầng giao dịch được tích hợp trong một webserver nào đó, ví dụ như Tomcat, Apache, Internet Information Server. Đó là một ứng dụng phía server nhiệm vụ chính của nó thường là tiếp nhận các yêu cầu từ client, lấy dữ liệu từ cơ sở dữ liệu theo yêu cầu client, trình bày dữ liệu theo cấu hình định sẵn hoặc theo yêu cầu của client và trả kết quả về theo yêu cầu. Tùy theo yêu cầu của client mà kết quả về khác nhau: có thể là một hình ảnh dạng bitmap (jpeg, gif, png) hay dạng vector được mã hóa như SVG, KML, GML,... Một khi dạng vector được trả về thì việc trình bày hình ảnh bản đồ được đảm nhiệm bởi Client, thậm trí client có thể xử lý một số bài toán về không gian. Thông thường các response và request đều theo chuẩn HTTP POST hoặc GET.

- Tầng dữ liệu

Là nơi lưu trữ các dữ liệu địa lý bao gồm cả các dữ liệu không gian và phi không gian.

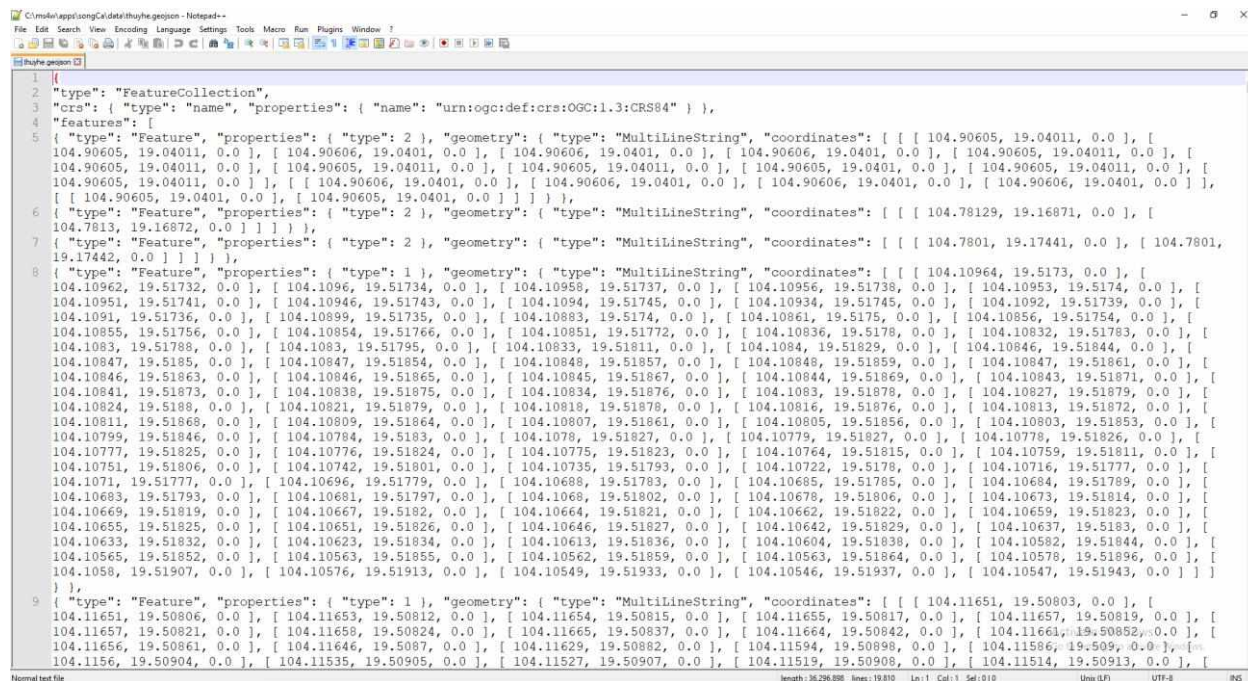
Dữ liệu được lưu trữ và quản lý bằng hệ quản trị cơ sở dữ liệu postgresSQL.



Hình 4.2: hệ quản trị CSDL PostgresSQL

Các dữ liệu địa lý của webGIS sông Cả được lưu ở 2 dạng chủ yếu là vector và raster.

Nghiên cứu đề xuất phương án phối hợp vận hành điều tiết nước hợp lý các hồ chứa thủy lợi, thủy điện trên sông Cả phục vụ cấp nước và phòng, chống lũ cho hạ du



Hình 4.3: Cấu trúc file GeoJSON

Cấu trúc file dữ liệu geojson

Các dữ liệu này được quản trị bởi các hệ quản trị cơ sở dữ liệu như ORACLE, MS SQL SERVER, ESRI SDE, POSTGRESQL,... hoặc là các file dữ liệu dạng flat như shapefile, tab, XML,... Các dữ liệu này được thiết kế, cài đặt và xây dựng theo từng quy trình, từng quy mô bài toán ... mà lựa chọn hệ quản trị cơ sở dữ liệu phù hợp.

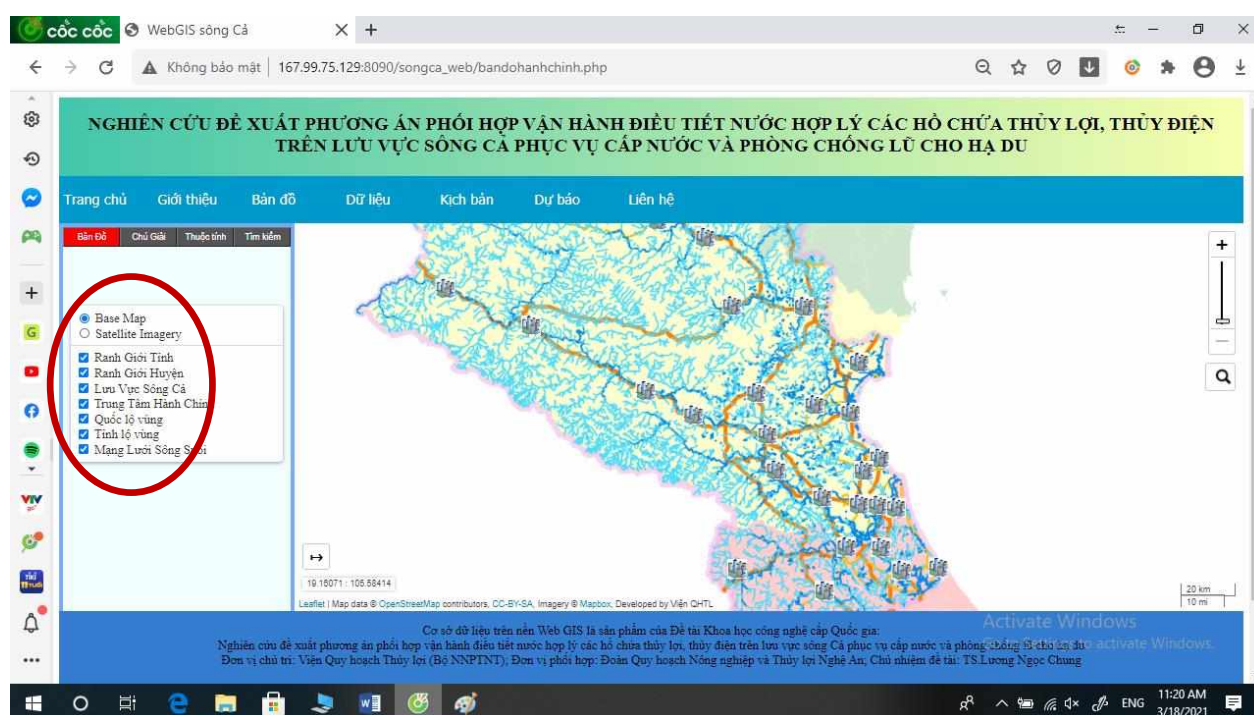
Cơ sở dữ liệu không gian sẽ được dùng để quản lý và truy xuất dữ liệu không gian, được đặt trên data server. Nhà kho hay nơi lưu trữ được dùng để lưu trữ và duy trì những siêu dữ liệu về dữ liệu không gian tại những data server khác nhau. Dựa trên những thành phần quản lý dữ liệu, ứng dụng server và mô hình server được dùng cho ứng dụng hệ thống để tính toán thông tin không gian thông qua các hàm cụ thể. Tất cả kết quả tính toán của ứng dụng server sẽ được gửi đến web server để thêm vào các gói HTML, gửi cho phía client và hiển thị nơi trình duyệt web.

4.2.2. Kiểm thử hệ thống

Như đã định nghĩa ở trên, mục đích của kiểm thử hệ thống là kiểm tra toàn bộ hệ thống sau khi tích hợp có thỏa mãn các yêu cầu đã đặt ra hay không.

- Kiểm tra chức năng

Các chức năng cơ bản của webGIS sông Cả được kiểm tra như phóng to, thu nhỏ, kéo, bật tắt hiển thị các lớp (bao gồm lớp nền địa lý ứng dụng thư viện của các công ty lớn như ESRI, Google, và các lớp bản đồ chuyên đề của đề tài). Các chức năng này hoạt động bình thường.

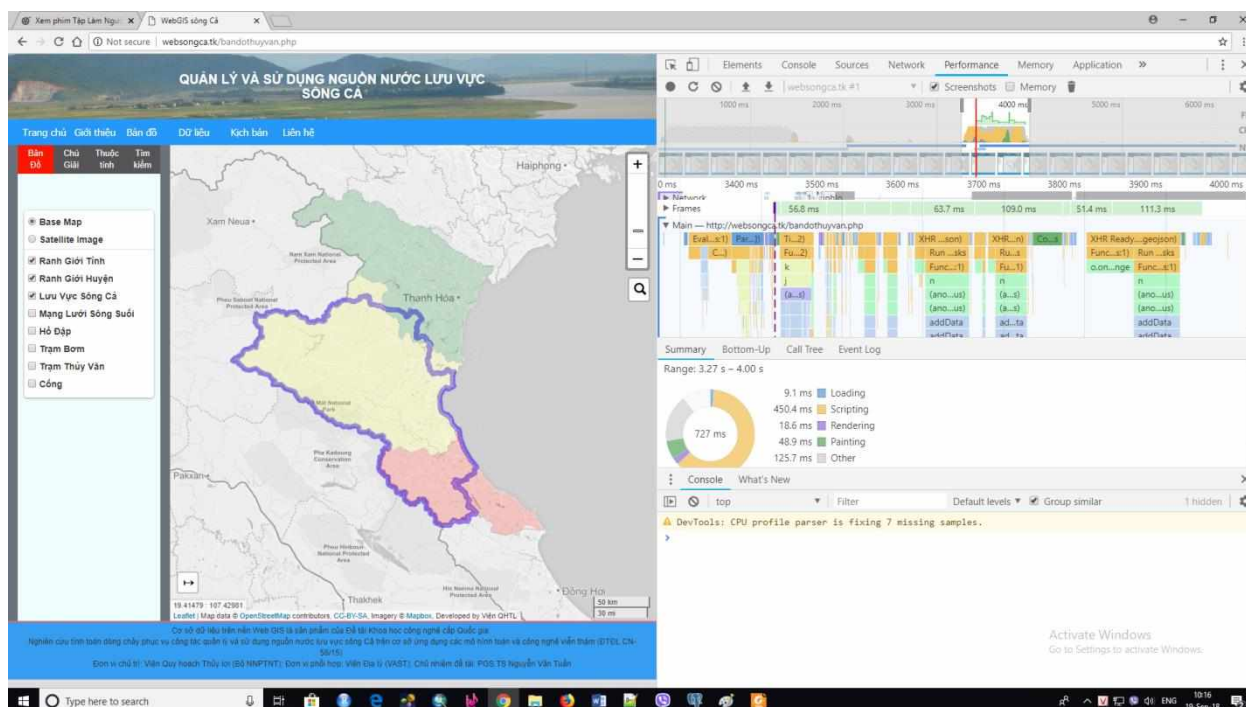


Hình 4.4: Các chức năng cơ bản của webgis sông Cả

- Kiểm tra khả năng vận hành

WebGIS sông Cả được kiểm tra khả năng vận hành để bảo đảm tối ưu việc phân bổ tài nguyên hệ thống (bộ nhớ) nhằm đạt các chỉ tiêu như thời gian xử lý hay đáp ứng câu truy vấn. Kết quả kiểm tra cho thấy tối độ hiển thị đạt được ở mức bình thường

Nghiên cứu đề xuất phương án phối hợp vận hành điều tiết nước hợp lý các hồ chứa thủy lợi, thủy điện trên sông Cả phục vụ cấp nước và phòng, chống lũ cho hạ du



Hình 4.5: kết quả kiểm tra hiệu năng

4.2.3. Kiểm tra khả năng chịu tải

-Mục đích

Nghiên cứu các bộ tiêu chí của các tổ chức tiêu chuẩn quốc tế và đưa ra lựa chọn để xây dựng bộ tiêu chí đánh giá kết quả thử nghiệm cho hệ thống phần mềm đảm bảo được các mục tiêu cụ thể sau:

Cung cấp các kết quả khảo sát, đánh giá của các bộ tiêu chí đánh giá phần mềm theo chuẩn quốc tế

Dựa trên các kết quả đó, tài liệu đưa ra bộ tiêu chí đánh giá đưa ra phải phù hợp với các chức năng và mục tiêu của hệ thống.

Để tiếp cận đối với những người chưa có nhiều kinh nghiệm, thân thiện dễ hiểu với người đọc.

Yêu cầu

- Nghiên cứu các công nghệ nền hiện nay
- Đánh giá ưu nhược điểm của các công nghệ
- Lựa chọn công nghệ áp dụng cho xây dựng phần mềm

- Phương Pháp Nghiên Cứu, Cách Tiếp Cận

Để đáp ứng được mục đích và yêu cầu đặt ra cho chuyên đề này, tác giả sử dụng phương pháp nghiên cứu truyền thống. Cụ thể:

Xác định thông tin, tài liệu tham khảo chính cho chuyên đề: bằng việc tìm kiếm các thông tin, tài liệu liên quan đến nội dung nghiên cứu của chuyên đề, tiến hành phân tích, đánh giá, lựa chọn và được nhóm triển khai đề tài thông qua;

Kế thừa, phát triển các nội dung, kết quả đã nghiên cứu;

Lấy ý kiến chuyên gia: tổ chức họp lấy ý kiến chuyên gia về lĩnh vực, nội dung nghiên cứu của đề tài để từng bước hoàn thiện nội nghiên cứu, đáp ứng mục tiêu và yêu cầu đặt ra của chuyên đề.

Nghiên cứu tiêu chí và quy trình đánh giá sản phẩm phần mềm của các tổ chức tiêu chuẩn quốc tế

-Tổng quan về chất lượng sản phẩm phần mềm

Theo định nghĩa hình thức về chất lượng sản phẩm phần mềm của Tổ chức tiêu chuẩn quốc tế ISO trong bộ tiêu chuẩn 8402, "chất lượng là khả năng đáp ứng toàn diện nhu cầu của người dùng về tính năng cũng như công dụng được nêu ra một cách tường minh hoặc không tường minh trong những ngữ cảnh xác định". Ngay trong định nghĩa này chất lượng cũng được định nghĩa một cách rất "mờ", thiếu yếu tố định lượng. Thêm nữa, để hiểu hết nhu cầu của người sử dụng quả thực là rất khó. Với những khó khăn về định lượng trong khái niệm chất lượng phần mềm, để có được một phần mềm tốt cách thông thường nhất là tiếp cận theo lối chất lượng quy trình. Nghĩa là nếu chúng ta có quy trình sản xuất tốt thì sẽ có khả năng sản xuất ra sản phẩm tốt.

