

Web based GIS Package for Small Area Using Open Source Software

Roheet Bhatnagar, and Gagan Bansal

Abstract—Geographic Information System is a collection of hardware, software and data which is used for capturing and managing geographically referenced information. It allows users to view and interpret data in forms of maps, reports and charts etc. Open Source refers to any program whose source code is made available for use or modification as users or developers see fit. It is usually developed as a public collaboration and made freely available. The main purpose of the project is to develop a web based GIS package for a small area using Open Source softwares, which can be used as a service to store, add, edit or delete Road features overlaid over a base map. The web service created will provide a UI design where users can place mark any road feature on the Google map and add information to that feature like Name, Width, and Type. In addition to this a user can also edit and delete previously existing features on the map. Client-Server architecture is used in this application where Open Layers (A Java Script library for displaying map data) is used on the client side and Python and CherryPy on the server side. Apache has been used as the web server with Google maps as the base layer and ZODB as the database.

Keywords—GIS, Open Source, Web Service, Map Features.

I. INTRODUCTION

THE application of Geographical Information System (GIS) has been widely demonstrated in varied areas over the years. GIS is a tool which helps users to view and interpret data in form of maps, reports and charts etc. [1] With the advent of new technologies, Commerce and Industry discovered the huge potential of GIS, and started using it as a great application tool. [2] The Geographical Information Systems (GIS) has been developed for decades and commercial software packages have been successfully developed, for instance, ArcView, ArcGIS and MapInfo etc. In the last few years, there has been a significant development in the area of free and open source geospatial software e.g. OpenStreetMap, Map Server, Open Layers etc. The purpose of this work is to create a web based GIS package for small area using Open Source Software, which can be used as a web service to add, store, edit and update any feature layer (like road, landuse, administrative boundary, drainage etc) over a map and that can serve for small city, campus, locality or

village. We have focused on the road layer. A user can tag any road (add Name, Type and Width of the road) over the map. The user can also edit and delete the road features on the map. A person who is new to the locality can benefit greatly with this application as he will get information regarding different locations in the campus with one look at the map. To achieve this task client-server architecture [3] was employed where JavaScript and HTML were used for client side coding and python with cherrypy for the server side. Open Layers [4] is an Open Source map viewing framework written in JavaScript for displaying map data in most modern web browsers and implements a JavaScript API for building rich web-based geographic applications, similar to the Google Maps and MSN Web based GIS package for small area Using Open Source Software Virtual Earth APIs, with one important difference - OpenLayers is Open Source Software. Python is object oriented programming language and cherrypy and routes are used here to create web service. We have used Apache as the web server and ZODB as database. ZODB is an object oriented database for storing python objects. As a reference the campus GIS of Oregon University, USA was studied. The interactive map of University of Oregon shows a campus map with various layers like Level, construction, bus Stop, Road, Parking etc. The users are allowed to select from the options of various layers. Once the layers are selected, that layer of the map is highlighted. The users can use the navigation tools to browse the map, and can also use the search box to find information and locations of key features, buildings, and resources.

A. Geographic Information System

Geographical Information System or Geo Informatics System can be defined as a collection of hardware, software and data which is used for capturing and managing of geographically referenced information. GIS allows us to view and interpret data in many ways in forms of maps, reports and charts etc. [1] One can answer questions and solve problems by looking at the data in a way that is quickly understood and easily shared with the help of GIS. In a general sense, the term describes any information system that integrates, stores analyzes geographic information for informing decision making. The application of GIS has been widely demonstrated in varied areas over the years. Initially, such application was limited to the realm of natural resources where most of the input sources predominantly consisted of spatial data such as satellite images, aerial photographs and conventional paper

Roheet Bhatnagar is working as Professor & Head, Department of Computer Science & Engineering in Manipal University Jaipur, Rajasthan (+91 8003897115; e-mail: roheet.bhatnagar@jaipur.manipal.edu).

Gagan Bansal is working as an Architect, GIS division, Rediff, Mumbai, India (e-mail: gagan_bansal@yahoo.co.in).

maps. In course of time, with the advent of new technologies, commerce and industry discovered the huge potential of GIS, and started using it as a great application tool. [2] Recent studies concerning GIS show that it is the fastest growing segment (both hardware & software) of the graphical computer market.

B. Web Services

A web service can be defined as a method of communication between two devices over a network. It is a software system which supports two interoperable machines to interact over network. Web services are published, found and used through web. Web applications are simple applications that run on the web. These are built around the web browser standards and can be used by any browser or any platform. By using web services our application can publish its function or message to the rest of the world. With Web services we can exchange data between different applications and different platforms.

