Web Based Geospatial Data Repository

By

Hassaan Ahmed Zuberi
(2010-NUST-SCEE-BE-GI-21)

Rehan Aziz Chaudhry
(2010-NUST-SCEE-BE-GI-32)

Umer Sharif Khan
(2010-NUST-SCEE-BE-GI-38)

Muhammad Umair
(2010-NUST-SCEE-BE-GI-29)

Ahmed Meer
(2010-NUST-SCEE-BE-GI-13)

A final year project submitted in partial fulfillment of the requirements for the degree Bachelors of Engineering in Geoinformatics

Institute of Geographical Information Systems

School of Civil and Environmental Engineering

National University of Sciences & Technology
Islamabad, Pakistan

(June, 2014)

CERTIFICATE

Certified that the contents and form of Final Year Project entitled **"Web Based Geospatial Data Repository"** submitted by Hassaan Ahmed Zuberi, Ahmed Meer, Rehan Aziz Chaudhry, Umer Sharif Khan and Muhammad Umair have been found satisfactory for the requirement of the degree.

Supervisor: ____________________

Assistant Professor (Dr. Ali Tahir, IGIS-NUST)

Member: _______________________

Lecturer (Ms. Khunsa Fatima, IGIS-NUST)

Member: _______________________

Lecturer (Ms.Saadia Irum, IGIS-NUST)

External Examiner: _________________

Name______________________________

Designation_________________________

DEDICATION

With consummate sincerity, we continue to affectionately dedicate this project to our parents and to all those, whose prayers pave the way for our success.

ACKNOWLEDGEMENTS

Thanks to Almighty Allah for giving us knowledge, power and strength to accomplish this task. We learned a lot while doing this project and this will certainly help us in our forth coming life.

We like to thank our supervisor Dr. Ali Tahir and co-supervisors Ms. Khunsa Fatima and Ms. Saadia Irum, who went beyond the call of our duty in providing us the numerous detailed comments on technical and pedagogical matters. Their supervision helped us a lot during the times of difficulties.

We wish to thank our teachers for their detailed suggestions and insights especially Dr. Hamid Mehmood, characteristics of their thoughtful teaching style on the overall development of IGISpatia. We also like to thank Mr. Ariel Nunez and Mr. Simone Dalmasso from Open Source Geonode Community for their technical help and support. We appreciate the efforts by Mr. Ishfaq Ahmed, NUST-SEECS and NUST-ICT for the hosting and web deployment of IGISpatia. We received an excellent input from the university and provide a very good prospect to share geospatial data online while focusing the requirements of the students at IGIS-NUST. So it was a very good experience.

We want to thank for the attention given by senior faculty of IGIS-NUST to make our geoportal (IGISpatia) worthwhile.

In the end, we would like to thank all of our class fellows, juniors, and people from masters for their support and encouragement.

Hassaan Ahmed Zuberi

Rehan Aziz Chaudhry

Muhammad Umair

Umer Sharif Khan

Ahmed Meer

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Explanation
NSDI	National Spatial Data Infrastructure
WMS	Web Map Service
WFS	Web Feature Service
WCS	Web Coverage Service
OGC	Open Geospatial Consortium
ISO	International Organization for Standardization
PND	Portable Navigation Device
CMS	Content Management System
HTTP	Hyper Text Transfer Protocol
SMTP	Simple Mail Transfer Protocol
HTML	Hyper Text Markup Language
SOP	Survey Of Pakistan
CSS	Cascading Style Sheets
API	Application Programming Interface
FTP	File Transfer Protocol
XML	Extensible Markup Language
UML	Unified Modelling Language
LTS	Long Term Support
MVC	Model-View-Controller architecture
CSW	Cataloging Service for Web
OMG	Object Management Group
SDLC	Software Development Life Cycle
GIS	Geographical Information Systems
DBMS	Database Management System
GSDIA	Global Spatial Data Infrastructure Association
NEDF	National Elevation Data Framework
VGI	Volunteered Graphic Information