Bộ tiêu chuẩn chất lượng ISO 9001-3 của tổ chức ISO, quy định về "Quy trình đảm bảo chất lượng" trong các tổ chức phát triển phần mềm. Chứng chỉ ISO 9001 xác nhận các tổ chức, đơn vị có quy trình đảm bảo chất lượng hợp chuẩn. Bên cạnh đó, một mô hình khác là CMM (Capability Maturity Model) cũng đang rất được quan tâm tại Việt Nam. Công ty nhận được chứng chỉ CMM nghĩa là công ty đó đã đạt được mức độ tương ứng với các cấp độ CMM của chứng chỉ.

Một doanh nghiệp phát triển phần mềm, nếu có chứng chỉ CMM hoặc ISO 9001 đều có khả năng sản xuất ra các phần mềm tốt hơn hẳn các công ty chưa có chứng chỉ. Tuy nhiên, chúng ta cần lưu ý đây chỉ là "*khả năng*" chứ không phải là "*chắc chắn*". Vẫn có doanh nghiệp có quy trình tốt nhưng sản xuất ra sản phẩm chất lượng không cao. Điều này chứng tỏ cách tiếp cận theo chất lượng quy trình chưa phải là cách tiếp cận toàn diện mà chỉ giải quyết vấn đề ở mức căn bản.

Những năm cuối thế kỷ 20, tổ chức ISO đã tập trung rất nhiều vào các tiêu chuẩn chất lượng cho phần mềm. Cách tiếp cận về chất lượng của ISO đã thực sự tiến thêm một bậc, toàn diện hơn, phù hợp hơn. Kết quả của sự tập trung này là một loạt các bộ tiêu chuẩn đã ra đời, nhằm hướng tới đánh giá chất lượng toàn diện trong suốt vòng đời của sản phẩm phần mềm, từ khi phôi thai cho tới lúc lạc hậu cần thay thế. Theo cách tiếp cận của ISO, chất lượng toàn diện của phần mềm cần phải được quan tâm từ chất lượng quy trình, tới chất lượng phần mềm nội bộ (chất lượng trong), chất lượng phần mềm đối chiếu với yêu cầu của người dùng (chất lượng ngoài) và chất lượng phần mềm trong sử dụng (chất lượng sử dụng).

Ở một góc nhìn khác, vòng đời của một sản phẩm phần mềm bắt đầu từ các bài toán thực tiễn và được thể hiện theo quy trình sau:

1. Từ các bài toán thực tiễn, nhu cầu để phần mềm hình thành;
2. Nhu cầu này được thể hiện qua các tài liệu yêu cầu (Requirements);
3. Nhu cầu sẽ xác định yêu cầu chất lượng ngoài. Thỏa mãn được yêu cầu chất lượng này sẽ thỏa mãn được yêu cầu của người sử dụng;
4. Các yêu cầu chất lượng thể hiện trong tài liệu đặc tả hệ thống (Specification)
5. Yêu cầu chất lượng ngoài là tiền đề cho yêu cầu chất lượng trong;
6. Trong quá trình thiết kế phần mềm, các yêu cầu chất lượng trong được thể hiện trong các tiêu chí của phần mềm và chuyển thành chất lượng trong;
7. Ứng với chất lượng trong có các độ đo chất lượng trong mà phần mềm phải đáp ứng;

8. Tới giai đoạn tích hợp chạy thử, vấn đề được quan tâm sẽ là chất lượng ngoài. Phần mềm được gọi là có chất lượng khi tất cả các độ đo chất lượng ngoài được đảm bảo;

9. Trong quá trình vận hành, vẫn sử dụng các độ đo ngoài, chất lượng của phần mềm trong quá trình vận hành, sử dụng sẽ tiếp tục được xem xét và cải tiến;

10. Quá trình cải tiến sẽ diễn ra liên tục cho tới khi phần mềm trở nên lạc hậu hoàn toàn, cần được thay thế bằng một phần mềm mới.

ISO/IEC 9126

ISO-9126 thiết lập một mô hình chất lượng chuẩn cho các sản phẩm phần mềm.

Bộ tiêu chuẩn này được chia làm bốn phần:

9126-1 Đưa ra mô hình chất lượng sản phẩm phần mềm.

9126-2 Phép đánh giá chất lượng ngoài.

9126-3 Phép đánh giá chất lượng trong.

9126-4 Phép đánh giá chất lượng sản phẩm phần mềm trong quá trình sử dụng.

ISO-9126 là tiêu chuẩn quốc tế đánh giá phần mềm. Được phân chia thành 4 phần tuân theo các tiêu chí một cách nghiêm ngặt: mẫu chất lượng, hệ đo lường bên ngoài và bên trong, hệ đo lường chất lượng khi sử dụng.

Mô hình chất lượng ISO-9126 trên thực tế được mô tả là một phương pháp phân loại và chia nhỏ những thuộc tính chất lượng, nhằm tạo nên những đại lượng đo đếm được dùng để kiểm định chất lượng của sản phẩm phần mềm.

Phạm vi mô hình chất lượng ISO-9126

ISO-9126 mô tả một mô hình chất lượng sản phẩm phần mềm gồm hai phần:

Chất lượng trong và chất lượng ngoài

Chất lượng sử dụng

Phần thứ nhất của mô hình xác định 6 tiêu chí của chất lượng trong, 6 tiêu chí chất lượng ngoài; các tiêu chí này sau đó lại được chia nhỏ thành nhiều tiêu chí con. Những tiêu chí này được bộc lộ ra ngoài khi phần mềm được coi như là

một phần của hệ thống máy tính và là kết quả của các thuộc tính phần mềm bên trong.

Phần thứ hai của mô hình mô tả 4 tiêu chí chất lượng sử dụng. Chất lượng sử dụng là hệ quả của 6 tiêu chí chất lượng sản phẩm phần mềm đối với người dùng.

Các tiêu chí sản phẩm phần mềm này có thể áp dụng cho tất cả các loại phần mềm. Những tiêu chí sản phẩm phần mềm tạo ra sự nhất quán đối với chất lượng sản phẩm phần mềm, đồng thời cung cấp một khung cho việc xác định các yêu cầu đối với chất lượng phần mềm.

Trong phần này, chất lượng sản phẩm phần mềm được xác định và đánh giá theo nhiều hướng, gắn với kết quả thu được, các yêu cầu, sự phát triển, sử dụng, đánh giá, hỗ trợ, tính ổn định, đảm bảo chất lượng và kiểm định của phần mềm. Nó có thể được sử dụng bởi nhà phát triển, tổ chức sử dụng, nhân viên đảm bảo chất lượng phần mềm hay người đánh giá độc lập. Đồng thời nó đặc biệt thích hợp cho việc xác định và đánh giá chất lượng sản phẩm phần mềm. Ví dụ, mô hình chất lượng này có thể được dùng để:

Kiểm tra tính đáp ứng đối với những yêu cầu đã đặt ra.

Xác định các yêu cầu phần mềm.

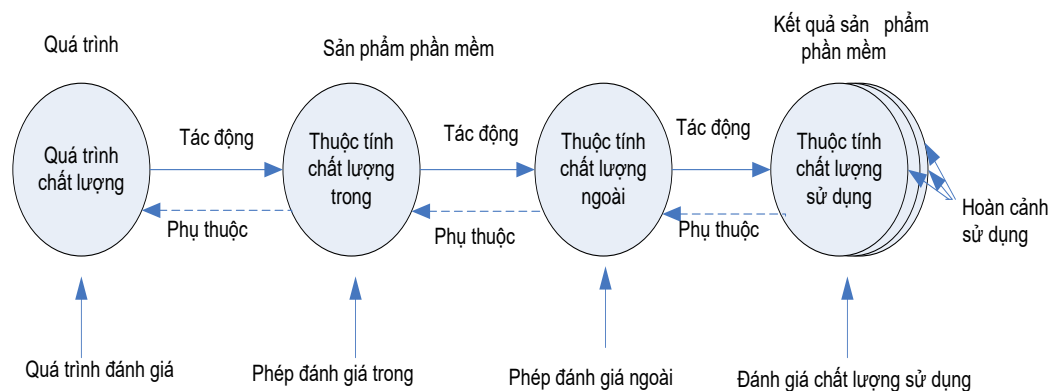
Xác định các đối tượng thiết kế phần mềm.

Xác định các đối tượng kiểm thử phần mềm.

Xác định các tiêu chuẩn đảm bảo chất lượng.

Xác định các tiêu chuẩn chấp nhận cho một sản phẩm phần mềm hoàn chỉnh.

Tiêu chí chất lượng



Hình 4.6: Chất lượng trong vòng đời sản phẩm

Việc đánh giá sản phẩm phần mềm để thỏa mãn các yêu cầu chất lượng là một trong những quy trình trong vòng đời phát triển của phần mềm (Hình 1). Chất lượng sản phẩm phần mềm cần được đánh giá bằng việc đo kiểm các thuộc tính bên trong (thường là các phương pháp đo tĩnh trên các sản phẩm trung gian), hoặc bằng cách đo kiểm các thuộc tính bên ngoài (thường là đo các đáp ứng của mã lệnh khi thực thi), hoặc bằng cách đo kiểm chất lượng các thuộc tính sử dụng. Mục đích là để sản phẩm đáp ứng được những yêu cầu trong từng trường hợp sử dụng cụ thể.

Quy trình chất lượng góp phần cải tiến chất lượng sản phẩm, và chất lượng sản phẩm góp phần cải tiến chất lượng sử dụng. Do đó, việc đánh giá và cải tiến một quy trình đồng nghĩa với cải tiến chất lượng sản phẩm. Tương tự, việc đánh giá chất lượng sử dụng có thể tác động ngược trở lại để cải tiến một sản phẩm và đánh giá một sản phẩm phần mềm có thể tác động trở lại để cải tiến một quy trình.

Các thuộc tính trong thích hợp của phần mềm là yêu cầu tiên đề để đạt được các phản ứng bên ngoài, và các hoạt động bên ngoài thích hợp là yêu cầu tiên đề để đạt được chất lượng sử dụng .

Các yêu cầu cho chất lượng sản phẩm phần mềm sẽ bao gồm các tiêu chí đánh giá chất lượng trong, chất lượng ngoài và chất lượng sử dụng, để đáp ứng yêu cầu của người sử dụng, người bảo dưỡng, tổ chức sử dụng, và người dùng cuối.

Các yêu cầu về chất lượng của người sử dụng được xác định là các yêu cầu chất lượng trong phép đo chất lượng sử dụng, phương pháp đo chất lượng ngoài, thậm chí cả phương pháp đo chất lượng trong. Những yêu cầu này được xác định bằng phương pháp đo và sử dụng như chuẩn khi đánh giá sản phẩm. Để có được một sản phẩm thoả mãn nhu cầu của người dùng đòi hỏi quá trình phát triển phần mềm phải liên tục và luôn luôn có những phản hồi từ phía họ.

Các yêu cầu chất lượng ngoài xác định các mức yêu cầu đối với chất lượng theo hướng nhìn từ bên ngoài. Chúng bao gồm các yêu cầu xuất phát từ nhu cầu người sử dụng, gồm các yêu cầu chất lượng sử dụng. Các yêu cầu chất lượng ngoài được sử dụng như là đích của quá trình kiểm tra tại mỗi giai đoạn phát triển. Các yêu cầu chất lượng ngoài cho tất cả các tiêu chí chất lượng xác định trong phần này nên được đặt trong các đặc tả yêu cầu chất lượng sử dụng phương pháp đo ngoài, nên được chuyển đổi sang các yêu cầu chất lượng trong, và nên được sử dụng như là chuẩn để kiểm tra sản phẩm.

Các yêu cầu chất lượng trong xác định các mức chất lượng yêu cầu theo hướng nhìn từ bên trong của sản phẩm. Các yêu cầu chất lượng trong được sử dụng để xác định tiêu chí của các sản phẩm trung gian. Chúng có thể bao gồm các mô hình tĩnh hoặc động, các tài liệu và mã nguồn khác nhau. Các yêu cầu chất lượng trong có thể được coi là đích cho các kiểm tra tại các giai đoạn khác nhau trong quá trình phát triển. Chúng cũng có thể được sử dụng để xác định các chiến lược phát triển, chuẩn để đánh giá, các kiểm tra trong quá trình phát triển. Có thể sử dụng một số phương pháp đo mở rộng (ví dụ: cho việc tái sử dụng), nằm ngoài phạm vi của ISO/IEC 9126. Các yêu cầu chất lượng trong nên được xác định một cách định lượng qua việc sử dụng phương pháp đo trong.

Chất lượng trong là tổng hợp của tất cả các tiêu chí của sản phẩm phần mềm theo cách nhìn từ bên trong. Chất lượng trong được đo kiểm và đánh giá theo các yêu cầu chất lượng trong. Các chi tiết của chất lượng sản phẩm phần mềm có thể được cải tiến trong suốt quá trình triển khai mã hoá, kiểm thử, nhưng bản chất cơ

bản của chất lượng sản phẩm phần mềm thể hiện qua chất lượng trong thì không thay đổi trừ khi có sự thiết kế lại.

Chất lượng ngoài ước lượng (dự đoán) là chất lượng mà ước lượng hoặc dự đoán được của sản phẩm phần mềm tại cuối mỗi giai đoạn trong quá trình phát triển đối với mỗi tiêu chí chất lượng, dựa trên những hiểu biết về chất lượng trong.

Chất lượng ngoài là tổng hợp của các tiêu chí của sản phẩm phần mềm theo cách nhìn từ bên ngoài. Đó là chất lượng khi phần mềm hoạt động, thường được đo kiểm, ước lượng trong khi kiểm thử trong môi trường giả lập với dữ liệu giả lập, sử dụng phương pháp đo ngoài. Trong quá trình kiểm thử, hầu hết các lỗi cần được phát hiện và khắc phục. Tuy nhiên, sau kiểm thử, vẫn còn lại một số lỗi. Bởi vì rất khó để sửa chữa kiến trúc và các vấn đề liên quan đến thiết kế cơ bản của phần mềm, nên thiết kế cơ bản của phần mềm thường không thay đổi khi kiểm thử.

Chất lượng sử dụng ước lượng (dự đoán) là chất lượng mà ước lượng hay dự đoán được của sản phẩm phần mềm tại cuối mỗi giai đoạn phát triển đối với mỗi tiêu chí chất lượng sử dụng, dựa trên hiểu biết về chất lượng trong và ngoài.

Chất lượng sử dụng là cách nhìn của người dùng về chất lượng của sản phẩm phần mềm khi nó được sử dụng trong một môi trường và hoàn cảnh cụ thể. Nó xác định phạm vi mà người sử dụng có thể đạt được mục đích của mình trong một môi trường cụ thể, hơn là xác định các tiêu chí của bản thân phần mềm.