Web Services offer many advantages over other types of distributed computing architectures. [5]

- Interoperability - Web Services typically work outside of private networks, offering developers a non-proprietary route to their solutions. Moreover, Web Services are virtually platform-independent. Web based GIS package for small area Using Open Source Softwares
- Usability - This gives your applications the freedom to choose the Web Services that they need. This allows us to develop services and/or client-side code using the languages and tools that we want.
- Reusability - This makes it easy to reuse Web Service components as appropriate in other services. It also makes it easy to deploy legacy code as a Web Service.

Disadvantages of Web service:-

- Web services use plain text protocols. This means that Web service requests are larger than requests encoded with a binary protocol. The extra size is really only an issue over low-speed connections, or over extremely busy connections.
- Although HTTP and HTTPS (the core Web protocols) are simple, they weren't really meant for long-term sessions. Typically, a browser makes an HTTP connection, requests a Web page and maybe some images, and then disconnects. The interaction is difficult with Web services.
- The problem with HTTP and HTTPS when it comes to Web services is that these protocols are "stateless"—the interaction between the server and client is typically brief and when there is no data being exchanged, the server and client have no knowledge of each other. More specifically, if a client makes a request to the server, receives some information, and then immediately crashes due to a power outage, the server never knows that the client

is no longer active. The server needs a way to keep track of what a client is doing and also to determine when a client is no longer active.

C. Open Source Technology

In general, open source refers to any program whose source code is made available for use or modification as users or other developers see fit. Open source software is usually developed as a public collaboration and made freely available. Open source gained hold with the rise of the Internet, and the attendant need for massive retooling of the computing source code. Opening the source code enables a self-enhancing diversity of production models, communication paths, and interactive communities which is very helpful in the easy and fast development. Developing software using an open-source model is becoming increasingly popular. The main advantages that open-source software has over proprietary (closed-source) software are:

- Lower development cost. Individuals and smaller companies may aid in developing the software, reducing the number of programmers you have to pay yourself.
- Bug detection and correction. With more people viewing your code from more diverse backgrounds, bugs will be found and fixed faster. This is often referred to as "Linus' Law": "Given enough eyeballs, all bugs are shallow."
- Reuse of code and shorter development time. Code from an older or different project can be incorporated into your new project, reducing the time it takes to create and develop it.
- More independence - with a closed source project, you will have great problems Web based GIS package for small area Using Open Source Softwares if the developer goes out of business and you need your software updated or fixed. With the code open, any company can step in to fill the void. There may be some disadvantages to open-source software from a development perspective, though rarely any from the end-user perspective.

The main problems are:

- Disclosure of trade secrets. Having the code for a program available implies revealing how it works. This includes disclosure of algorithms and how a device with a unique design might function. Revealing this information to others may cause duplication and loss of financial advantage.
- Loss of revenue through traditional sales. If the source is available, then it is unlikely that consumers will pay a large amount for a CD or license. Revenue must instead be garnered through support agreements.

D. Open Layers

Open Layers [6] is a pure JavaScript library for displaying map data in most modern web browsers, with no server-side dependencies. Open Layers implements a JavaScript API for building rich web-based geographic applications, similar to the Google Maps and MSN Virtual Earth APIs, with one

important difference - Open Layers is Open Source Software. Furthermore, Open Layers implements industry-standard methods for geographic data access, such as the OpenGIS Consortium's Web Mapping Service (WMS) and Web Feature Service (WFS) protocols. Open Layers is written in object oriented JavaScript. Open Layers is intended to separate map tools from map data so that all the tools can operate on all the data sources. Open Layers makes it easy to put a dynamic map in any web page. It can display map tiles and markers loaded from any source. Open Layers is developed and supported by a number of organizations around the world. Open Layers also has capability for vector data editing at client side for all geometry types.

E. Apache

Apache [7] is web server software notable for playing a key role in the initial growth of the World Wide Web. In 2009 it became the first web server software to surpass the 100 million web site milestone. It has evolved as a rival to other web servers in terms of functionality and performance. Apache supports a variety of features, many implemented as compiled modules which extend the core functionality. These can range from server side programming language support to authentication schemes. Some common language interfaces support Perl, Python and PHP. Apache features configurable error messages, DBMS-based authentication databases, and content negotiation. It is also supported by several graphical user interfaces (GUIs). Although the main design goal of Apache is not to be the "fastest" web server, Apache does have performance comparable to other "high-performance" web servers. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation.

F. Python

Python is an object-oriented high-level programming language with dynamic semantics that lets us work more quickly and integrate our systems more effectively. Its high-level built in data structures, combined with dynamic typing and dynamic binding, making it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. It helps in immediate gains in productivity and lower maintenance costs. Python is a remarkably powerful dynamic programming language that is used in a wide variety of application domains. Python runs on Windows, Linux/Unix, Mac OS X, and has been ported to the Java and .NET virtual machines. Python is free to use, even for commercial products, because of its OSI-approved open source license.