ABSTRACT

This project titled as 'Web-Based Geospatial Data Repository' is a complete and fully functional web application which can be termed as a Geospatial Web Portal or Geoportal. The name of the final product is **IGISpatia**. The basic purpose of developing such a product is to provide an online interface for better management of geospatial data resources and sophisticated data discovery options. This product is intended to be used by the students, staff and faculty of Institute of Geographical Information Systems (IGIS), NUST. It is hosted over a live IP and can be accessed globally through the URL http://spatia.igis.nust.edu.pk. The application consists of various features and functionalities which enable a registered user to benefit from the repository. Only registered users have the access to use these features and view and/or download the geospatial data. To register for this site, one should be a member of IGIS-NUST, which means that he/she must be in possession of an email id which is on the domain of IGIS-NUST i.e. example@igis.nust.edu.pk .

A registered user can upload any geospatial data that he/she might possess. Further, he/she can also download the data uploaded by any other user if he/she allows it. Using the integrated application 'GeoExplorer', one can also create and edit/update the geospatial data and create and publish online maps. The geospatial web services WMS, WFS and WCS can be used by providing the credentials of registered user in a Desktop or Web applications. The geospatial data is uploaded based on metadata standards which are certified by ISO and Open Geospatial Consortium (OGC). The users can update and maintain their profiles like the user profiles in typical social networking sites. The users can also upload/download documents related to the uploaded data or maps. Further, the geospatial data resources can be searched using various methods which include searching based on keywords, searching based on geographic extent and searching based on dates.

As one can see that IGISpatia is fully equipped with all the features and functionalities of a typical Geoportal or SDI along with some extra functionalities which make it slightly different from those SDIs. These extra functionalities are like user interaction, commenting and rating options and much more. Therefore, this application is suited best for the purpose it was intended to be developed for, which is to facilitate the students of IGIS to promote geospatial data sharing and management

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

With the increasing urge of mankind to facilitate its standards of life, various fields of engineering, science and technology have emerged as a tool to provide human-like analysis and thinking capabilities to its end users. Geoinformatics engineering has served the humanity for the cause of providing a complete mashup of latest Information Technologies with Geography, Environment, Disaster and various other domains of studies. The basic purpose is to provide capabilities of spatial analyses with respect to some frame of reference. The latest tools and applications like Google Maps, Portable Navigation Devices (PNDs), Flood and Risk Management Applications and Urban Planning Tools are the best examples of the end-product of Geoinformatics research and engineering.

Since this domain of research and engineering is focused on the integration of Information Technology and other domains of study, there is always a need to manage and assemble the resources in the software or softcopy format. In other words, for any Geoinformatics product or application, there is always a spatial database at its backend which is responsible for managing the geospatial data resources associated with that application. This leads to a very common problem which is usually faced by the Software Industry professionals, termed as 'Data Management'. In the context of Geoinformatics, Data Management refers to the assembling of geospatial data resources in a useful and proper way so that it can be retrieved back anytime when needed without wasting much time and spending other costs over it.

Geospatial refers to the digital data having some locational information. Hence, the data management and maintenance of geospatial data is more complex and challenging as compared to normal digital data which may refer to images, text, videos etc. The data used in Geoinformatics domain needs to preserve its spatial information as well as the non-spatial information and the relation between them. By Geospatial data management, we mean that the data is stored in a spatial database management system which is able to retrieve the same when queried. Further, one should be able to search for any data in the spatial database or repository. This search might be based on spatial or non-spatial attributes of the data. For example, one may need to search for the data that lies within Islamabad (Spatial search) and one may search for data which was uploaded between the year 2013 and 2014.

The Geospatial industry has always faced the problem of geospatial data management. As a result, various solutions and products have been developed by different organizations and institutions world-wide. The term Spatial Data Infrastructure (SDI) refers to such a system which is capable of performing all the desired actions needed for the proper management of geospatial data resources. An organization maintains its Spatial Data Infrastructure to make its geospatial data available and easily accessible by its employees. It consists of various components which will be discussed in detail later. A Geospatial Data Repository may act as a node or a connecting point in a Spatial Data Infrastructure. A Spatial Data Infrastructure implemented nation-wide by a country is termed as National Spatial Data Infrastructure (NSDI). In this project, we have developed an individual Geospatial Data Repository which is actually a web-based product and can act as a connecting node for an SDI. This Web-Based Geospatial Data Repository has been developed for IGIS-NUST and deployed over web to provide a global access.