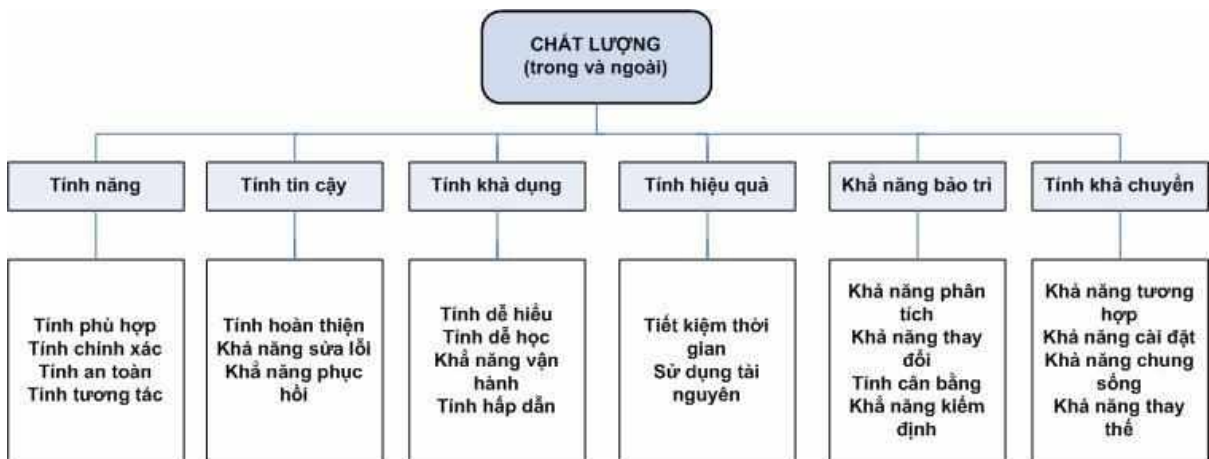
Chất lượng trong môi trường của người sử dụng có thể khác với trong môi trường của người phát triển, đó là do sự khác nhau giữa nhu cầu và khả năng của những người sử dụng khác nhau, và sự khác nhau giữa các phần cứng và môi trường. Người sử dụng chỉ đánh giá các tiêu chí của phần mềm mà họ dùng tới. Đôi khi, các thuộc tính của phần mềm, xác định bởi người sử dụng trong khi phân tích yêu cầu không đáp ứng được nhu cầu của người sử dụng trong quá trình sử dụng, đó là do những thay đổi yêu cầu của người sử dụng và các khó khăn trong việc xác định nhu cầu.

Mô hình chất lượng

Chất lượng sản phẩm phần mềm có thể được đánh giá qua một mô hình chất lượng cụ thể. Sản phẩm phần mềm nên được phân tách theo cấp bậc vào một mô hình phần mềm với những tiêu chí và những tiêu chí con, sao cho có thể sử dụng chúng như một danh sách để kiểm tra những vấn đề phát sinh liên quan đến chất lượng.

Mô hình chất lượng trong và mô hình chất lượng ngoài

Mô hình chất lượng ISO-9126 trên thực tế được mô tả là một phương pháp phân loại và chia nhỏ những thuộc tính chất lượng, nhằm tạo nên những đại lượng đo đếm được dùng để kiểm định chất lượng của sản phẩm phần mềm. Mô hình chất lượng trong và chất lượng ngoài của sản phẩm trong ISO-9126 thể hiện trên hình 2.



Hình 4.7: Mô hình chất lượng cho chất lượng trong và ngoài

Mỗi tiêu chí chất lượng, tiêu chí chất lượng con của phần mềm đều được định nghĩa. Với mỗi tiêu chí và các tiêu chí con, khả năng của phần mềm được xác định bằng tập các thuộc tính trong có thể đo đạc được. Các tiêu chí và các tiêu chí con cũng có thể đo đạc trong phạm vi khả năng của hệ thống chứa phần mềm.

Tính chức năng

Khả năng của phần mềm cung cấp các chức năng đáp ứng được nhu cầu sử dụng khi phần mềm làm việc trong điều kiện cụ thể.

- Tính phù hợp: là khả năng của một phần mềm có thể cung cấp một tập các chức năng thích hợp cho công việc cụ thể phục vụ mục đích của người sử dụng.

- Tính chính xác: là khả năng của phần mềm có thể cung cấp các kết quả hay hiệu quả đúng đắn hoặc chấp nhận được với độ chính xác cần thiết.

- Khả năng hợp tác làm việc: khả năng tương tác với một hoặc một vài hệ thống cụ thể của phần mềm.

- Tính an toàn: khả năng bảo vệ thông tin và dữ liệu của sản phẩm phần mềm, sao cho người, hệ thống không được phép thì không thể truy cập, đọc hay chỉnh sửa chúng.

- Tính năng phù hợp: các phần mềm theo các chuẩn, quy ước, quy định.

Tính tin cậy

Là khả năng của phần mềm có thể hoạt động ổn định trong những điều kiện cụ thể.

- Tính hoàn thiện: khả năng tránh các kết quả sai

- Khả năng chịu lỗi: khả năng của phần mềm hoạt động ổn định tại một mức độ cả trong trường hợp có lỗi xảy ra ở phần mềm hoặc có những vi phạm trong giao diện.

- Khả năng phục hồi: khả năng của phần mềm có thể tái thiết lại hoạt động tại một mức xác định và khôi phục lại những dữ liệu có liên quan trực tiếp đến lỗi.

- Tính tin cậy phù hợp: phần mềm thoả mãn các chuẩn, quy ước, quy định.

- Tính khả dụng: Là khả năng của phần mềm có thể hiểu được, học được, sử dụng được và hấp dẫn người sử dụng trong từng trường hợp sử dụng cụ thể.

- Có thể hiểu được: người dùng có thể hiểu được xem phần mềm có hợp với họ không và sử dụng chúng thế nào cho những công việc cụ thể.

- Có thể học được: người sử dụng có thể học các ứng dụng của phần mềm.

- Có thể sử dụng được: khả năng của phần mềm cho phép người dùng sử dụng và điều khiển nó.

- Tính hấp dẫn: khả năng hấp dẫn người sử dụng của phần mềm

- Tính khả dụng phù hợp: phần mềm thoả mãn các chuẩn, quy ước, quy định

- Tính hiệu quả: Khả năng của phần mềm có thể hoạt động một cách hợp lý, tương ứng với lượng tài nguyên nó sử dụng, trong điều kiện cụ thể.

- Đáp ứng thời gian: khả năng của phần mềm có thể đưa ra một trả lời, một thời gian xử lý và một tốc độ thông lượng hợp lý khi nó thực hiện công việc của mình, dưới một điều kiện làm việc xác định.

- Sử dụng tài nguyên: khả năng của phần mềm có thể sử dụng một lượng, một loại tài nguyên hợp lý để thực hiện công việc trong những điều kiện cụ thể.

- Tính hiệu quả phù hợp: thoả mãn các chuẩn, quy ước, quy định.

- Khả năng bảo hành, bảo trì: Khả năng của phần mềm có thể chỉnh sửa. Việc chỉnh sửa bao gồm: sửa lại cho đúng, cải tiến và làm phần mềm thích nghi được với những thay đổi của môi trường, của yêu cầu và của chức năng xác định.

- Có thể phân tích được: phần mềm có thể được chẩn đoán để tìm những thiếu sót hay những nguyên nhân gây lỗi hoặc để xác định những phần cần sửa.

- Có thể thay đổi được: phần mềm có thể chấp nhận một số thay đổi cụ thể trong quá trình triển khai.

- Tính ổn định: khả năng tránh những tác động không mong muốn khi chỉnh sửa phần mềm.

- Có thể kiểm tra được: khả năng cho phép đánh giá được phần mềm chỉnh sửa.

- Khả năng bảo hành bảo trì phù hợp: thoả mãn các chuẩn, quy ước, quy định.

- Tính khả chuyển: Là khả năng của phần mềm cho phép nó có thể được chuyển từ môi trường này sang môi trường khác.

- Khả năng thích nghi: khả năng của phần mềm có thể thích nghi với nhiều môi trường khác nhau mà không cần phải thay đổi.

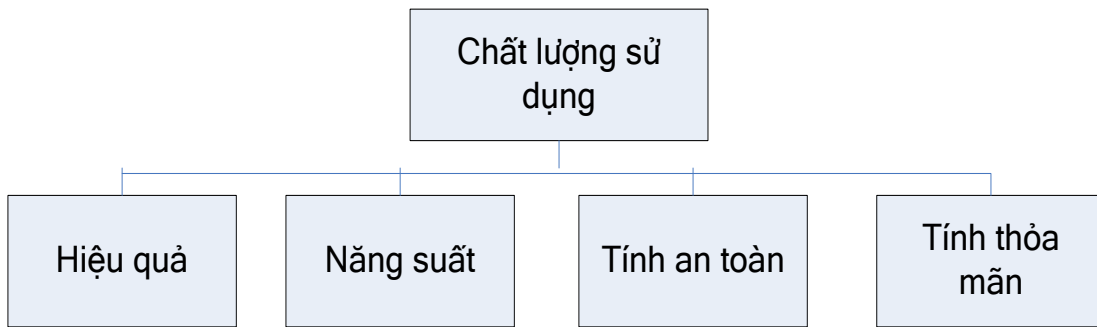
- Có thể cài đặt được: phần mềm có thể cài đặt được trên những môi trường cụ thể.

- Khả năng cùng tồn tại: phần mềm có thể cùng tồn tại với những phần mềm độc lập khác trong một môi trường chung, cùng chia sẻ những tài nguyên chung.

- Khả năng thay thế: phần mềm có thể dùng thay thế cho một phần mềm khác, với cùng mục đích và trong cùng môi trường.

- Tính khả chuyển phù hợp: thoả mãn các chuẩn, quy ước, quy định.

Mô hình chất lượng sử dụng: Chất lượng sử dụng bao gồm 4 tiêu chí: tính hiệu quả, năng suất, tính an toàn và tính thoả mãn (hình 3).



Hình 4.8: Mô hình chất lượng sử dụng

a) Tính hiệu quả: khả năng của phần mềm cho phép người dùng đạt được mục đích một cách chính xác và hoàn toàn, trong điều kiện làm việc cụ thể.

b) Tính năng suất: khả năng của phần mềm cho phép người dùng sử dụng lượng tài nguyên hợp lý tương đối để thu được hiệu quả công việc trong những hoàn cảnh cụ thể.

c) Tính an toàn: phần mềm có thể đáp ứng mức độ rủi ro chấp nhận được đối với người sử dụng, phần mềm, thuộc tính, hoặc môi trường trong điều kiện cụ thể.

d) Tính thoả mãn: phần mềm có khả năng làm thoả mãn người sử dụng trong từng điều kiện cụ thể.

Phương pháp đánh giá chất lượng theo ISO-9126

<i>Tên phương pháp đo</i>	<i>Mục đích phương pháp đo</i>	<i>Phương thức áp dụng</i>	<i>Phương pháp đo, cách thức tính toán các thành phần dữ liệu</i>
Đầy đủ chức năng	Các chức năng được đáp ứng đầy đủ như thế nào ?	So sánh số lượng các chức năng thực hiện các nhiệm vụ đặc tả và số lượng chức năng được đánh giá	$X = 1 - A/B$ A = Số lượng các chức năng có lỗi B = Số lượng các chức năng được đánh giá
Tính toàn vẹn triển khai chức năng	Việc triển khai các chức năng như thế nào so với các đặc tính yêu cầu	Thực hiện các bài kiểm tra chức năng cho hệ thống theo các đặc tính yêu cầu. Tính toán số lượng các chức năng bị mất được phát hiện trong quá trình đánh giá và so sánh với số lượng các chức năng được miêu tả trong đặc tính yêu cầu.	$X = 1 - A / B$ A = Số lượng các chức năng bị mất được phát hiện trong quá trình đánh giá B = Số lượng các chức năng trong đặc tính yêu cầu

ISO/IEC 14598

ISO/IEC 14598 bao gồm 6 phần chính dưới tiêu đề chung: Công nghệ thông tin – Đánh giá sản phẩm phần mềm.

Phần 1: Tổng quan

Phần 2: Lập kế hoạch và quản lý

Phần 3: Quy trình cho người phát triển

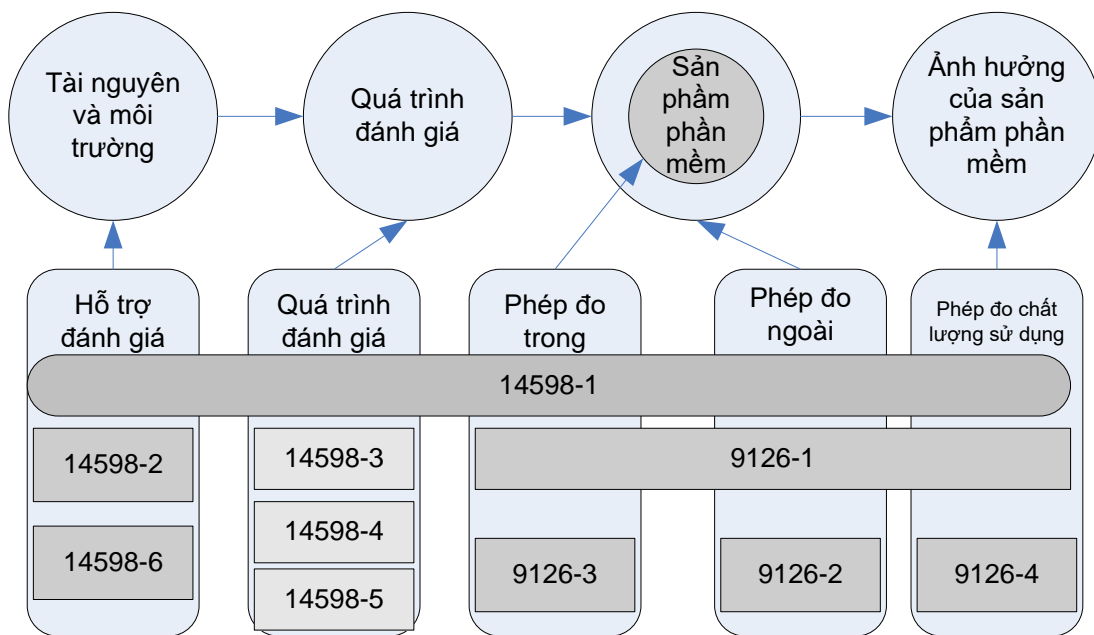
Phần 4: Quy trình cho người sử dụng

Phần 5: Quy trình cho người đánh giá

Phần 6: Tài liệu các hợp phần đánh giá

Phần I của chuẩn ISO/IEC 14598 giới thiệu chung, đưa ra quy trình đánh giá chung cho sản phẩm phần mềm. Nó cung cấp một cái nhìn tổng quát về các tiêu chuẩn và giải thích mối quan hệ giữa ISO/IEC 14598 và mô hình chất lượng trong ISO/IEC 9126 như trong hình 4. Phần này xác định một cách rõ ràng các thuật ngữ công nghệ được sử dụng trong các phần khác, bao gồm các yêu cầu chung, đánh giá chất lượng phần mềm và các khái niệm chung.

Bên cạnh đó phần I cung cấp mô hình cơ bản để đánh giá chất lượng cho các sản phẩm phần mềm và các yêu cầu cho các phương pháp đo và đánh giá sản phẩm phần mềm. ISO/IEC 14598 được sử dụng cho người phát triển, người sử dụng và những người đánh giá độc lập có trách nhiệm đánh giá sản phẩm phần mềm.



Hình 4.9: Mối liên hệ giữa tiêu chuẩn ISO 9126 và ISO 14598.

Quá trình đánh giá sản phẩm phần mềm trong ISO 14598 dành cho 3 đối tượng khác nhau:

- *Người phát triển*: các tiến trình đánh giá phần mềm dành cho người phát triển có thể được áp dụng cho các tổ chức có kế hoạch phát triển các sản phẩm mới hay cải tiến các sản phẩm sẵn có. Nó cũng phù hợp với các

tổ chức dự định sử dụng chính nhân viên kỹ thuật của mình để thực hiện đánh giá sản phẩm phần mềm.

- *Người mua sản phẩm*: các quá trình đánh giá phần mềm dành cho người mua sản phẩm có thể được sử dụng bởi các tổ chức có kế hoạch khai thác, tái sử dụng các sản phẩm phần mềm đã có hoặc sắp phát triển. Nó có thể được sử dụng để xác định xem sản phẩm phần mềm có chấp nhận được không hoặc để lựa chọn phần mềm thích hợp giữa các sản phẩm cùng chức năng.
- *Người đánh giá*: người đánh giá (thường làm việc cho một bên thứ 3) sử dụng những quá trình đánh giá riêng để có những kết luận độc lập về một sản phẩm phần mềm. Người phát triển hệ thống, người khai thác hay một bên liên quan nào đó có thể yêu cầu thực hiện những quá trình này.