II. METHODOLOGY

A. Scope of Research

The scope of our research aimed at developing a web based GIS package for a small city, campus, locality or village, which can be used as a web service to add, store, edit and update the features over a map.

The GIS package developed should be capable of serving the following purposes:-

- Creation of new features like Roads over the map, and setting its attributes.
- Retrieval of information from Road layer e.g. road data from 'road layer'.
- Edit the Geometry and attributes of a feature.
- Use customized queries for locating a specific feature.
- Posting the modifications in feature data from client to server.
- Updating the existing information in the DB.

In this application, a user will be initially provided with a map of a small region with Navigation tools shown over the map. The user will be provided with different buttons named Navigation, New, Edit, Delete, Query and Save just above the map. Different buttons will be having different strategies associated to them. The users can use the navigation tools to browse the map. Once the user selects the New button, the draw control will be activated and user can draw roads over the map. Just after a user finishes drawing the road, a Sidebar will open at the right side of the map with a form for entering the attributes of the feature drawn. After the user clicks on the Submit button, the attributes values will be set for that feature. When the user clicks on Edit button, select control will be activated i.e. user can select a feature that he/she wants to edit. On the selection of a feature on the map, the attribute form showing attribute values will open at the right side of map. The user can edit the Geometry as well as attribute values of the selected feature. When the user clicks on the Delete button and selects the feature to be deleted, the feature will be deleted from the map. The user can also locate feature by framing customized queries with the help of Query button provide in the interface. After making any changes to the geometry and/or attribute values of a feature, user can use the Save button to save the changes made. Also after deleting a feature from the map, user should use the Save buttons to permanently delete the feature data from DB.

The GIS package thus developed can be used as a web service to add, store, edit, delete and update the different layers over a base map, and its feature attributes (e.g. Road layer in the present case). OpenLayers is used as the framework for viewing the Open Source map. The application developed is based on Client-Server architecture. The client side coding was done using JavaScript, HTML, and OpenLayers library. The server side coding was done using Python with CherryPy. Apache is used as the web-server, with ZODB serving as the database for storing the features on the map. The GUI for the application has been kept very simple, interactive and user friendly so that it can cater to the needs of both novice and the expert users of Geo Informatics System.

B. Implementation Goals

The GIS package is designed and developed in such a way that it can be used as a simple web service for storing and

managing the features over a map. For the initial version, Google maps are used as a base layer. The revisions to the application would include more base layers like OSM and Bing, and they can be integrated together to provide a better interface. Currently only one layer, Road layer is overlaid over the map; later versions would include more layers like Building layer, Water-bodies layer, Boundary layer etc. The subsequent revisions of the application would incorporate separate user and admin modules so that users can only post the changes in features to administrator. The administrator in turn, after verification, should post the request to server, to make suitable changes in the DB.

C. Proposed solution architecture

For the problem stated above, Open Layers was used as a framework for viewing the maps and rendering the features like Roads over it. GET method is used by the client for retrieving data from the server. POST method is used for posting data from client to server. Since Open Layers is a pure JavaScript Library which helps us to store and manage spatial data, the client side coding for creating GUI was done with the help of JavaScript and HTML. Google maps were used as the base layer and Road layer was added to the map with the help of vector-editing for managing the features and their attributes. Apache was used as the web server for rendering the maps and features in the browser. Server-side coding was done in Python with CherryPy. [8] For storing the features and the associated data, ZODB (Zope Object Data Base) was used as the database.

The architecture of the system has been shown in Fig 1.