1.2 RATIONALE

When it comes to Geospatial Data Management, it is definitely an obvious problem to be faced to Geoinformatics experts. The students or researchers often seem to be in need of some particular dataset covering some geographic extent over some specific period of time. This leads to the search for geospatial data resource through various sources like online platforms, professional organizations and individual contacts. In most of the cases where one cannot find the required dataset, it needs to be created by the person himself. This leads to an additional step of ‘Data Preparation’ in one’s project or research. As we all know that the creation and editing of spatial data from base maps or scanned documents is a hectic activity, it also takes lots of time to prepare the final data. We carried out an online survey through the world-wide geospatial community of professionals, students and researchers available on a social networking site to find out the responses of people around the world related to the same problem. The question asked was a very simple one stating that 'How much time you spend on the Data Collection & Preparation phase of any Geoinformatics project or research'. According to the statistics, more than 40% of the people who responded said that 50% to 80% of the time is spent on this phase of Data Collection & Preparation. The four options provided to the people were as follows:

- Less than 20% time
- 20% to 50% time
- 50% to 80% time
- More than 80% time

The following graph shows the responses given by the people. For each of the four options, the percentage of responses that chose that option is mentioned in the figure.

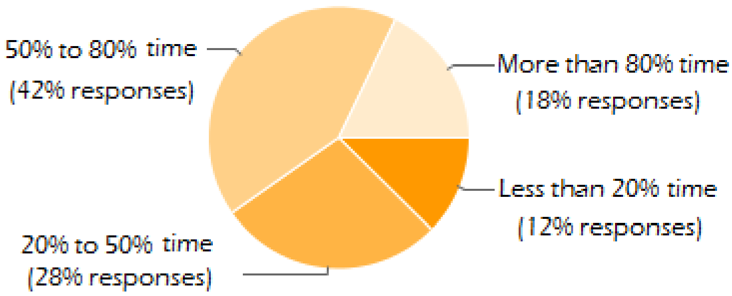

Figure 1 Online Survey results (January - June 2014)

These statistics support the objective of developing a product which can help reduce this time at least to some extent. The students in specific face lots of issues since they are not in a position to purchase any data from a professional organization or involve in any official processes in order to get their desired data. They always need a quick and free solution. The motivation behind this project is to facilitate the students of our institute in order to get the desired geospatial data if some other student have already prepared and uploaded it to share it with others. This will reduce the extra effort and repetitive tasks which usually form the basic component of any Geoinformatics project in the name of Redundant Data Preparation.

1.3 SCOPE

The Geospatial Data Repository is developed for the purpose of serving the students of IGIS-NUST. Any person from outside is not allowed to use the uploaded or shared data on the portal. A person needs to be registered to use all the features and services of this portal. The person who belongs to IGIS-NUST can be registered online.

It is important to mention that this project is scoped to serve this particular set of people as mentioned above. Therefore, it does not have a national or professional scope. It is not meant to be used by the general public. The instance which is deployed as its final product can be accessed online at http://spatia.igis.nust.edu.pk. A user needs to get himself or herself registered to view and download the data or create and use maps and web services.

The complete project consisted of two main components. First being the Analysis and Review component, in which the problem statement and our aims and objectives were carefully analyzed and worked out to give some output. This output helped in deciding what the actual product should be like. It set the standards and specifications of the final product. The second and last component was the Technical component. In this component, it was discussed among the product developers that which tools and technologies will serve the best for our product provided that all the previously defined features and specifications are properly met. For example, the decision as to whether we should use open source technology or the proprietary one for developing the final product.