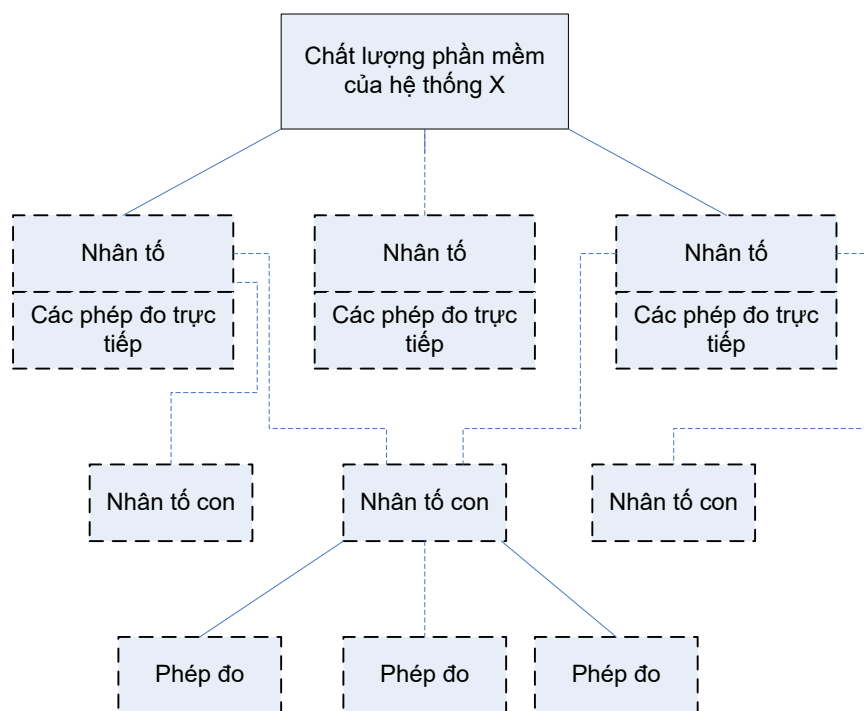
IEEE 1061 (1992)

Chuẩn này cung cấp phương pháp luận để xác định yêu cầu chất lượng cần đạt, đồng thời chỉ rõ cách phân tích, ứng dụng quy trình kiểm tra tham số phần mềm. Phương pháp luận này áp dụng cho tất cả các giai đoạn trong chu trình của bất cứ phần mềm nào.

Đối tượng sử dụng:

- Người quản lý dự án để xác định yêu cầu chất lượng cho hệ thống.
- Người phát triển hệ thống nên thiết kế phần mềm thế nào nhằm đáp ứng yêu cầu chất lượng.
- Người thẩm định phần mềm có thể đánh giá phần mềm có đạt được các yêu cầu chất lượng không.
- Người vận hành hệ thống để quản lý các thay đổi trong quá trình nâng cấp sản phẩm.
- Người sử dụng đặc tả yêu cầu chất lượng cho hệ thống.

Khung tham số chất lượng phần mềm (hình 5) trong tiêu chuẩn IEEE 1061 – 1992.



Hình 4.10: Khung tham số chất lượng phần mềm

Phương pháp xác định tham số chất lượng

Các bước trong phương pháp:

- Xây dựng yêu cầu chất lượng phần mềm.
- Xác định tham số chất lượng.
- Áp dụng tham số chất lượng: thu thập thông tin và áp dụng tham số chất lượng cho từng pha trong chu trình phần mềm.
- Phân tích kết quả tham số chất lượng phần mềm. Các kết quả phải được phân tích và báo cáo hỗ trợ điều chỉnh hướng phát triển phần mềm và hoàn thiện nó.

Thông qua tham số phần mềm.

Các bước	Đầu ra
Xây dựng yêu cầu chất lượng phần mềm	Tập yêu cầu chất lượng
Xác định tham số chất lượng	-Xác định tham số chất lượng - Thiết lập tập tham số - Phân tích giá thành-lợi ích
Áp dụng tham số chất lượng	- Mô tả thông tin sản phẩm - Tham số/thông tin sản phẩm - Ma trận - Lên kế hoạch
Phân tích kết quả tham số chất lượng phần mềm	Các thay đổi chu trình tổ chức và phát triển
Thông qua tham số phần mềm	Quá trình thông qua các kết quả

ISO 12119

Nội dung Bộ tiêu chuẩn ISO/IEC 12119 là về đánh giá gói sản phẩm phần mềm. Tiêu chuẩn ISO/IEC 12119:1994 được áp dụng để đánh giá chung cho các tài liệu hướng dẫn, tài liệu mô tả sản phẩm, chương trình và dữ liệu và kiểm thử phần mềm.

Mô tả sản phẩm: bao gồm các yêu cầu chung về mặt nội dung, các chỉ số và đưa ra kết quả về tính chức năng, độ tin cậy, tính khả dụng, tính hiệu quả, khả năng bảo hành bảo trì và tính khả chuyên.

Tài liệu hướng dẫn sử dụng: phải bao gồm các thông tin cần thiết cho việc sử dụng sản phẩm đó. Tất cả các chức năng có thể được truy xuất bởi người sử dụng trong chương trình sẽ được mô tả đầy đủ trong tài liệu sử dụng và bao gồm các yêu cầu về:

- Tính đầy đủ
- Tính chính xác

- Tính thống nhất
- Tính dễ hiểu
- Tính tổng quan

Chương trình và dữ liệu: bao gồm 06 tiêu chí giống với các tiêu chí trong mô hình chất lượng của tiêu chuẩn ISO-9126.

- Tính năng (Functionality)
- Độ tin cậy (Reliability)
- Tính khả dụng (Usability)
- Tính hiệu quả (Efficiency)
- Khả năng bảo hành bảo trì (Maintainability)
- Tính khả chuyển (Portability)

Xây dựng tiêu chí và tiêu chuẩn đánh giá chất lượng sản phẩm phần mềm

Qua việc thực hiện nghiên cứu tiêu chí đánh giá sản phẩm phần mềm, hướng dẫn đánh giá của các doanh nghiệp trong nước và các tổ chức quốc tế, nhóm chủ trì xây dựng tiêu chí đánh giá sản phẩm phần mềm theo mô hình chất lượng ISO-9126, đây được coi là mô hình chất lượng ưu việt nhất cho việc đánh giá sản phẩm phần mềm. Sử dụng mô hình chất lượng theo ISO-9126 (gồm 4 phần) và quy trình đánh giá sản phẩm phần mềm của ISO-14598 (gồm 6 phần) chúng ta có thể đánh giá sản phẩm phần mềm một cách toàn diện, từ lúc phát triển tới khi hoàn thành và cả khi sử dụng phần mềm.

Chất lượng trong và chất lượng ngoài

06 tiêu chí để đánh giá *chất lượng trong* và *chất lượng ngoài* của sản phẩm phần mềm và được trình bày trong dự thảo tiêu chuẩn phần 1 và 2, bao gồm:

- Tính năng (Functionality)
- Tính tin cậy (Reliability)
- Tính khả dụng (Usability)
- Tính hiệu quả (Efficiency)
- Khả năng bảo hành bảo trì (Maintainability)

- Tính khả chuyển (Portability)

Trong đó mỗi tiêu chí lại được chia thành những tiêu chí nhỏ hơn:

Tính năng (Functionality): Là khả năng của phần mềm cung cấp các chức năng thỏa mãn các yêu cầu được xác định rõ ràng cũng như các yêu cầu 'không rõ ràng' khi phần mềm được sử dụng trong những hoàn cảnh cụ thể. Bao gồm 4 tiêu chí nhỏ:

- Tính phù hợp (Suitability)
- Tính chính xác (Accuracy)
- Khả năng tương tác (Interoperability)
- Tính bảo mật/an toàn (Security)
- Tính tin cậy (Reliability): Là khả năng của phần mềm duy trì mức hiệu năng được chỉ định rõ khi sử dụng dưới những điều kiện cụ thể. Bao gồm các tiêu chí nhỏ:
 - Tính hoàn thiện (Maturity)
 - Khả năng chịu lỗi (Fault tolerant)
 - Khả năng phục hồi (Recoverability)
 - Tính khả dụng (Usability): Là khả năng của phần mềm để có thể hiểu được, học hỏi được, sử dụng được và hấp dẫn đối với người sử dụng.
 - Dễ hiểu (Understandability)
 - Dễ học (Learnability)
 - Khả năng vận hành (Operability)
 - Tính hấp dẫn (Attractiveness)
 - Tính hiệu quả (Efficiency): Là khả năng của phần mềm cung cấp hiệu năng thích hợp nhằm tiết kiệm tối đa tài nguyên và tăng tối đa hiệu suất công việc, dưới những điều kiện sử dụng nhất định.
 - Thời gian xử lý (Time behavior)
 - Sử dụng tài nguyên (Utilization)

- Khả năng bảo trì (Maintainability): Là khả năng của phần mềm cho phép sửa đổi, nâng cấp, bao gồm sửa chữa, cải tiến hoặc thích nghi của phần mềm thay đổi cho phù hợp với môi trường, các yêu cầu và chức năng mới.
- Khả năng phân tích (Analysability)
- Khả năng thay đổi được (Changeability)
- Tính ổn định (Stability)
- Khả năng kiểm thử được (Testability)
- Tính khả chuyển (Portability): Là khả năng của phần mềm có thể chuyển được từ môi trường này sang môi trường khác.
- Khả năng thích nghi (Adaptability)
- Khả năng cài đặt (Installability)
- Khả năng chung sống (Co-existence)
- Khả năng thay thế được (Replaceability)

Chất lượng sử dụng

04 tiêu chí đánh giá chất lượng sử dụng đã được lựa chọn và đưa vào tiêu chuẩn phần 3, bao gồm:

- Tính hiệu quả: khả năng của phần mềm cho phép người dùng đạt được mục đích một cách chính xác và hoàn toàn, trong điều kiện làm việc cụ thể.
- Tính năng suất: khả năng của phần mềm cho phép người dùng sử dụng lượng tài nguyên hợp lý tương đối để thu được hiệu quả công việc trong những hoàn cảnh cụ thể.
- Tính an toàn: phần mềm có thể đáp ứng mức độ rủi ro chấp nhận được đối với người sử dụng, phần mềm, thuộc tính, hoặc môi trường trong điều kiện cụ thể.
- Tính thoả mãn: phần mềm có khả năng làm thoả mãn người sử dụng trong từng điều kiện cụ thể.

Xây dựng quy trình đánh giá sản phẩm phần mềm

Quy trình đánh giá sản phẩm phần mềm được xây dựng theo ISO/IEC 14598.

Quy trình đánh giá sản phẩm phần mềm được thực hiện theo 4 bước được mô tả trong hình 4.11.

Xác lập mục đích đánh giá



Hình 4.11: Quy trình đánh giá sản phẩm phần mềm

Bước 1: Thiết lập các yêu cầu đánh giá

- Xác lập mục đích đánh giá
- Xác định loại sản phẩm cần đánh giá
- Xây dựng mô hình chất lượng
- Bước 2: Xác lập cơ chế đánh giá
- Xác lập các đại lượng và độ đo
- Thiết lập mức đo chuẩn
- Thiết lập các tiêu chí đánh giá

Bước 3: Thiết kế kế hoạch đánh giá sản phẩm phần mềm

- Quản lý ở mức tổ chức

- Hỗ trợ việc quản lý dự án
- Bước 4: Thực hiện đánh giá
- Bao gồm các bước chính sau:
- Thực hiện đo
- So sánh với tiêu chí đánh giá
- Đánh giá kết quả thu được

4.3 Đánh giá và hiệu chỉnh hệ thống

4.3.1 Các tiêu chí cơ bản

Hệ thống webGIS sông Cả đáp ứng đầy đủ các tiêu chí của một webGIS truyền thống. Đối tượng chính của webGIS là các đối tượng địa lý (có dữ liệu thuộc tính và có tọa độ), trong đó bao gồm cả 2 định dạng chính là dạng vector và raster.

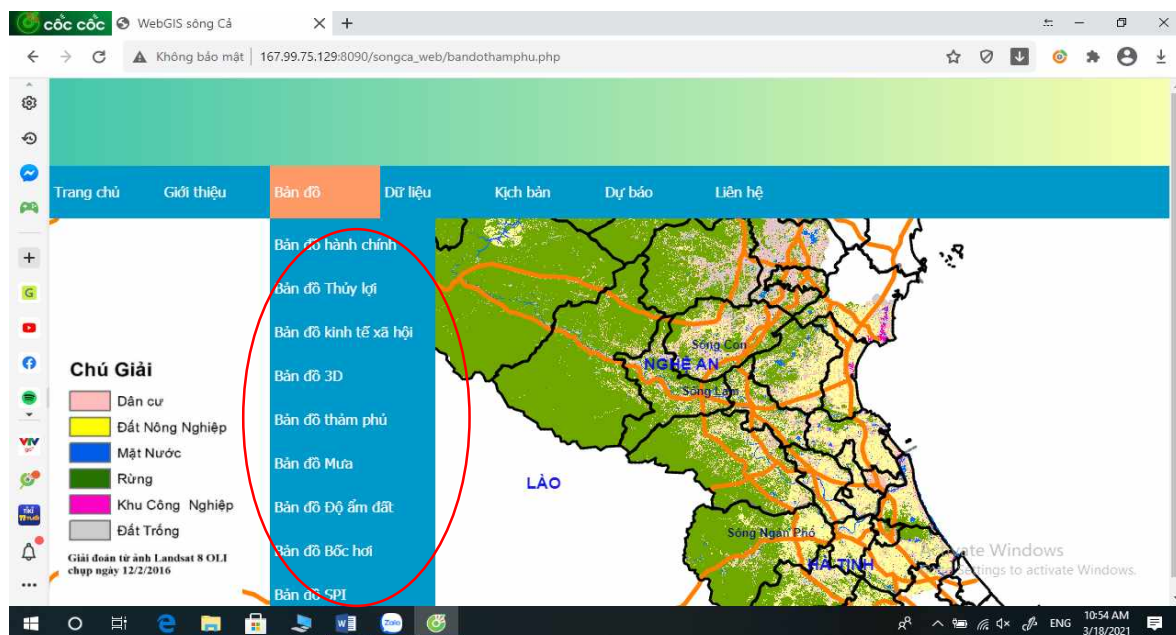
Phần chính của hệ thống webGIS nằm ở tab “Bản đồ” của thanh menu chính. Trong đó các loại bản đồ được hệ thống thành các nhóm bao gồm:

- Bản đồ hành chính
- Công trình thủy lợi
- Bản đồ 3D
- Bản đồ độ ẩm đất
- Bản đồ thảm phủ
- Bản đồ Mưa
- Bản đồ bốc hơi
- Bản đồ kinh tế xã hội

Trong đó Bản đồ độ ẩm đất, Bản đồ thảm phủ, Bản đồ bốc hơi và Bản đồ Mưa là các nhóm bản đồ raster, là kết quả được chiết xuất từ dữ liệu viễn thám. Các bản đồ còn lại như Bản đồ hành chính, bản đồ công trình thủy lợi, bản đồ kinh tế xã hội được xây dựng trong phần mềm mã nguồn mở QGIS. Các Bản đồ được hiển thị trên nền web ở hệ quy chiếu WGS84 (epsg: 4326). Các bản đồ nền

Nghiên cứu đề xuất phương án phối hợp vận hành điều tiết nước hợp lý các hồ chứa thủy lợi, thủy điện trên sông Cả phục vụ cấp nước và phòng, chống lũ cho hạ du

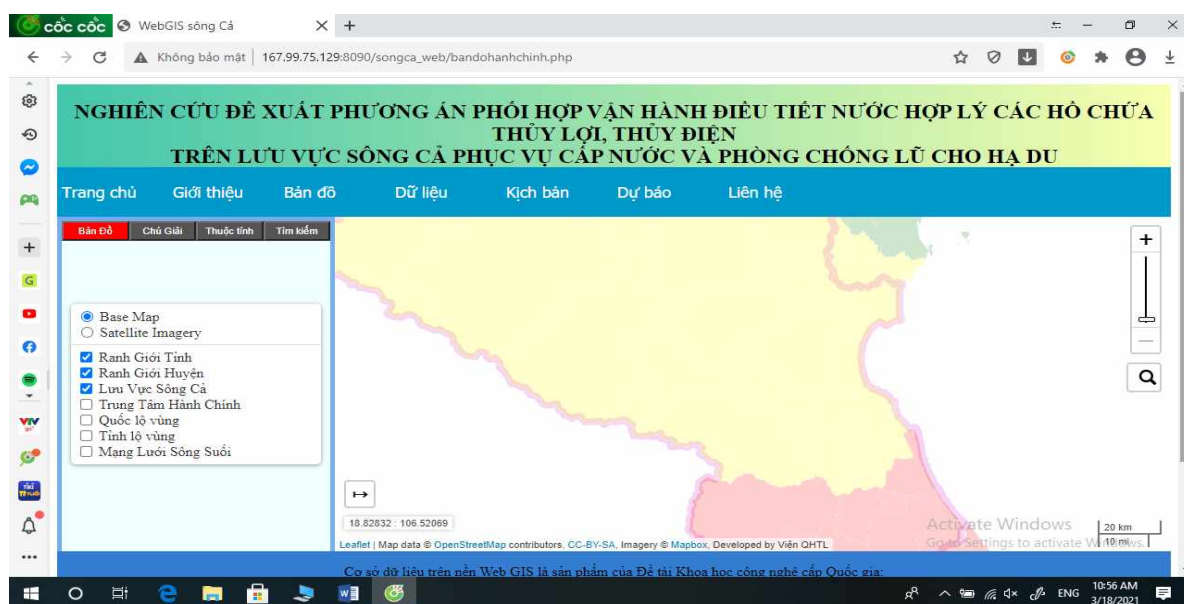
bao gồm: Bản đồ ảnh vệ tinh của ESRI và bản đồ đường phố của OpenStreetMap đều hiển thị bình thường.