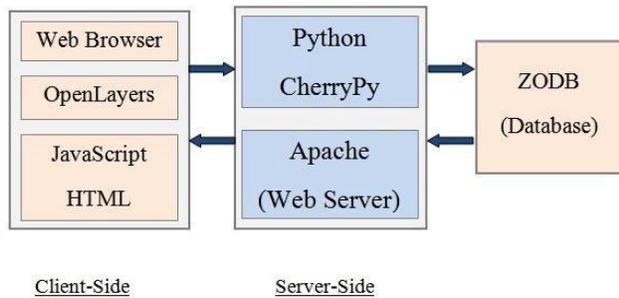


Fig. 1 Overall System Architecture

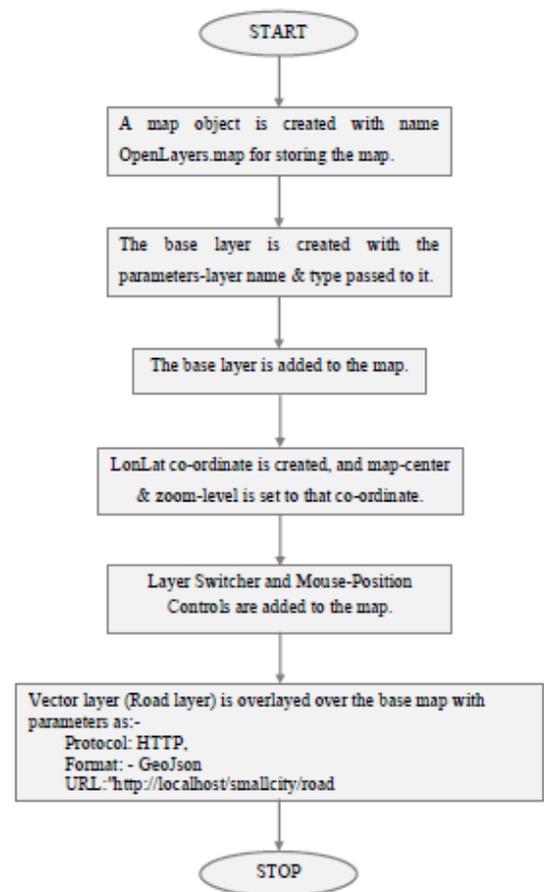


Fig 2 Digitisation process for creating the base layer

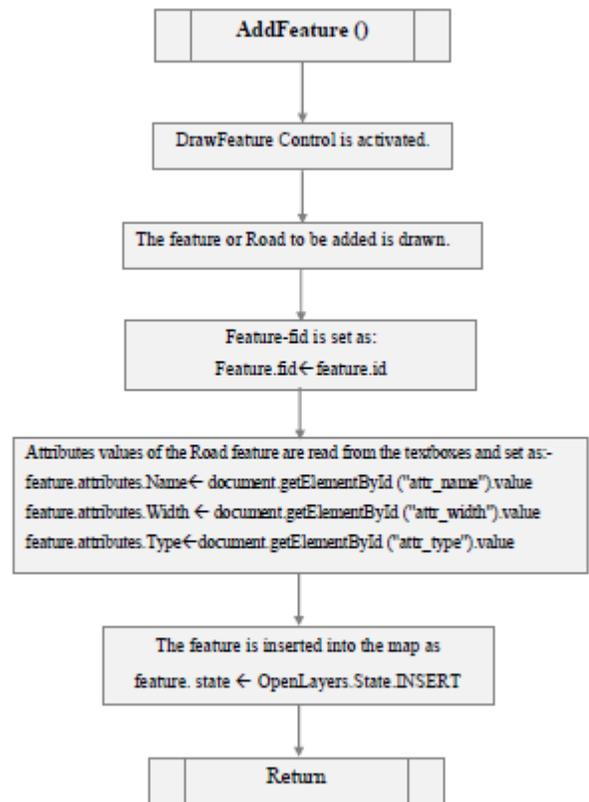


Fig 3 Adding new feature to the base map

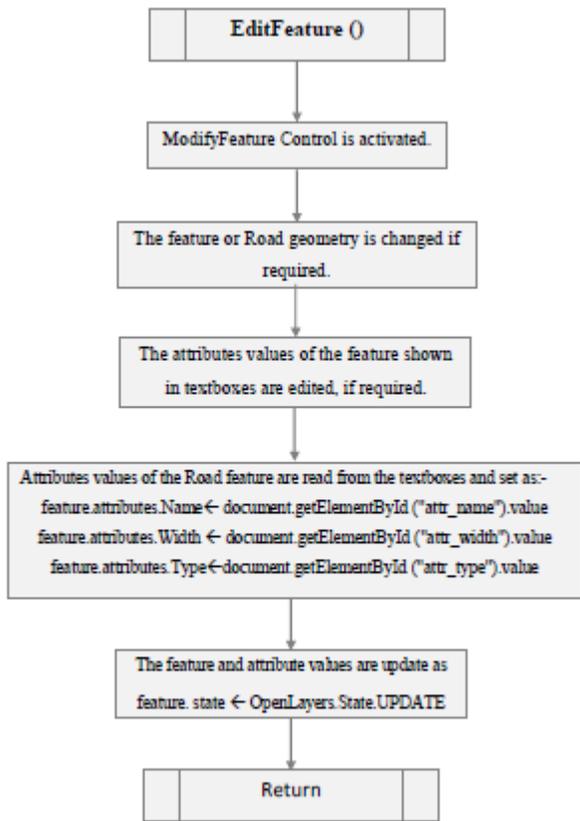


Fig 4 Editing the feature in the base map

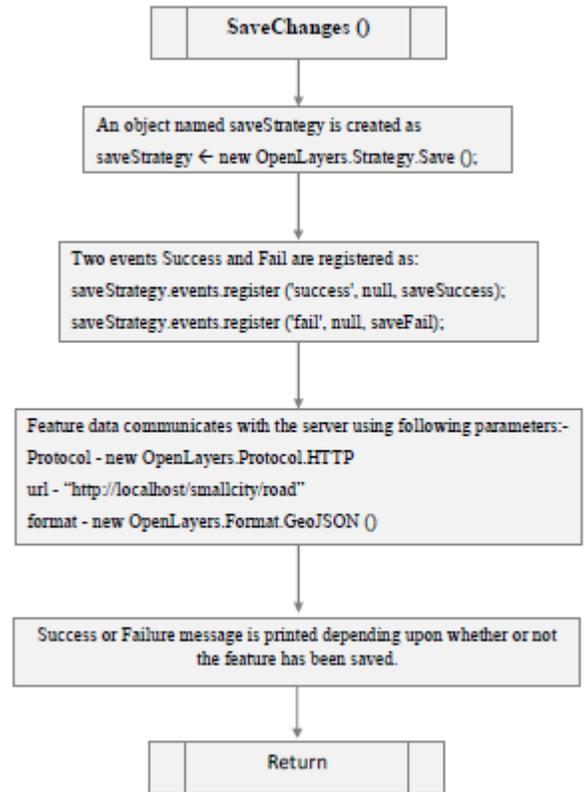


Fig 5 Saving the feature in the base map

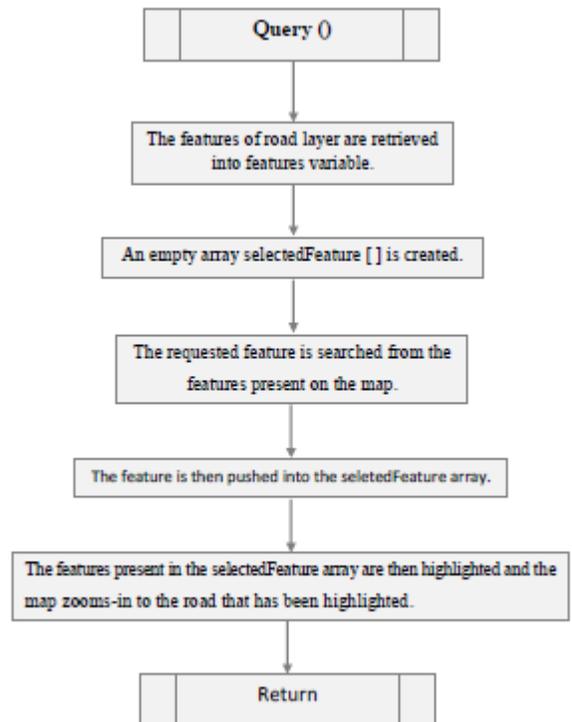
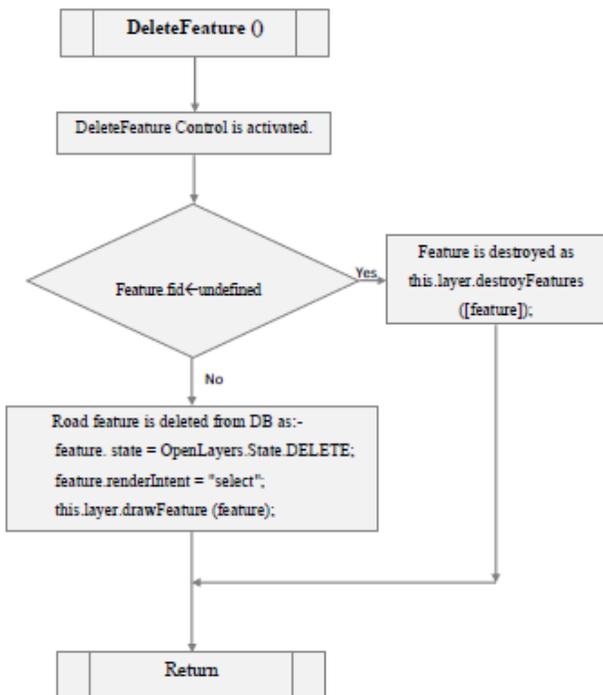


Fig. 6 Querying the feature from the base map

B. Pseudocode

Init() Module

/*
In this module the base map is created. We are using Google maps as the base map. After that a vector is overlaid over it to form the road layer

*/
Steps:
1. Begin
2. Map ← new OpenLayers.Map('map')
3. gsat ← new OpenLayers.Layer.Google("Base Layer")
4. map.projection ← "EPSG:900913"
5. saveStrategy ← new OpenLayers.Strategy.Save()
6. vectors ← new OpenLayers.Layer.Vector("Layer Name", strategies, protocol_Name(url,format))
7. map.addLayers ([gsat, vectors])
8. map.addControl (OpenLayers.Control.LayerSwitcher())
9. map.addControl(new OpenLayers.Control.MousePosition())
10. New ← OpenLayers.Control.DrawFeature(vectors,Handler_type)
11. Edit ← OpenLayers.Control.ModifyFeature(vectors,Handler_type)
12. Delete ← OpenLayers.Control.DeleteFeature(vectors,Handler_type)
13. map.setCenter(new OpenLayers.LonLat(latitude, longitude), zoom_level)

AddFeature() Module

/*
This module is used for creation of a feature with the help of vector editing.