After all the careful analyses through all the two stages of this project, the development team reached to some agreed criteria which was defining the overall specifications of the final product. At this stage, the project methodology was also developed which provided a guided path as to which set of tools and when to use them in which way. The basic idea behind the practical implementation of this product was to benefit the Geospatial Technology from the latest powerful technology of web. The concept of Content Management Systems (CMS) is widely used in the web development for the purpose of easy data management and access. The project's methodology defined our scope as the implementation of a Geospatial Content Management System with custom features which we intend to include in our portal. The front-end web application can be called a Geoportal since it serves all the basic purposes of a typical geospatial web portal. This portal can also play its role in the development of a greater SDI or in future for the NSDI of Pakistan. It follows the same concept of data sharing and data availability.

The usability of the web application is the component which is needs greater attention in the field of software development. For the purpose of this web portal, the usability factor was considered to be very important and the application was also tested from various dimensions and perspectives. The latest web technology makes it easier for us to use pre-defined templates made in HTML, CSS, JavaScript and these server as a general purpose template to deploy in one's application. The front end of the application, the color theme, the size of objects, the scale and many other factors were properly considered during the development process. The purpose of a developed product can only be achieved if the end user sees the interface as the developer expects him/her to see it. This is the reason that the software usability is always considered a very important factor in the software development life cycle.

It is to be mentioned here that all the tools and technologies used for the development and deployment of this product were open source. The reason to choose the open source technologies was to follow a cost effective pathway as well as the flexibility of these technologies in the context of core development.

1.3.1 Operating System

Linux is a Debian based open source operating system which is widely used by the open source community as well as in academic institutions for the purpose of learning and development. There are different flavors and versions of Linux operating system. Ubuntu 12.04 LTS was used as the operating system for developing this product.

1.3.2 Client-Side Development

The front end web development languages were used to prepare the end user interface. Hyper Text Markup Language (HTML) was used for the designing and layout of the front end. It is the basic web language needed to develop any web page or a web application. Cascading Style Sheets (CSS) was used for the purpose of adding the graphical element to the objects created using HTML. CSS allows the styling of objects over the web. It is featured with lot of options and techniques using which one can develop a very attractive and user-friendly interface. It adds the developer's choice by accessing the color, height, width, dimensions, coordinates and basic events of the HTML objects and their attributes. JavaScript was used as the client-side programming language to add dynamicity to the interface. The interactive images, slider, moving content, news feeds and other browser side functionality is provided by JavaScript and it is responsible for the element of usability in the web application.

1.3.3 Server-Side Development (Django Web Framework)

For the purpose of connecting to the server to put user generated data or retrieve already present data from the backend database, a server side language and a web development framework is used. For the purpose of this web application, the Django Web Application Framework was used. It is a Python based web application framework. Python is a highly interactive and user friendly open source High Level Language which is usually used for scripting of desktop applications or scripting over the web.

Django is a Model View Controller (MVC) based web application framework. Models refer to the abstract structure which defines the overall schema of a web application. Views refer to the templates which may or may not be based on Models. These provide the interface to transfer traffic between the server and the client. Controllers are responsible for the request response behavior between the Models and Views. It controls the flow of a request or a response that is generated as a result of some request.

1.3.4 Database (PostGIS)

The spatial extension of the PostgreSQL i.e. PostGIS was used to store and retrieve the spatial and non-spatial data. It is an open source Database Management System (DBMS) which is widely used in Desktop and Web applications. it provides the capability of storing, retrieving, analyzing and manipulation of spatial objects. Location based queries are supported by this DBMS.

Apart from these tools, the Apache2 web server was used for holding HTML pages and documents over the Hyper Text Transfer Protocol (HTTP). The basic application was deployed over this server. MOD_WSGI acted as a web application server for python specific apps in the application. Tomcat 7 was used as the web application server for the hosting of GIS Server i.e. GeoServer. Postfix was used as the Simple Mail Transfer Protocol (SMTP) server for the purpose of sending emails through the web application interface.