Hình 4.12: các nhóm Bản đồ của hệ thống webGIS sông Cả

Hệ thống webGIS sông Cả có đầy đủ các chức năng chính khi thao tác với dữ liệu, bao gồm: phóng to, thu nhỏ, tìm kiếm địa danh, bật và tắt các lớp.

Kết quả kiểm tra thao tác cho thấy các chức năng cơ bản của webGIS hoạt động bình thường, phản ánh đúng yêu cầu của người dùng. Bên cạnh đó, không phát hiện lỗi trong quá trình thao tác. Các công cụ rất dễ vận hành và dễ sử dụng.



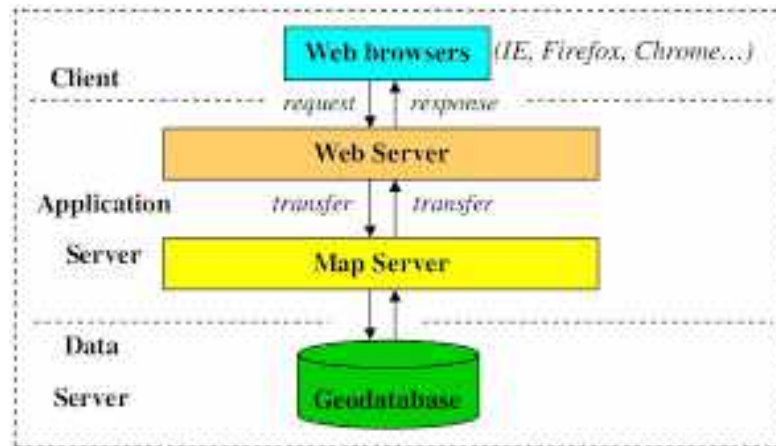
Hình 4.13: Các chức năng cơ bản của webGIS

4.3.2 Hiệu suất xử lý

Các tác vụ đòi hỏi nhiều tài nguyên máy tính và tốc độ mạng đã được tập trung kiểm tra xử lý, đặc biệt là các ứng dụng 3D và các thao tác với lớp dữ liệu lớn.

4.3.3 Tính dễ hiểu

Kiến trúc của webGIS sông Cả rất dễ hiểu



Hình 4.14: Kiến trúc áp dụng trong webGIS sông Cả

WebGIS sông Cả được xây dựng theo kiến trúc chuẩn bao gồm 3 tầng: tầng trình bày (client), tầng dữ liệu (data server), và tầng giao dịch (application server)

Tầng trình bày: Thông thường chỉ là các trình duyệt Internet Explorer, Mozilla Firefox, Chrome ... để mở các trang web theo URL đã được định sẵn là <http://www.websong.tk>. Ứng dụng client ở đây là một website được viết theo công nghệ chuẩn w3c. Các Client đôi khi cũng là một ứng dụng desktop tương tự như phần mềm MapInfo, ArcMap...

Tầng giao dịch

Thông thường, tầng giao dịch được tích hợp trong một webserver nào đó, ví dụ như Tomcat, Apache, Internet Information Server. Đó là một ứng dụng phía server nhiệm vụ chính của nó thường là tiếp nhận các yêu cầu từ client, lấy dữ liệu từ cơ sở dữ liệu theo yêu cầu client, trình bày dữ liệu theo cấu hình định sẵn hoặc theo yêu cầu của client và trả kết quả về theo yêu cầu. Tùy theo yêu cầu của client mà kết quả về khác nhau : có thể là một hình ảnh dạng bitmap (jpeg, gif,

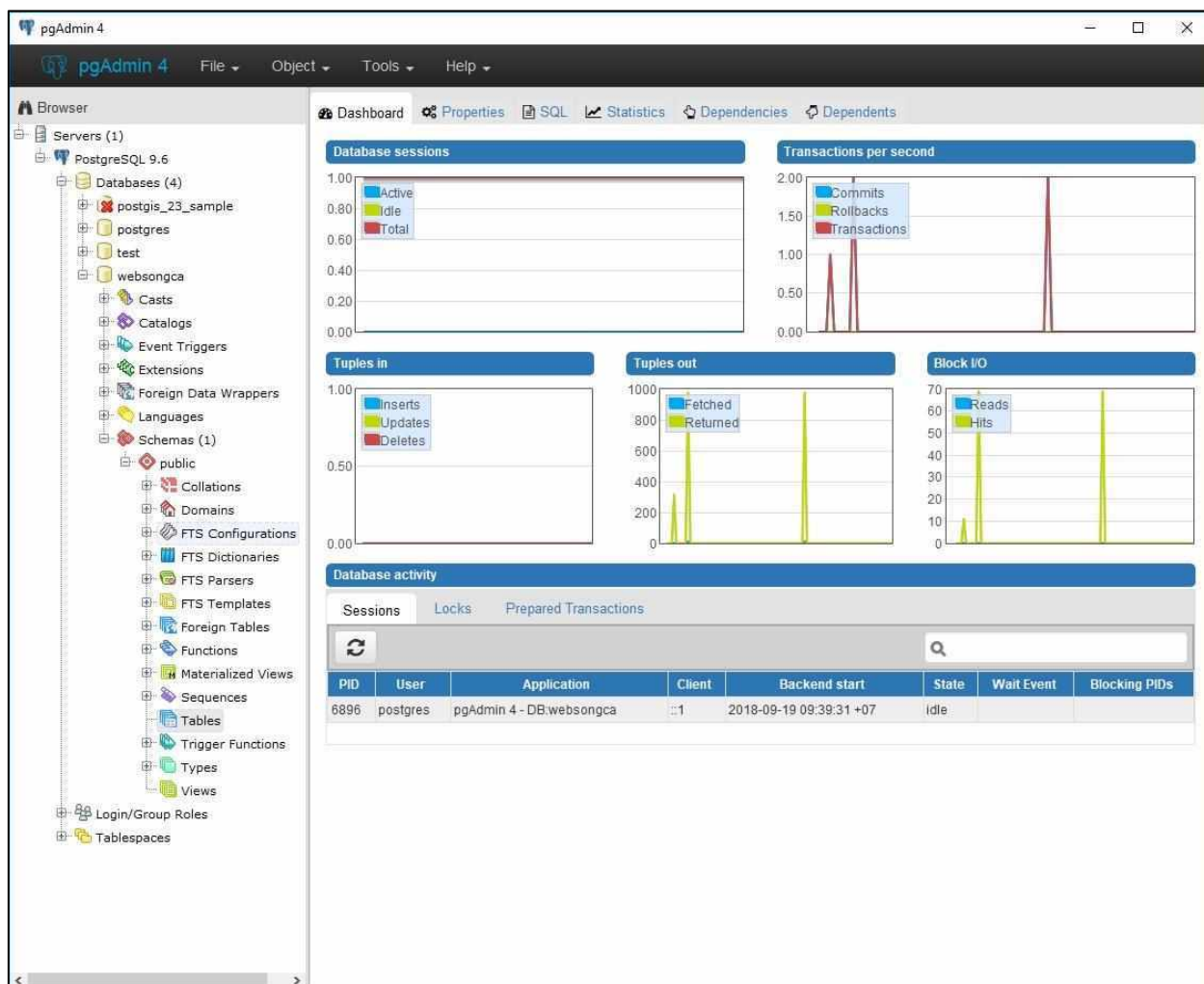
png) hay dạng vector được mã hóa như SVG, KML, GML,... Một khi dạng vector được trả về thì việc trình bày hình ảnh bản đồ được đảm nhiệm bởi Client, thậm chí client có thể xử lý một số bài toán về không gian. Thông thường các response và request đều theo chuẩn HTTP POST hoặc GET.

Tầng dữ liệu: Là nơi lưu trữ các dữ liệu địa lý bao gồm cả các dữ liệu không gian và phi không gian.

Các dữ liệu này được quản trị bởi các hệ quản trị cơ sở dữ liệu như ORACLE, MS SQL SERVER, ESRI SDE, POSTGRESQL,... hoặc là các file dữ liệu dạng flat như shapefile, tab, XML,... Các dữ liệu này được thiết kế, cài đặt và xây dựng theo từng quy trình, từng quy mô bài toán ... mà lựa chọn hệ quản trị cơ sở dữ liệu phù hợp.

Cơ sở dữ liệu không gian sẽ được dùng để quản lý và truy xuất dữ liệu không gian, được đặt trên data server. Nhà kho hay nơi lưu trữ được dùng để lưu trữ và duy trì những siêu dữ liệu về dữ liệu không gian tại những data server khác nhau. Dựa trên những thành phần quản lý dữ liệu, ứng dụng server và mô hình server được dùng cho ứng dụng hệ thống để tính toán thông tin không gian thông qua các hàm cụ thể. Tất cả kết quả tính toán của ứng dụng server sẽ được gửi đến web server để thêm vào các gói HTML, gửi cho phía client và hiển thị nơi trình duyệt web.

Trong ứng dụng webGIS sông Cả, dữ liệu được lưu trữ và quản lý bằng hệ quản trị cơ sở dữ liệu postgresSQL. Đây là 1 lựa chọn đúng đắn bởi PostgreSQL không chỉ là cơ sở dữ liệu quan hệ, nó là quan hệ hướng đối tượng. Điều này cung cấp cho nó một vài lợi thế so với các cơ sở dữ liệu SQL mã nguồn mở khác như MySQL, MariaDB và Firebird.



Hình 4.16: giao diện hệ quản trị CSDL PostgreSQL

Bên cạnh đó, dữ liệu địa lý là thành phần chính của hệ thống webGis, là phần cốt lõi của ứng dụng. PostgreSQL từ lâu đã hỗ trợ một loạt kiểu dữ liệu hình học như points, lines, circles, và polygons. Kiểu dữ liệu PATH là một trong số đó. Một path bao gồm nhiều điểm trong một chuỗi và có thể là mở (điểm đầu và điểm cuối không nối với nhau) hoặc đóng (điểm đầu và điểm cuối nối với nhau). Hãy sử dụng một ví dụ đường tròn đi bộ đường dài như là một path. Trong trường hợp này, đường tròn đi bộ đường dài của tôi là một vòng lặp vì thế điểm bắt đầu và điểm kết thúc của tôi được nối với nhau, và vì vậy path của tôi là đóng. Cặp ngoặc đơn xung quanh tập các tọa độ báo hiệu một path đóng trong khi một cặp ngoặc [] báo hiệu path mở.