*/
Steps:
1. feature ← map.featureToEdit;
2. feature.fid ← feature.id
3. feature.attributes.Name ← document.getElementById("attr_name").value
4. feature.attributes.Width ← document.getElementById("attr_width").value
5. feature.attributes.Type ← document.getElementById("attr_type").value
6. feature.state ← OpenLayers.State.INSERT
7. map.featureToEdit ← null
8. Return

EditFeature() Module

/*
This module deals with editing the existing features of the map.

*/
Steps:
1. feature ← map.featureToEdit;
2. feature.attributes.Name ← document.getElementById("attr_name").value
3. feature.attributes.Width ← document.getElementById("attr_width").value
4. feature.attributes.Type ← document.getElementById("attr_type").value
5. feature.state ← OpenLayers.State.INSERT
6. map.featureToEdit ← null
7. Return

DeleteFeature(feature) Module

/*
This module implements the deletion of a feature from the

map.

*/
Steps:
1. If feature.fid = undefined, then
2. this.layer.destroyFeatures([feature])
3. else
4. feature.state ← OpenLayers.State.DELETE
5. feature.renderIntent ← "select"
6. this.layer.drawFeature(feature)
7. End if
8. Return.

Query() Module

/*
This module deals with the steps needed to use customized queries based on user requirements.

*/
Steps:
1. layer ← map.getLayersByName("Road Layer")[0]
2. features ← layer.features
3. selectedFeatures ← []
4. for I ← 0 to features.length, do
5. var feature = features[i];
6. if(feature.attributes.Name/Width/Type== require_value, then
7. selectedFeatures.push(feature);
8. layer.selectedFeatures ← layer.selectedFeatures.concat(selectedFeatures)
9. End if.
10. End for.
11. for I ← 0 to selectedFeatures.length, do
12. selectedFeatures[i].renderIntent = "select";
13. vectors.redraw();
14. bounds ← selectedFeatures[i].geometry.getBounds();
15. End for.
16.
map.setCenter(bounds.getCenterLonLat(),map.getZoomForExtent(bounds));
17. Return

III. RESULTS AND DISCUSSIONS

The modules of the application were tested for their functionalities and they were found to be up to the mark vis-à-vis functionality. The working web based GIS package for small area thus can be used as a service to add, store, edit, delete and update the road feature and the associated details. The user interface provides nice and easy approach for the user and system interaction showing the power of open source technologies. A very simple GUI developed is easy to understand and allows the user to explore different functionalities of the package. Appropriate warning messages to indicate the errors have been provided when erroneous input is given by the users. These error handling features guide the user at each step while using the application. The user is prompted to re-enter the corresponding correct input value, when improper input is entered. The GIS package developed is very user-friendly, and provides a very convenient method to manage the information related to the roads of a small area.

This package can be used for any small city, campus or

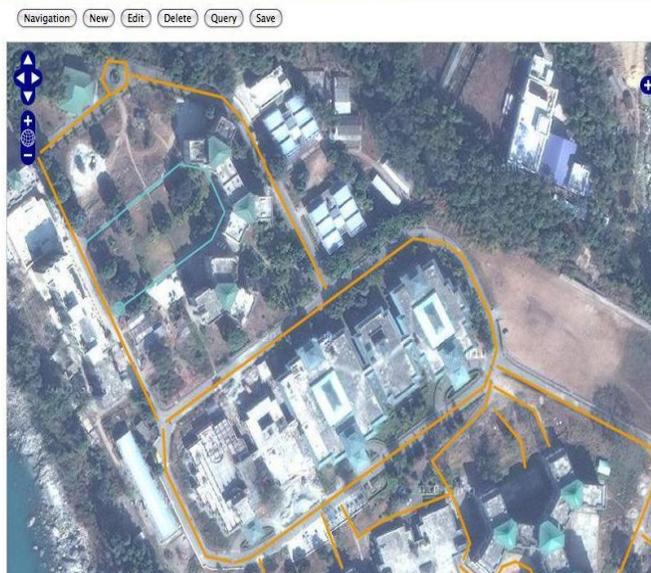
locality.