1.4 OBJECTIVE

The purpose, scope and background related knowledge as to why such a product was needed to develop have already been described in the previous sections. After the careful analysis of the whole development process, one needs to define a set of aims and objectives which support the execution of the whole process. The major objective behind the development of this product can also be inferred from the already described rationale and scope about this product. Keeping in mind the end users who are targeted to use this application and the purpose for which they will be using this product, one can easily define the objectives for this effort. It is necessary to define the objectives before starting the execution. In this way, one becomes very clear and straight forward during the development process and any other situation where some kind of decision making is involved. The aims and objectives defined for the development of our product are listed below in the form of points.

- To provide accessibility and management of geospatial data at no cost and encourage spatial data sharing among the students of IGIS-NUST.
- Provide online mapping capabilities and publishing web services.
- A step towards a complete academic SDI.

1.5 LITERATURE REVIEW

1.5.1 Spatial Data Infrastructure (SDI)

The concept about spatial data infrastructure was originated in 1993 by the U.S. National Research Council. It is basically a framework of technologies, policies, and institutional arrangements that provides the creation, exchange, and use of geospatial data and the related information. It also includes the data shared by some organization or community. We can implement this type of framework to provide the sharing of geospatial data and information within an organization or at broader level for use at national, regional, or global level. When we create a spatial data infrastructure (SDI), it ensures that data and resources are available to the organizations and various stakeholders that need them.**Invalid source specified.**.

The main components of SDI include:

- Policies and Institutional Arrangements (governance, data privacy, security, data sharing, cost recovery)
- People (training, professional development)
- Data (digital base map, thematic, statistical, etc.)
- Technology (hardware, software, networks, databases, technical implementation plans)

A spatial data infrastructure helps and facilitates sharing and management of spatial data among different people of the spatial data community. Carrying this this concept different countries in the world are developing SDIs so that the rich spatial assets could be used and managed in an effective way. The concept used in spatial data infrastructure is multi-disciplinary means it includes people, data, technical standards, government policies, technologies which are essential for the effective distribution of spatial data.

SDIs are developed at state and national levels. The government makes certain policies and initiatives that aid in the development of SDIs. Government also take the expertise of private sector in development and utilization of SDIs. We take the example of An Australian state Victoria. Within the state all the topographic and network data are prepared, managed and maintained by the private sectors taking into account the government policies and the service agreement between the two**Invalid source specified.**. The concept and implementation of SDIs are improved with the advancement of technologies like web 2.0.

1.5.2 Geoportal and their role in SDIs

A web portal is a website which act as an entry point of information in an organized and ordered way. The information can be videos, articles, documents, news, tutorials, datasets and services etc. Web portals are implemented using Service Oriented Architecture (SOA) based cataloging which help to link different resources or services, which means that they can be accessed and retrieved easily without any interoperability problems as SOA will manage the interoperability problems. SOA is implemented using the services concept (Halil AKINCI, 2014)

Geo portals uses the same concept as simple web portals but it is specially designed to deal with spatial datasets. Here the main purpose of Geo portals are the search and discovery of spatial data and other geospatial services that includes WMS, WFS and

WCS etc. Geo portals are used to implement SDIs (Spatial Data Infrastructures) and SDIs can also be said an interoperability infrastructures although they are not using the SOA concept (Halil AKINCI, 2014). The SOA concept is widely used to build interoperable distributed systems by providing the end user with the services that they can use in their own application. Unfortunately now days Geoportals cannot be implemented using the SOA concept because this technology is not mature and that's why all the SDIs are catalog portal based not SOA based. The SOA based SDIs would have been much more efficient in its performance.

The need for spatial data is increasing day by day as more and more people and organizations need geospatial data for variety of tasks, therefore the amount of spatial data is increasing day by day which need to manage that could be efficiently stored and retrieved when required. This necessity led us to build Geoportals that can easily fulfill these requirements. Geoportals have some basic functionalities that every geoportal is based on. The following diagram shows how user and data publisher interact with geoportal.

There are two broad categories of geoportal the catalog geoportal and application geoportal.