Nghiên cứu đề xuất phương án phối hợp vận hành điều tiết nước hợp lý các hồ chứa thủy lợi, thủy điện trên sông Cả phục vụ cấp nước và phòng, chống lũ cho hạ du

```
1 {
2   "type": "FeatureCollection",
3   "crs": { "type": "name", "properties": { "name": "urn:ogc:def:crs:OGC:1.3:CRS84" } },
4   "features": [
5     { "type": "Feature", "properties": { "type": 2 }, "geometry": { "type": "MultiLineString", "coordinates": [ [ [
6       104.90605, 19.04011, 0.0 ], [ 104.90605, 19.04011, 0.0 ], [ 104.90606, 19.0401, 0.0 ], [ 104.90606, 19.0401, 0.0 ], [
7       104.90606, 19.0401, 0.0 ], [ 104.90605, 19.04011, 0.0 ], [ 104.90605, 19.04011, 0.0 ], [ 104.90605, 19.04011, 0.0 ], [
8       104.90605, 19.04011, 0.0 ], [ 104.90605, 19.0401, 0.0 ], [ 104.90605, 19.04011, 0.0 ], [ 104.90605, 19.04011, 0.0 ], [
9       104.90606, 19.0401, 0.0 ], [ 104.90606, 19.0401, 0.0 ], [ 104.90606, 19.0401, 0.0 ], [ 104.90606, 19.0401, 0.0 ], [
10      104.90605, 19.0401, 0.0 ], [ 104.90605, 19.0401, 0.0 ] ] ] ] },
11    { "type": "Feature", "properties": { "type": 2 }, "geometry": { "type": "MultiLineString", "coordinates": [ [ [
12      104.78129, 19.16871, 0.0 ], [ 104.7813, 19.16872, 0.0 ] ] ] } },
13    { "type": "Feature", "properties": { "type": 2 }, "geometry": { "type": "MultiLineString", "coordinates": [ [ [
14      104.7801, 19.17441, 0.0 ], [ 104.7801, 19.17442, 0.0 ] ] ] } },
15    { "type": "Feature", "properties": { "type": 1 }, "geometry": { "type": "MultiLineString", "coordinates": [ [ [
16      104.10964, 19.5173, 0.0 ], [ 104.10962, 19.51732, 0.0 ], [ 104.1096, 19.51734, 0.0 ], [ 104.10958, 19.51737, 0.0 ], [
17      104.10956, 19.51738, 0.0 ], [ 104.10953, 19.5174, 0.0 ], [ 104.10951, 19.51741, 0.0 ], [ 104.10946, 19.51743, 0.0 ], [
18      104.1094, 19.51745, 0.0 ], [ 104.10934, 19.51745, 0.0 ], [ 104.1092, 19.51739, 0.0 ], [ 104.1091, 19.51736, 0.0 ], [
19      104.10899, 19.51735, 0.0 ], [ 104.10883, 19.5174, 0.0 ], [ 104.10861, 19.5175, 0.0 ], [ 104.10856, 19.51754, 0.0 ], [
20      104.10855, 19.51756, 0.0 ], [ 104.10854, 19.51766, 0.0 ], [ 104.10851, 19.51772, 0.0 ], [ 104.10836, 19.5178, 0.0 ], [
21      104.10832, 19.51783, 0.0 ], [ 104.1083, 19.51788, 0.0 ], [ 104.1083, 19.51795, 0.0 ], [ 104.10833, 19.51811, 0.0 ], [
22      104.1084, 19.51829, 0.0 ], [ 104.10846, 19.51844, 0.0 ], [ 104.10847, 19.5185, 0.0 ], [ 104.10847, 19.51854, 0.0 ], [
23      104.10848, 19.51857, 0.0 ], [ 104.10848, 19.51859, 0.0 ], [ 104.10847, 19.51861, 0.0 ], [ 104.10846, 19.51863, 0.0 ], [
24      104.10846, 19.51865, 0.0 ], [ 104.10845, 19.51867, 0.0 ], [ 104.10844, 19.51869, 0.0 ], [ 104.10843, 19.51871, 0.0 ], [
25      104.10841, 19.51873, 0.0 ], [ 104.10838, 19.51875, 0.0 ], [ 104.10834, 19.51876, 0.0 ], [ 104.1083, 19.51878, 0.0 ], [
26      104.10827, 19.51879, 0.0 ], [ 104.10824, 19.5188, 0.0 ], [ 104.10821, 19.51879, 0.0 ], [ 104.10818, 19.51878, 0.0 ], [
27      104.10816, 19.51876, 0.0 ], [ 104.10813, 19.51872, 0.0 ], [ 104.10811, 19.51868, 0.0 ], [ 104.10809, 19.51864, 0.0 ], [
28      104.10807, 19.51861, 0.0 ], [ 104.10805, 19.51856, 0.0 ], [ 104.10803, 19.51853, 0.0 ], [ 104.10799, 19.51846, 0.0 ], [
29      104.10784, 19.5183, 0.0 ], [ 104.1078, 19.51827, 0.0 ], [ 104.10779, 19.51827, 0.0 ], [ 104.10778, 19.51826, 0.0 ], [
30      104.10777, 19.51825, 0.0 ], [ 104.10776, 19.51824, 0.0 ], [ 104.10775, 19.51823, 0.0 ], [ 104.10764, 19.51815, 0.0 ], [
31      104.10759, 19.51811, 0.0 ], [ 104.10751, 19.51806, 0.0 ], [ 104.10742, 19.51801, 0.0 ], [ 104.10735, 19.51793, 0.0 ], [
32      104.10729, 19.51787, 0.0 ], [ 104.10722, 19.5178, 0.0 ], [ 104.10715, 19.51773, 0.0 ], [ 104.10708, 19.51766, 0.0 ], [
33      104.10701, 19.51759, 0.0 ], [ 104.10694, 19.51752, 0.0 ], [ 104.10687, 19.51745, 0.0 ], [ 104.1068, 19.51738, 0.0 ], [
34      104.10673, 19.51731, 0.0 ], [ 104.10666, 19.51724, 0.0 ], [ 104.10659, 19.51717, 0.0 ], [ 104.10652, 19.5171, 0.0 ], [
35      104.10645, 19.51703, 0.0 ], [ 104.10638, 19.51696, 0.0 ], [ 104.10631, 19.51689, 0.0 ], [ 104.10624, 19.51682, 0.0 ], [
36      104.10617, 19.51675, 0.0 ], [ 104.1061, 19.51668, 0.0 ], [ 104.10603, 19.51661, 0.0 ], [ 104.10596, 19.51654, 0.0 ], [
37      104.10589, 19.51647, 0.0 ], [ 104.10582, 19.5164, 0.0 ], [ 104.10575, 19.51633, 0.0 ], [ 104.10568, 19.51626, 0.0 ], [
38      104.10561, 19.51619, 0.0 ], [ 104.10554, 19.51612, 0.0 ], [ 104.10547, 19.51605, 0.0 ], [ 104.1054, 19.51598, 0.0 ], [
39      104.10533, 19.51591, 0.0 ], [ 104.10526, 19.51584, 0.0 ], [ 104.10519, 19.51577, 0.0 ], [ 104.10512, 19.5157, 0.0 ], [
40      104.10505, 19.51563, 0.0 ], [ 104.10498, 19.51556, 0.0 ], [ 104.10491, 19.51549, 0.0 ], [ 104.10484, 19.51542, 0.0 ], [
41      104.10477, 19.51535, 0.0 ], [ 104.1047, 19.51528, 0.0 ], [ 104.10463, 19.51521, 0.0 ], [ 104.10456, 19.51514, 0.0 ], [
42      104.10449, 19.51507, 0.0 ], [ 104.10442, 19.515, 0.0 ], [ 104.10435, 19.51493, 0.0 ], [ 104.10428, 19.51486, 0.0 ], [
43      104.10421, 19.51479, 0.0 ], [ 104.10414, 19.51472, 0.0 ], [ 104.10407, 19.51465, 0.0 ], [ 104.104, 19.51458, 0.0 ], [
44      104.10393, 19.51451, 0.0 ], [ 104.10386, 19.51444, 0.0 ], [ 104.10379, 19.51437, 0.0 ], [ 104.10372, 19.5143, 0.0 ], [
45      104.10365, 19.51423, 0.0 ], [ 104.10358, 19.51416, 0.0 ], [ 104.10351, 19.51409, 0.0 ], [ 104.10344, 19.51402, 0.0 ], [
46      104.10337, 19.51395, 0.0 ], [ 104.1033, 19.51388, 0.0 ], [ 104.10323, 19.51381, 0.0 ], [ 104.10316, 19.51374, 0.0 ], [
47      104.10309, 19.51367, 0.0 ], [ 104.10302, 19.5136, 0.0 ], [ 104.10295, 19.51353, 0.0 ], [ 104.10288, 19.51346, 0.0 ], [
48      104.10281, 19.51339, 0.0 ], [ 104.10274, 19.51332, 0.0 ], [ 104.10267, 19.51325, 0.0 ], [ 104.1026, 19.51318, 0.0 ], [
49      104.10253, 19.51311, 0.0 ], [ 104.10246, 19.51304, 0.0 ], [ 104.10239, 19.51297, 0.0 ], [ 104.10232, 19.5129, 0.0 ], [
50      104.10225, 19.51283, 0.0 ], [ 104.10218, 19.51276, 0.0 ], [ 104.10211, 19.51269, 0.0 ], [ 104.10204, 19.51262, 0.0 ], [
51      104.10197, 19.51255, 0.0 ], [ 104.1019, 19.51248, 0.0 ], [ 104.10183, 19.51241, 0.0 ], [ 104.10176, 19.51234, 0.0 ], [
52      104.10169, 19.51227, 0.0 ], [ 104.10162, 19.5122, 0.0 ], [ 104.10155, 19.51213, 0.0 ], [ 104.10148, 19.51206, 0.0 ], [
53      104.10141, 19.51199, 0.0 ], [ 104.10134, 19.51192, 0.0 ], [ 104.10127, 19.51185, 0.0 ], [ 104.1012, 19.51178, 0.0 ], [
54      104.10113, 19.51171, 0.0 ], [ 104.10106, 19.51164, 0.0 ], [ 104.10099, 19.51157, 0.0 ], [ 104.10092, 19.5115, 0.0 ], [
55      104.10085, 19.51143, 0.0 ], [ 104.10078, 19.51136, 0.0 ], [ 104.10071, 19.51129, 0.0 ], [ 104.10064, 19.51122, 0.0 ], [
56      104.10057, 19.51115, 0.0 ], [ 104.1005, 19.51108, 0.0 ], [ 104.10043, 19.51101, 0.0 ], [ 104.10036, 19.51094, 0.0 ], [
57      104.10029, 19.51087, 0.0 ], [ 104.10022, 19.5108, 0.0 ], [ 104.10015, 19.51073, 0.0 ], [ 104.10008, 19.51066, 0.0 ], [
58      104.10001, 19.51059, 0.0 ], [ 103.99994, 19.51052, 0.0 ], [ 103.99987, 19.51045, 0.0 ], [ 103.9998, 19.51038, 0.0 ], [
59      103.99973, 19.51031, 0.0 ], [ 103.99966, 19.51024, 0.0 ], [ 103.99959, 19.51017, 0.0 ], [ 103.99952, 19.5101, 0.0 ], [
60      103.99945, 19.51003, 0.0 ], [ 103.99938, 19.51, 0.0 ], [ 103.99931, 19.50993, 0.0 ], [ 103.99924, 19.50986, 0.0 ], [
61      103.99917, 19.50979, 0.0 ], [ 103.9991, 19.50972, 0.0 ], [ 103.99903, 19.50965, 0.0 ], [ 103.99896, 19.50958, 0.0 ], [
62      103.99889, 19.50951, 0.0 ], [ 103.99882, 19.50944, 0.0 ], [ 103.99875, 19.50937, 0.0 ], [ 103.99868, 19.5093, 0.0 ], [
63      103.99861, 19.50923, 0.0 ], [ 103.99854, 19.50916, 0.0 ], [ 103.99847, 19.50909, 0.0 ], [ 103.9984, 19.50902, 0.0 ], [
64      103.99833, 19.50895, 0.0 ], [ 103.99826, 19.50888, 0.0 ], [ 103.99819, 19.50881, 0.0 ], [ 103.99812, 19.50874, 0.0 ], [
65      103.99805, 19.50867, 0.0 ], [ 103.99798, 19.5086, 0.0 ], [ 103.99791, 19.50853, 0.0 ], [ 103.99784, 19.50846, 0.0 ], [
66      103.99777, 19.50839, 0.0 ], [ 103.9977, 19.50832, 0.0 ], [ 103.99763, 19.50825, 0.0 ], [ 103.99756, 19.50818, 0.0 ], [
67      103.99749, 19.50811, 0.0 ], [ 103.99742, 19.50804, 0.0 ], [ 103.99735, 19.50797, 0.0 ], [ 103.99728, 19.5079, 0.0 ], [
68      103.99721, 19.50783, 0.0 ], [ 103.99714, 19.50776, 0.0 ], [ 103.99707, 19.50769, 0.0 ], [ 103.997, 19.50762, 0.0 ], [
69      103.99693, 19.50755, 0.0 ], [ 103.99686, 19.50748, 0.0 ], [ 103.99679, 19.50741, 0.0 ], [ 103.99672, 19.50734, 0.0 ], [
70      103.99665, 19.50727, 0.0 ], [ 103.99658, 19.5072, 0.0 ], [ 103.99651, 19.50713, 0.0 ], [ 103.99644, 19.50706, 0.0 ], [
71      103.99637, 19.50699, 0.0 ], [ 103.9963, 19.50692, 0.0 ], [ 103.99623, 19.50685, 0.0 ], [ 103.99616, 19.50678, 0.0 ], [
72      103.99609, 19.50671, 0.0 ], [ 103.99602, 19.50664, 0.0 ], [ 103.99595, 19.50657, 0.0 ], [ 103.99588, 19.5065, 0.0 ], [
73      103.99581, 19.50643, 0.0 ], [ 103.99574, 19.50636, 0.0 ], [ 103.99567, 19.50629, 0.0 ], [ 103.9956, 19.50622, 0.0 ], [
74      103.99553, 19.50615, 0.0 ], [ 103.99546, 19.50608, 0.0 ], [ 103.99539, 19.50601, 0.0 ], [ 103.99532, 19.50594, 0.0 ], [
75      103.99525, 19.50587, 0.0 ], [ 103.99518, 19.5058, 0.0 ], [ 103.99511, 19.50573, 0.0 ], [ 103.99504, 19.50566, 0.0 ], [
76      103.99497, 19.50559, 0.0 ], [ 103.9949, 19.50552, 0.0 ], [ 103.99483, 19.50545, 0.0 ], [ 103.99476, 19.50538, 0.0 ], [
77      103.99469, 19.50531, 0.0 ], [ 103.99462, 19.50524, 0.0 ], [ 103.99455, 19.50517, 0.0 ], [ 103.99448, 19.5051, 0.0 ], [
78      103.99441, 19.50503, 0.0 ], [ 103.99434, 19.50496, 0.0 ], [ 103.99427, 19.50489, 0.0 ], [ 103.9942, 19.50482, 0.0 ], [
79      103.99413, 19.50475, 0.0 ], [ 103.99406, 19.50468, 0.0 ], [ 103.99399, 19.50461, 0.0 ], [ 103.99392, 19.50454, 0.0 ], [
80      103.99385, 19.50447, 0.0 ], [ 103.99378, 19.5044, 0.0 ], [ 103.99371, 19.50433, 0.0 ], [ 103.99364, 19.50426, 0.0 ], [
81      103.99357, 19.50419, 0.0 ], [ 103.9935, 19.50412, 0.0 ], [ 103.99343, 19.50405, 0.0 ], [ 103.99336, 19.50398, 0.0 ], [
82      103.99329, 19.50391, 0.0 ], [ 103.99322, 19.50384, 0.0 ], [ 103.99315, 19.50377, 0.0 ], [ 103.99308, 19.5037, 0.0 ], [
83      103.99301, 19.50363, 0.0 ], [ 103.99294, 19.50356, 0.0 ], [ 103.99287, 19.50349, 0.0 ], [ 103.9928, 19.50342, 0.0 ], [
84      103.99273, 19.50335, 0.0 ], [ 103.99266, 19.50328, 0.0 ], [ 103.99259, 19.50321, 0.0 ], [ 103.99252, 19.50314, 0.0 ], [
85      103.99245, 19.50307, 0.0 ], [ 103.99238, 19.503, 0.0 ], [ 103.99231, 19.50293, 0.0 ], [ 103.99224, 19.50286, 0.0 ], [
86      103.99217, 19.50279, 0.0 ], [ 103.9921, 19.50272, 0.0 ], [ 103.99203, 19.50265, 0.0 ], [ 103.99196, 19.50258, 0.0 ], [
87      103.99189, 19.50251, 0.0 ], [ 103.99182, 19.50244, 0.0 ], [ 103.99175, 19.50237, 0.0 ], [ 103.99168, 19.5023, 0.0 ], [
88      103.99161, 19.50223, 0.0 ], [ 103.99154, 19.50216, 0.0 ], [ 103.99147, 19.50209, 0.0 ], [ 103.9914, 19.50202, 0.0 ], [
89      103.99133, 19.50195, 0.0 ], [ 103.99126, 19.50188, 0.0 ], [ 103.99119, 19.50181, 0.0 ], [ 103.99112, 19.50174, 0.0 ], [
90      103.99105, 19.50167, 0.0 ], [ 103.99098, 19.5016, 0.0 ], [ 103.99091, 19.50153, 0.0 ], [ 103.99084, 19.50146, 0.0 ], [
91      103.99077, 19.50139, 0.0 ], [ 103.9907, 19.50132, 0.0 ], [ 103.99063, 19.50125, 0.0 ], [ 103.99056, 19.50118, 0.0 ], [
92      103.99049, 19.50111, 0.0 ], [ 103.99042, 19.50104, 0.0 ], [ 103.99035, 19.50097, 0.0 ], [ 103.99028, 19.5009, 0.0 ], [
93      103.99021, 19.50083, 0.0 ], [ 103.99014, 19.50076, 0.0 ], [ 103.99007, 19.50069, 0.0 ], [ 103.99, 19.50062, 0.0 ], [
94      103.98993, 19.50055, 0.0 ], [ 103.98986, 19.50048, 0.0 ], [ 103.98979, 19.50041, 0.0 ], [ 103.98972, 19.50034, 0.0 ], [
95      103.98965, 19.50027, 0.0 ], [ 103.98958, 19.5002, 0.0 ], [ 103.98951, 19.50013, 0.0 ], [ 103.98944, 19.50006, 0.0 ], [
96      103.98937, 19.500, 0.0 ], [ 103.9893, 19.49993, 0.0 ], [ 103.98923, 19.49986, 0.0 ], [ 103.98916, 19.49979, 0.0 ], [
97      103.98909, 19.49972, 0.0 ], [ 103.98902, 19.49965, 0.0 ], [ 103.98895, 19.49958, 0.0 ], [ 103.98888, 19.49951, 0.0 ], [
98      103.98881, 19.49944, 0.0 ], [ 103.98874, 19.49937, 0.0 ], [ 103.98867, 19.4993, 0.0 ], [ 103.9886, 19.49923, 0.0 ], [
99      103.98853, 19.49916, 0.0 ], [ 103.98846, 19.49909, 0.0 ], [ 103.98839, 19.49902, 0.0 ], [ 103.98832, 19.49895, 0.0 ], [
100     103.98825, 19.49888, 0.0 ], [ 103.98818, 19.49881, 0.0 ], [ 103.98811, 19.49874, 0.0 ], [ 103.98804, 19.49867, 0.0 ], [
101     103.98797, 19.4986, 0.0 ], [ 103.9879, 19.49853, 0.0 ], [ 103.98783, 19.49846, 0.0 ], [ 103.98776, 19.49839, 0.0 ], [
102     103.98769, 19.49832, 0.0 ], [ 103.98762, 19.49825, 0.0 ], [ 103.98755, 19.49818, 0.0 ], [ 103.98748, 19.49811, 0.0 ], [
103     103.98741, 19.49804, 0.0 ], [ 103.98734, 19.49797, 0.0 ], [ 103.98727, 19.4979, 0.0 ], [ 103.9872, 19.49783, 0.0 ], [
104     103.98713, 19.49776, 0.0 ], [ 103.98706, 19.49769, 0.0 ], [ 103.98699, 19.49762, 0.0 ], [ 103.98692, 19.49755, 0.0 ], [
105     103.98685, 19.49748, 0.0 ], [ 103.98678, 19.49741, 0.0 ], [ 103.98671, 19.49734, 0.0 ], [ 103.98664, 19.49727, 0.0 ], [
106     103.98657, 19.4972, 0.0 ], [ 103.9865, 19.49713, 0.0 ], [ 103.98643, 19.49706, 0.0 ], [ 103.98636, 19.49699, 0.0 ], [
107     103.98629, 19.49692, 0.0 ], [ 103.98622, 19.49685, 0.0 ], [ 103.98615, 19.49678, 0.0 ], [ 103.98608, 19.49671, 0.0 ], [
108     103.98601, 19.49664, 0.0 ], [ 103.98594, 19.49657, 0.0 ], [ 103.98587, 19.4965, 0.0 ], [ 103.9858, 19.49643, 0.0 ], [
109     103.98573, 19.49636, 0.0 ], [ 103.98566, 19.49629, 0.0 ], [ 103.98559, 19.49622, 0.0 ], [ 103.98552, 19.49615, 0.0 ], [
110     103.98545, 19.49608, 0.0 ], [ 103.98538, 19.49601, 0.0 ], [ 103.98531, 19.49594, 0.0 ], [ 103.98524, 19.49587, 0.0 ], [
111     103.98517, 19.4958, 0.0 ], [ 103.9851, 19.49573, 0.0 ], [ 103.98503, 19.49566, 0.0 ], [ 103.98496, 19.49559, 0.0 ], [
112     103.98489, 19.49552, 0.0 ], [ 103.98482, 19.49545, 0.0 ], [ 103.98475, 19.49538, 0.0 ], [ 103.98468, 19.49531, 0.0 ], [
113     103.98461, 19.49524, 0.0 ], [ 103.98454, 19.49517, 0.0 ], [ 103.98447, 19.4951, 0.0 ], [ 103.9844, 19.49503, 0.0 ], [
114     103.98433, 19.49496, 0.0 ], [ 103.98426, 19.49489, 0.0 ], [ 103.98419, 19.49482, 0.0 ], [ 103.98412, 19.49475, 0.0 ], [
115     103.98405, 19.49468, 0.0 ], [ 103.98398, 19.49461, 0.0 ], [ 103.98391, 19.49454, 0.0 ], [ 103.98384, 19.49447, 0.0 ], [
116     103.98377, 19.4944, 0.0 ], [ 103.9837, 19.49433, 0.0 ], [ 103.98363, 19.49426, 0.0 ], [ 103.98356, 19.49419, 0.0 ], [
117     103.98349, 19.49412, 0.0 ], [ 103.98342, 19.49405, 0.0 ], [ 103.98335, 19.49398, 0.0 ], [ 103.9
```