The latitude and longitude information are displayed at the top right corner of the map. A panel is there on the top left corner of frame, which facilitates the zoom-in and zoom-out functionalities. The switcher layer, which is used for switching between various layers, is shown in blue at top right side of the map just below the latitude-longitude information. It can be observed that Google Satellite image has been used as the base map and Road layer is the overlay. Some key screenshots are as shown below:

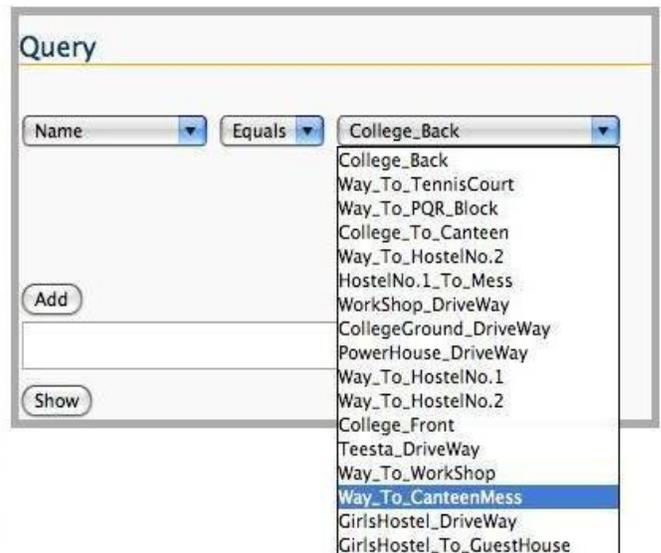
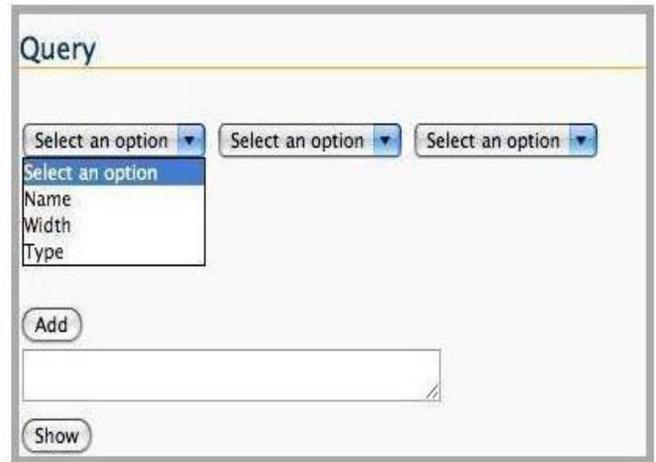
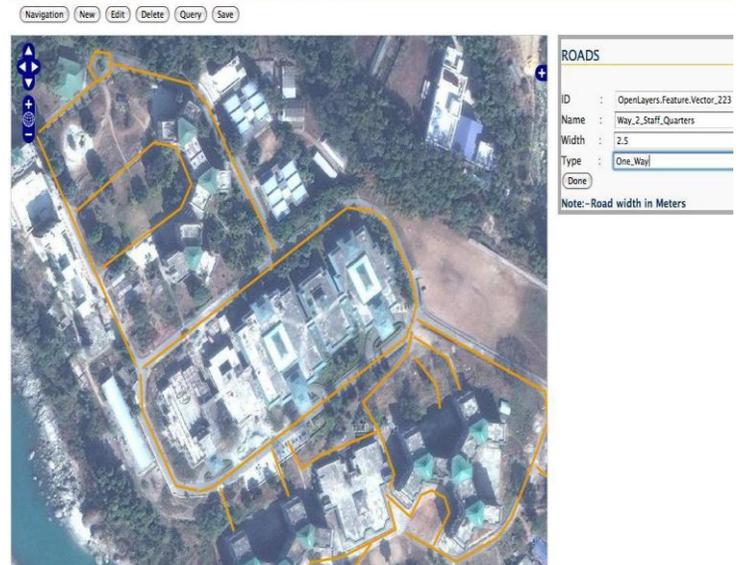
SmallCity - Together let's manage our neighbourhood!

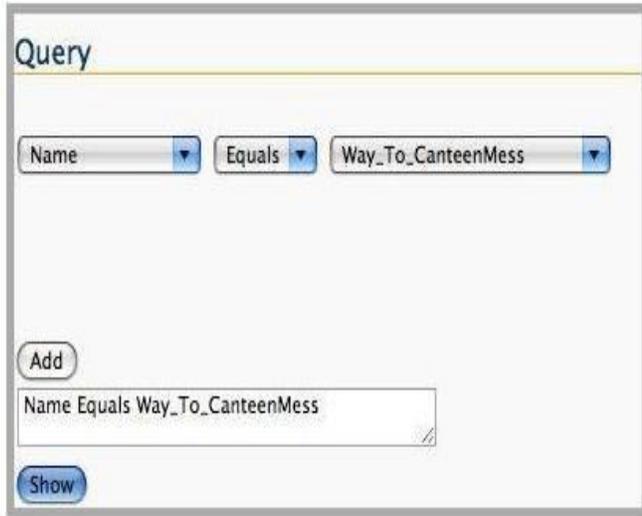


SmallCity - Together let's manage our neighbourhood!