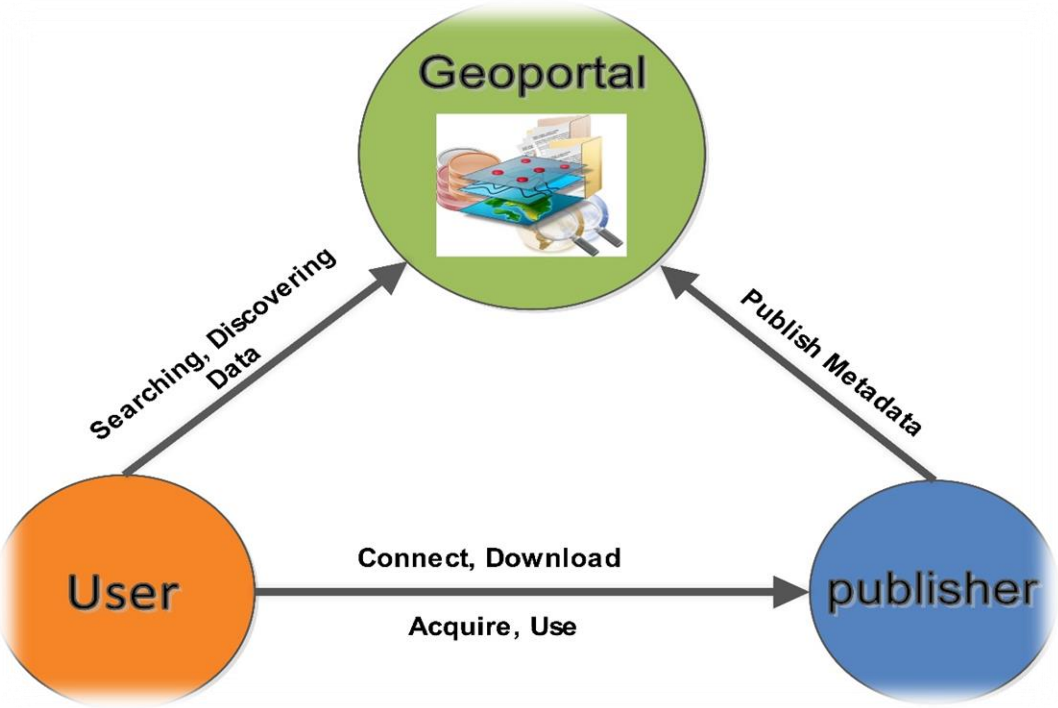

Figure 2: The Role of Geoportal in SDI

Application geoportal are those that online geospatial web services like MapQuest which provides routing services. While catalog geoportal provide cataloging services of the spatial data. Catalog geoportal are used for the implementation of SDIs. (David J. Maguire, 2014)

Geoportals are built using the web technology and the GIS concept. The user or client interact with the servers using the HTTP (Hyper Text Transfer Protocol). A geoportal is said to be a web site which is connected web server, geospatial server and the mapping server then connect to geo databases server which contain geospatial services and metadata information. The client request the required services through HTTP request and the services are transmitted using HTTP in an encoded message through XML (Extensible Markup language) (David J. Maguire, 2014).

The development of SDI is a step toward NSDI (National Spatial Data Infrastructure). Each SDI acts as a Node in NSDI. National data infrastructure contains all the policies and technologies used to manage, share and distribute geospatial data properly and in effective way. Many Government agencies and organizations are asked for quick reply to natural disasters, industrial accidents, environmental issues, and security problems. The information needed to make any decisions in such cases is based on geography.

There is pressure to make wise decisions in a more effective and efficient manner. Accurate and current geospatial data are very important to make these type of decisions.

It is the need of an hour that we have all the geospatial data on a single platform. It saves the data redundancy, which reduces the millions in terms of cost of data production. The NSDI is basically as the technology, policies, and people necessary to promote geospatial data sharing throughout the government sector, the private and non-profit sectors, and academic institutes. (The National Spatial Data Infrastructure, 2005)

The NSDI provides a base or structure of relations among data producers and users that will enhance data sharing capability. The higher the ability to share data through common standards and networks, will in turn, serve as a base for growth.