KẾT LUẬN VÀ KIẾN NGHỊ

Tầm quan trọng của mô hình đã được lĩnh hội một cách thấu đáo trong hầu như tất cả các ngành khoa học kỹ thuật từ nhiều thế kỷ nay. Bất kỳ ở đâu, khi muốn xây dựng một vật thể nào đó, đầu tiên người ta đã tạo nên các bản vẽ để quyết định cả ngoại hình lẫn phương thức hoạt động của nó. Chẳng hạn các bản vẽ kỹ thuật thường gặp là một dạng mô hình quen thuộc. Mô hình nhìn chung là một cách mô tả của một vật thể nào đó. Vật đó có thể tồn tại trong một số giai đoạn nhất định, dù đó là giai đoạn thiết kế hay giai đoạn xây dựng hoặc chỉ là một kế hoạch. Nhà thiết kế cần phải tạo ra các mô hình mô tả tất cả các khía cạnh khác nhau của sản phẩm. Ngoài ra, một mô hình có thể được chia thành nhiều hướng nhìn, mỗi hướng nhìn trong số chúng sẽ mô tả một khía cạnh riêng biệt của sản phẩm hay hệ thống cần được xây dựng. Một mô hình cũng có thể được xây dựng trong nhiều giai đoạn và ở mỗi giai đoạn, mô hình sẽ được bổ sung thêm một số chi tiết nhất định.

Mô hình thường được mô tả trong ngôn ngữ trực quan, điều đó có nghĩa là đa phần các thông tin được thể hiện bằng các ký hiệu đồ họa và các kết nối giữa chúng, chỉ khi cần thiết một số thông tin mới được biểu diễn ở dạng văn bản; Theo đúng như câu ngạn ngữ "Một bức tranh nói nhiều hơn cả ngàn từ". Tạo mô hình cho các hệ thống phần mềm trước khi thực sự xây dựng nên chúng, đã trở thành một chuẩn mực trong việc phát triển phần mềm và được chấp nhận trong cộng đồng làm phần mềm giống như trong bất kỳ một ngành khoa học kỹ thuật nào khác.

Phương pháp và các giải pháp để hỗ trợ phát triển hệ thống thông tin nhằm cho người sử dụng biết phải tương tác với hệ thống như thế nào. Trong một hệ thống thì đều bao gồm các chức năng khác nhau, trong nghiên cứu này nhóm thực hiện đề tài đã phát triển nhiều chức năng khác nhau như chức năng xuất nhập dữ liệu, các chức năng liên quan đến bản đồ. Nhiệm vụ chính của chức năng xuất nhập dữ liệu là chuyển dữ liệu ảnh vệ tinh thành dạng bảng grid (dạng vector) để

chuyên vào lưu trữ trong cơ sở dữ liệu. Phân khoảng giá trị độ xám và gán giá trị tương ứng.

Phương pháp để xây dựng tầng ứng dụng là nhằm đưa ra giải pháp tối ưu nhất để hiển thị bản đồ ô nhiễm không khí trên nền web và thực hiện các thao tác phân tích, phối màu, trình bày bản đồ ô nhiễm không khí. Người dùng bình thường có thể truy cập vào trang web và xem các thông tin về tình trạng ô nhiễm không khí trên bản đồ nhưng không được thao tác với dữ liệu gốc.

- Hệ điều hành sử dụng:

- Hệ điều trung tâm: Sử dụng hệ điều hành Microsoft Windows server 2000 trở lên (Khuyến dùng windows 2008)
- Hệ điều hành máy trạm: Sử dụng hệ điều hành Microsoft Windows XP Professional (Service Pack 2 trở lên) , Vista 32, Windows 7, Windows 2000, Windows server 2000 trở lên.
- Công cụ duyệt Web như Firefox, Googel Chrome, IE7 trở lên, Safari trên iOS ..

- Hệ quản trị cơ sở dữ liệu:

- PostgreSQL

- Môi trường nền:

- Oracle ADF
- Weblogic server 11.1.2.3

- Môi trường phát triển ứng dụng:

- Sử dụng Middleware Jdeveloper 11.1.2.3 của Oracle
- Sử dụng iReport cho các mẫu biểu và báo cáo thống kê.

- Công nghệ GIS sử dụng trong hệ thống:

- PostGIS
- Geoserver

TÀI LIỆU THAM KHẢO

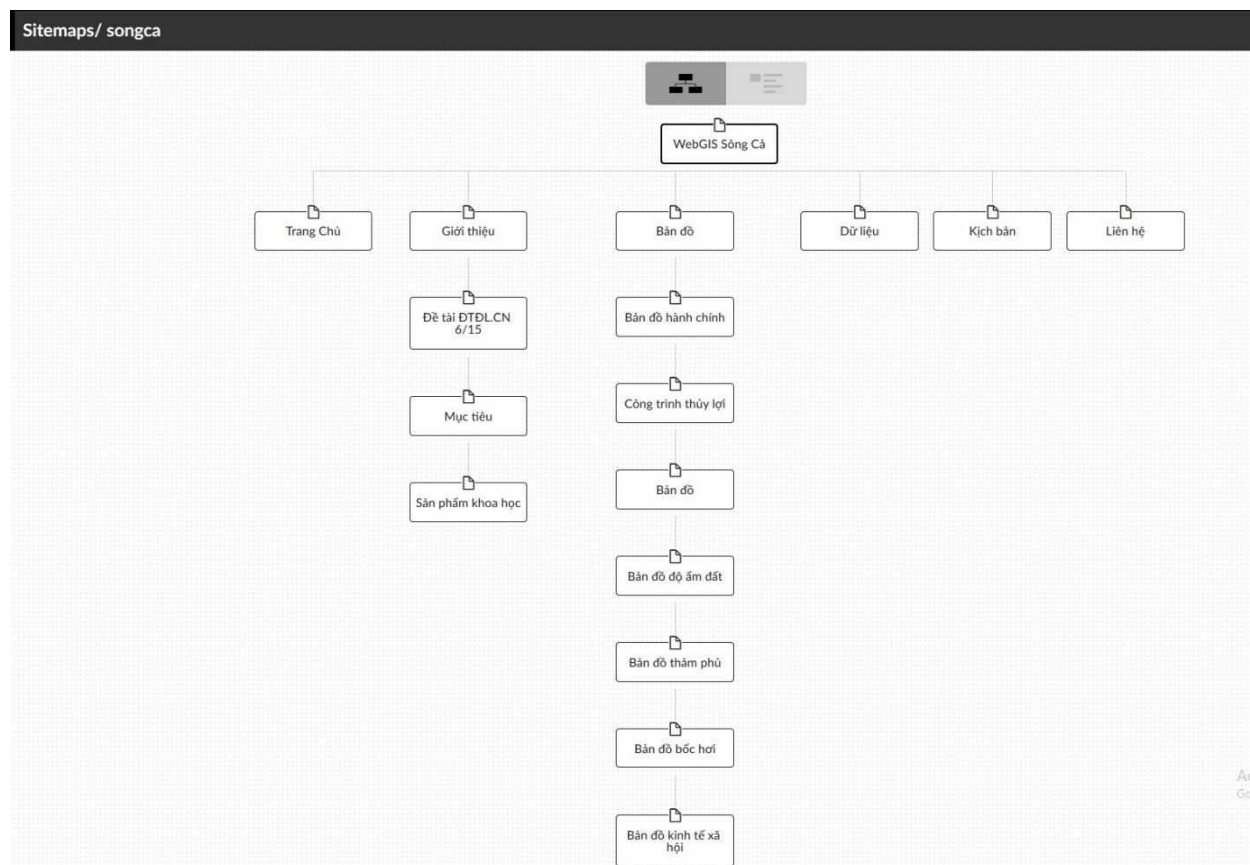
1. Các tiêu chí đánh giá chất lượng sản phẩm phần mềm của các tổ chức quốc tế
http://mic.gov.vn/Upload/Store/tintuc/vietnam/3/TMTC_2010_a2_7.doc
2. Driver, M. "Oracle Application Development Framework: Past, Present and Future." (2012).
3. Dyché, Jill (2000), *Turning Data Into Information With Data Warehousing*, Addison Wesley, [ISBN 978-0-201-65780-7](https://www.amazon.com/dp/0201657807)
4. Feldman, S 2010. [History of Web Mapping - slide deck](#) and [History of Web Mapping - mind map](#)
5. Fu, P., and J. Sun. 2010. *Web GIS: Principles and Applications*. ESRI Press. Redlands, CA. [ISBN 1-58948-245-X](https://www.amazon.com/dp/158948245X).
6. Hệ quản trị cơ sở dữ liệu
http://vi.wikipedia.org/wiki/Hệ_quản_trị_cơ_sở_dữ_liệu
7. ISO/IEC/IEEE 29119 Software Testing
<http://www.softwaretestingstandard.org/>
8. Kraak, Menno-Jan and Allan Brown (2001): *Web Cartography – Developments and prospects*, Taylor & Francis, New York, [ISBN 0-7484-0869-X](https://www.amazon.com/dp/074840869X).
9. Martyn A Ould & Charles Unwin (ed), *Testing in Software Development*, BCS (1986), p71
10. Mitchell, Tyler (2005): *Web Mapping Illustrated*, O'Reilly, Sebastopol, 350 pages, [ISBN 0-596-00865-1](https://www.amazon.com/dp/0596008651). This book discusses various Open Source Web Mapping projects and provides hints and tricks as well as examples.
11. Mitchell, Tyler. *Web mapping illustrated: using open source GIS toolkits*. " O'Reilly Media, Inc.", 2005.
12. Mysql documentation <https://dev.mysql.com/doc/>
13. OpenLayers documentation <http://docs.openlayers.org/>

14. Peterson, Michael P. (ed.) (2003): *Maps and the Internet*, Elsevier, [ISBN 0-08-044201-3](#).
15. PostgreSQL documentation
<https://www.postgresql.org/files/documentation/pdf/9.6/postgresql-9.6-A4.pdf>
16. Reynolds, George (2009), *Ethics in Information Technology*, Cengage Learning, [ISBN 978-0-538-74622-9](#)
17. Satish Mishra (1997). ["Visual Modeling & Unified Modeling Language \(UML\):](#)
18. Zeiler, Michael (1999). *Modeling Our World: The Esri Guide to Geodatabase Design*. Esri. p. 4.

PHỤ LỤC

Hướng dẫn sử dụng công cụ trên WebGIS quản lý và sử dụng nguồn nước sông Cả

II. SITEMAP



Hình 1: Sơ đồ "Mục Lục" của webgis sông Cả

III. Menu chính



Hình 2: Thanh menu chính của webGIS

III.1. Trang chủ

Đây là trang hiển thị tổng quan về khu vực nghiên cứu: lưu vực sông Cả. Sông Cả là một con sông liên quốc gia, bắt nguồn từ dãy núi cao thuộc tỉnh Xiêng Khoảng

Nghiên cứu đề xuất phương án phối hợp vận hành điều tiết nước hợp lý các hồ chứa thủy lợi, thủy điện trên sông Cả phục vụ cấp nước và phòng, chống lũ cho hạ du

bên Lào có độ cao đỉnh núi 2.000 m, chảy theo hướng Tây Bắc - Đông Nam vào Việt Nam, đổ ra biển tại cửa Hội.