SmallCity - Together let's manage our neighbourhood!





IV. FUTURE SCOPE

The system still has immense scope for expansions and some of them are enlisted below:-

- More base layers can be added like Virtual Earth, Bing, and MSN.
- More vector layers like Building layer, boundary layer, water body layer can be added.
- Separate user and admin modules should be implemented for better functionality and maintainability.
- To make it scalable the GET strategy need to be changed, features should be accessed within the bounding box or tile wise.
- Notice publishing feature can be included where users can put up notices indicating recent changes in the features like 'Road block for some specific reason'.
- Further the customization of the web service to provide for localized services can be done.

ACKNOWLEDGEMENT

Authors would like to acknowledge and thank Mr. Vineet Kumar, Mr. Suraj Pratap Singh and Mr. Adarsh Sameer Sharma ex-students of Sikkim Manipal Institute of Technology, Sikkim, India for the implementation of the idea.

REFERENCES

- [1] C. P. Lo & Albert K. W. Yeung, "Concepts and Techniques of Geographic Information Systems", 2nd edition, Prentice Hall, Aug 2006
- [2] Ian Heywood, Sarah Cornelius, Steve Carver & Srinivasa Raju, "An Introduction to Geographical Information Systems", 3rd edition, Pearson, 2010.
- [3] Peter Robins, "Using OpenLayers: Editing Vectors", April 2010, [available online <http://www.peterrobins.co.uk/it/oledit.html>]
- [4] Client Server Architecture, June 16, 2009, [available online <http://www.roseindia.net/java/network/client-server.shtml>]

- [5] Advantages and Disadvantages of Webservices, November 19, 2007, [available online <http://social.msdn.microsoft.com/forums/en-US/asmxandxml/thread/435f43a9-ee17-4700-8c9d-d9c3ba57b5ef/>]
- [6] OpenLayers Documentation:Controls, [available online <http://docs.openlayers.org/library/controls.html>]
- [7] Apache HTTP Server Version 2.2 Documentation, [available online <http://httpd.apache.org/docs/2.2/>]
- [8] CherryPy Tutorials, [available online <http://www.cherrypy.org/wiki/CherryPyTutorial>]



Dr. Roheet Bhatnagar Received his B.Tech. in Computer Science and Engineering and M.Tech. in Remote Sensing from Birla Institute of Technology, Mesra, Ranchi, India in 1996 and 2004 respectively. He did his PhD in Computer Science & Engineering from Sikkim Manipal Institute of Technology under Sikkim Manipal University, India. He is having more than 16.5 years of varied experience in the software industries and academics. He had worked in multinationals viz; Xerox Modicorp Ltd., Samsung SDS India Pvt. Ltd. and USHA Soft (a software subsidiary of USHA Martin Ltd.) in Gurgaon from 1997 till 2003 just after his graduation. During his stint in the industry he had a good exposure to software development executing many projects with different roles and responsibilities. He joined Department of Remote Sensing at BIT Mesra, Ranchi in the year 2003 and worked as Assistant Professor till 2008. He joined as an Associate Professor in Sikkim Manipal Institute of Technology (SMIT) - a constituent college of Sikkim Manipal University (SMU) in 2008 and is presently serving as Professor & Head in the Department of Computer Science and Engineering at Manipal University Jaipur. He has a number of publications in indexed international journals and national and international conferences. He is a life member of professional societies like Indian Society of Remote Sensing (ISRS), Indian Society of Technical Education (ISTE), and International Association of Engineers (IAENG). His current areas of interest are, soft computing, fuzzy and neural networks, database management systems, data mining and knowledge discovery, Remote Sensing and Geographical Information Systems (RS-GIS), and software engineering. He can be reached at roheetbhatnagar@yahoo.com and roheet.bhatnagar@jaipur.manipal.edu



Gagan Bansal is working as an Architect Rediff Maps (Application Development) at Rediff.com, Mumbai. He received his Engineering Degree in Civil Engineering from GZS College of Engineering and Technology, Bathinda, Punjab, India. He did his MTech in Remote Sensing from BIT Mesra, Ranchi and is vastly Experienced Professional. Prior to joining Rediff, he has worked with NRSA, Hyderabad and Reliance, Mumbai.