Dòng chính sông Cả dài khoảng 514 km, trong đó phần chảy trên lãnh thổ Việt Nam dài 360 km. Phần thượng lưu lòng sông cắt sâu vào địa hình tạo thành lòng hẹp, sâu, do sông dốc nên ít bãi bồi. Phần hạ lưu sông mở rộng có nhiều bồn trũng như Hữu Thanh Chương, vùng Nam Đàn - Đức Thọ. Tổng diện tích lưu vực 27.200 km² trong đó diện tích thuộc lãnh thổ Việt Nam là 20.136 km². Giao diện trang chủ thể hiện vị trí của lưu vực sông Cả ở dạng dữ liệu vệ tinh.



Hình 3: Giao diện trang chủ của webGIS

III.2. Giới thiệu

Menu này cung cấp thông tin liên quan đến đề tài, Dự án, các mục tiêu thực hiện, nội dung và sản phẩm khoa học của dự án.

Nghiên cứu đề xuất phương án phối hợp vận hành điều tiết nước hợp lý các hồ chứa thủy lợi, thủy điện trên sông Cả phục vụ cấp nước và phòng, chống lũ cho hạ du



Hình 4: Giao diện trang giới thiệu của webgis

III.3. Bản đồ

Là trang quan trọng nhất của website, thể hiện các kết quả về bản đồ của dự án bằng cách liệt kê danh mục các bản đồ kết quả. Các bản đồ được chia thành các nhóm sau:

- Bản đồ hành chính
- Công trình thủy lợi
- Bản đồ 3D
- Bản đồ độ ẩm đất
- Bản đồ tham phủ
- Bản đồ mưa
- Bản đồ bốc hơi
- Bản đồ kinh tế xã hội

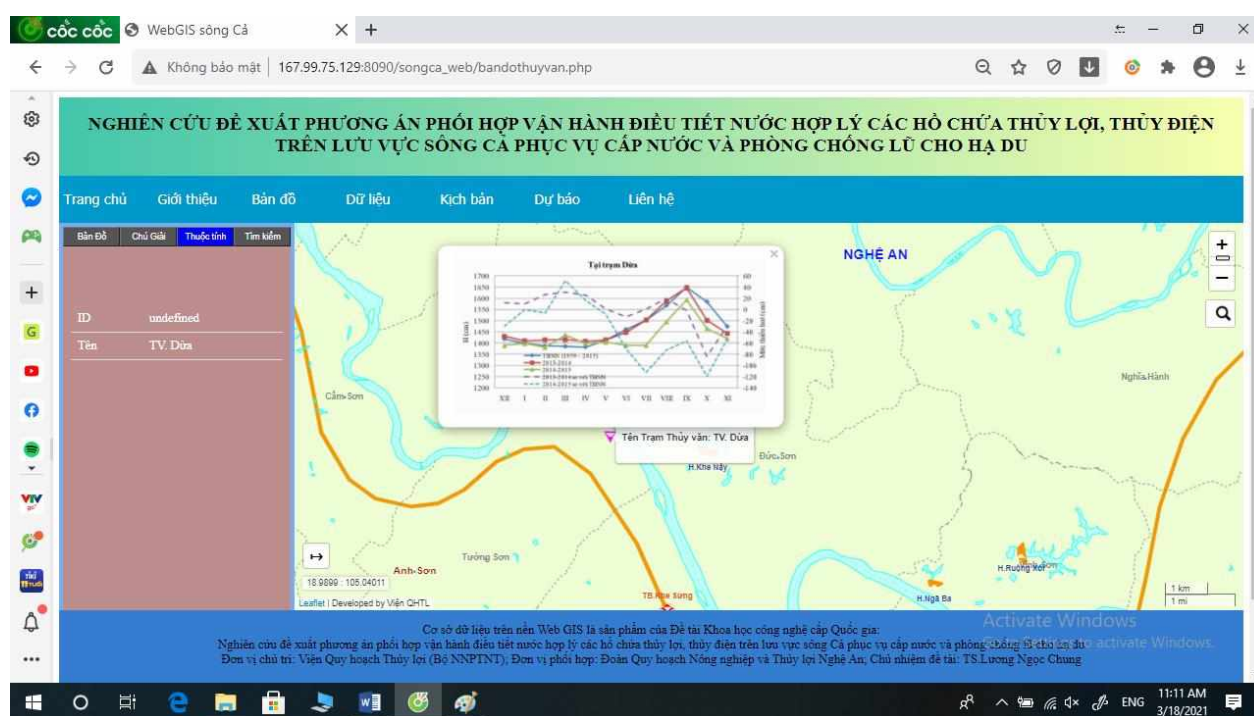
1. Công cụ tương tác với bản đồ

Nền chính của menu này là bản đồ và các công cụ tương tác với bản đồ. Bản đồ nền là bản đồ thể hiện các đối tượng nền (sông suối, ranh giới hành chính, giao thông) lấy nguồn từ Mapbox.

2. Truy vấn thông tin

Có 2 cách truy vấn thông tin

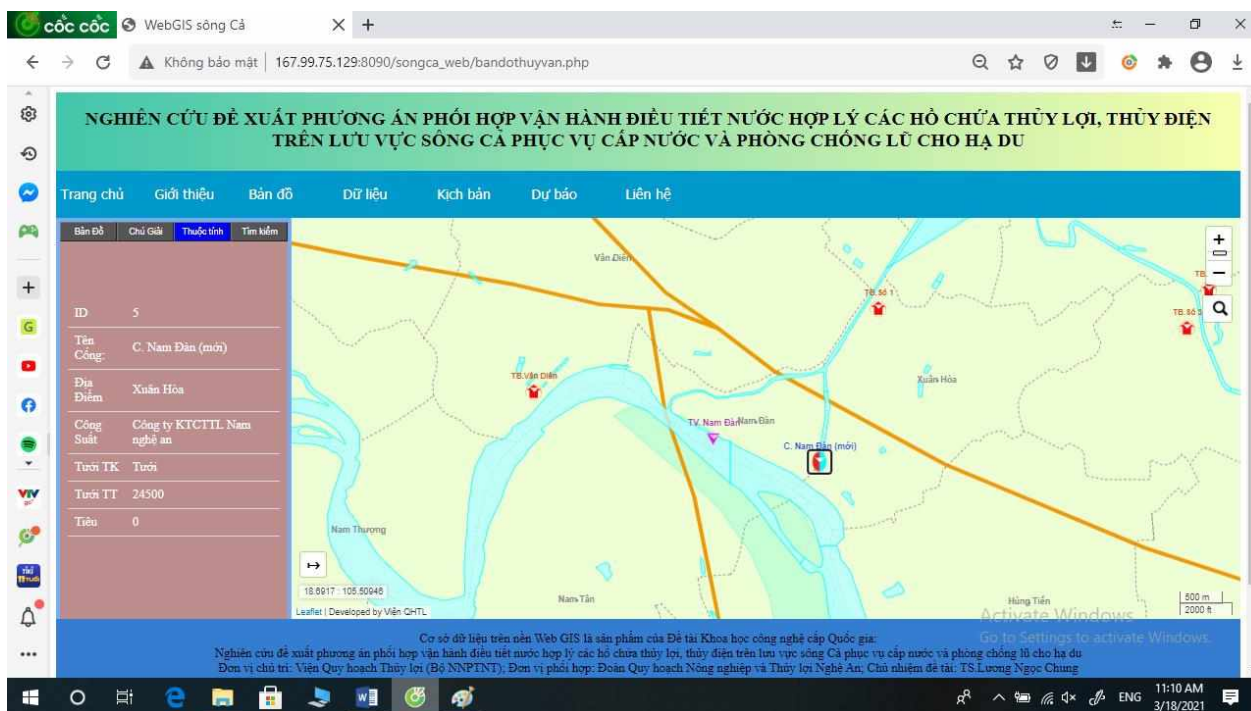
Cách 1: di chuyển con trỏ chuột qua đối tượng cần truy vấn thông tin. Các thông tin hiển thị về đối tượng sẽ được hiển thị dưới dạng tooltip (1 cửa sổ hình chữ nhật có chứa thông tin về đối tượng)



Hình 5: Truy vấn thông tin qua tooltip

Cách 2: click chuột vào đối tượng. Các thông tin đầy đủ về đối tượng

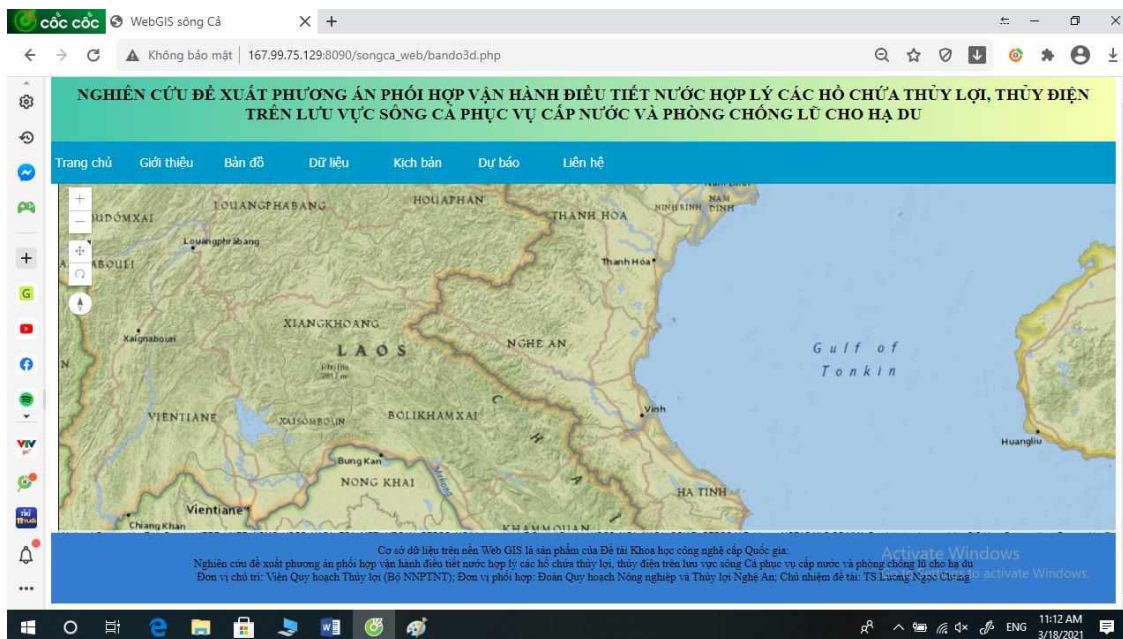
Nghiên cứu đề xuất phương án phối hợp vận hành điều tiết nước hợp lý các hồ chứa thủy lợi, thủy điện trên lưu vực sông Cả phục vụ cấp nước và phòng, chống lũ cho hạ du



Hình 6: Truy vấn thông tin qua bảng thuộc tính


3. Bản đồ 3D

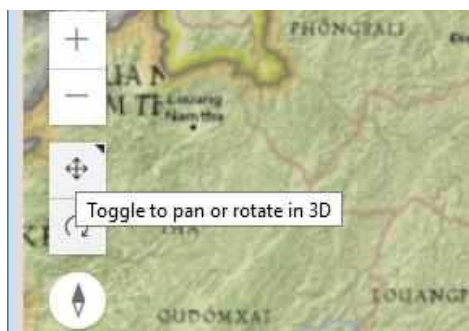
Bản đồ 3D sử dụng thư viện dữ liệu của ESRI. Trang “Bản đồ 3D” cho phép người dùng quan sát khu vực nghiên cứu dưới dạng 3D, để có 1 hình dung cụ thể hơn về độ cao.



Hình 7: Giao diện trang Bản đồ 3D

Nghiên cứu đề xuất phương án phối hợp vận hành điều tiết nước hợp lý các hồ chứa thủy lợi, thủy điện trên sông Cả phục vụ cấp nước và phòng, chống lũ cho hạ du

Giao diện của trang Bản đồ 3D ban đầu sẽ có dạng 2D. Để thao tác với chế độ bản đồ 3D, người dùng chọn biểu tượng  để kích hoạt chế độ xem 3D



Hình 8: Công cụ để chuyển sang chế độ xem 3D

Sau khi kích hoạt chế độ xem 3D, kết quả thu được sẽ giống như ở hình dưới



Độ cao và địa danh của một số vị trí sẽ được hiển thị.

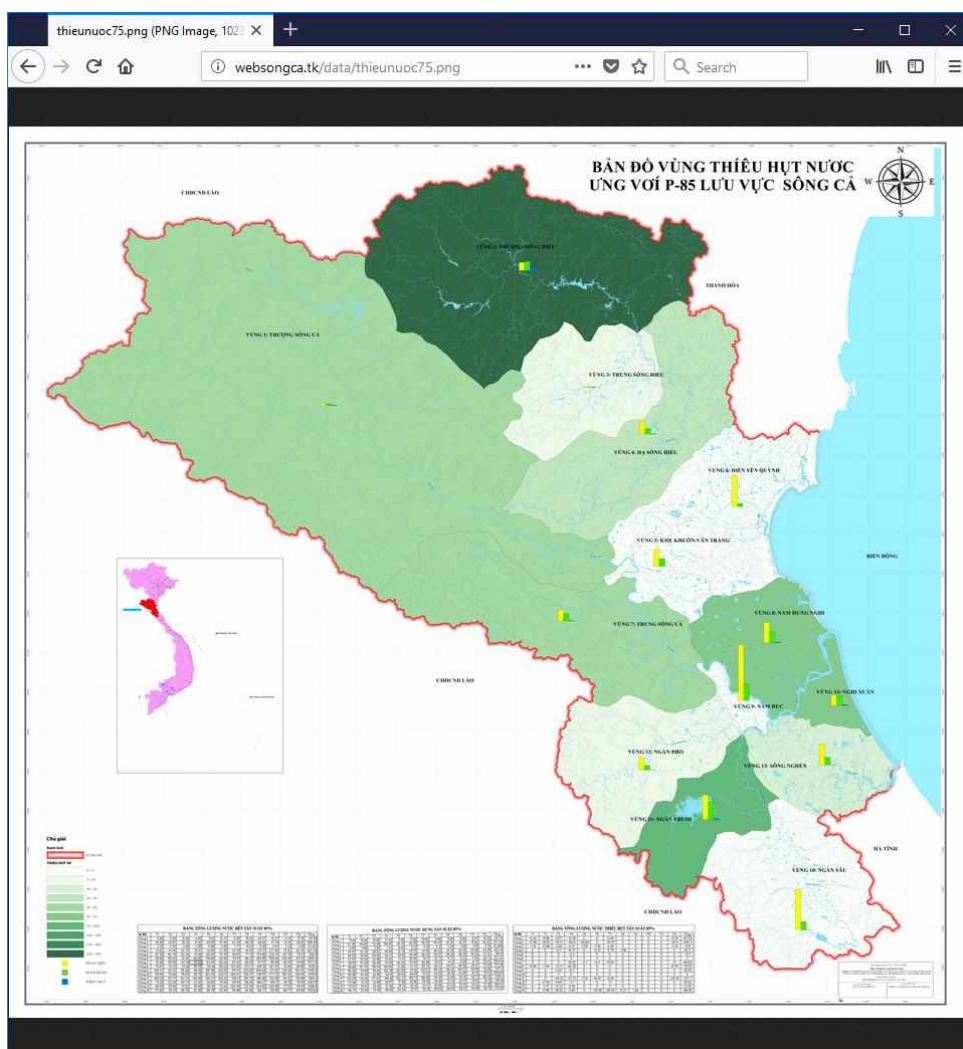
Để định hướng lại bản đồ (hướng Bắc hướng lên trên), người dùng ấn vào biểu tượng hình nam châm



Hình 9: Công cụ thể định hướng lại Bản đồ

IV. Kịch bản

Trang này thể hiện các kịch bản thiếu hụt nước ở Lưu vực sông Cả ứng với 2 trường hợp là P-75 và P-85.



Hình 10: Giao diện trang "Kịch bản"

VI. Liên hệ: Trang này chủ yếu cung cấp thông tin về bản quyền của sản phẩm