Building an effective NSDI requires a well-coordinated effort between Federal, tribal, State, local government, and academic institutions, as well as a various private sector geographic, statistical, demographic, and other business information providers and users. When we put all this effort then the NSDI will become a reality.

The first era when NSDI evolved is a government centric and based on data duplication to share geospatial data. Corporations mainly dependent on FTP sites to collaborate on data. The factors like time and cost makes difficult for the users to share geospatial data. The web 2.0 gives the new platform to share and collaborate on geospatial data. It includes the bottom-up flow of information, open and interoperable web services, mashups, Web-Oriented Architecture, and cloud computing. In the first generation NSDI which is NSDI 1.0 government was the main provider and collector of geospatial data. It was all government centric. But with the web 2.0 technology every citizen and private sector are involved in the collection of geospatial data. The term VGI (Volunteered geographic information) is used in geospatial context for people which voluntarily collect and contribute geospatial data. For example in Open Street Map people voluntarily contribute data. This is all possible because of web 2.0 which facilitated user participation in data collection and contribution. **Invalid source specified.** In the web 2.0 era VGI which the people (citizen) are providing are very crucial and playing very important role in NSDI 2.0. Before web 2.0 in the first generation NSDI citizen were neglected in the data collection state of the NSDI but now days it is not the case as we know that a lot of people and private sectors are involved the geospatial data collection and distribution. The only problem with the VGI is that we one cannot be sure about the quality and accuracy of such data. Another thing is the protection of personal privacy which is still not that much improved.**Invalid source specified.**

1.5.3 Global Spatial Data Infrastructure (GSDI)

GSDI (Global Spatial Data infrastructure) is another concept in which all the national SDIs will be integrated. As SDIs are developing in many countries in the world some international organization like Global Spatial Data infrastructure Association (GSDIA) and Group on Earth Observation (GEO) are working to promote and implement GSDI.**Invalid source specified.**

To develop GSDI all the SDIs should be harmonized. Multi-national SDIs are also promoting toward GSDI for example like the European Geoportal INSPIRE. There are many other programs that are developing towards The GSDI. Some of these programs include International Steering Committee for Global Mapping which is working to produce a Global Map. United Nations Educational Scientific and Cultural Organization (www.unesco.org) has played a role in the development of global soils databases. The united nation (UN) Food and Agriculture Organization has played a leading role in the development of global soli databases in 1970 on the map scale of 1:5,000,000. Another organization that is playing an important role is the United Nation Global Geospatial Information Management (UN-GGIM) organization which play a leading role in setting the agenda for the Development of GSDI. The Open GIS Consortium is an organization whose mission is the development of and use of open standards related to geospatial information and geoprocessing technologies.**Invalid source specified.**

Although the above work which is done by different organization and programs is not all which is required for the development of GSDI, these are considered as positive steps towards GSDI. There are many obstacle that need to be removed in order to complete the GSDI project. The collaboration of different countries and organization at different levels is necessary to harmonized all the technologies and policies that are considered crucial and important to effectively establish the steps and efforts required for the successful implementation of Global Spatial data infrastructure (GSDI) project.

As discussed earlier the second generation NSDI which was possible due to the web 2.0 is also playing an important role in development of GSDI because it enabled the VGI (volunteered Geographic information) which made the citizen capable of collecting and contributing geospatial data which is boosting data collection at global level.

1.5.4 SDI Examples

1.5.4.1 The INSPIRE Project

The European Union (EU) build an SDI known as INSPIRE. The INSPIRE portal is the Spatial Data infrastructure of European Union and set up by the European Parliament. It is established and operated by the 27 States of the European Union (EU). The people who are not familiar with the setup of the European Union regarding educational institutes, it is clear that the EU is not a federal state, it is a union of 27 Member States that agree on a number of international agreements.

In making of the INSPIRE SDI, the key decision-making bodies are, the national governments represented in the Council with a number of votes proportional to the size of the country. By including the European Parliament that is elected by universal suffrage every five years. The data comes from the responsible and professional organizations in the INSPIRE. The different SDI's of the European countries are combined to setup high level EU-SDI.

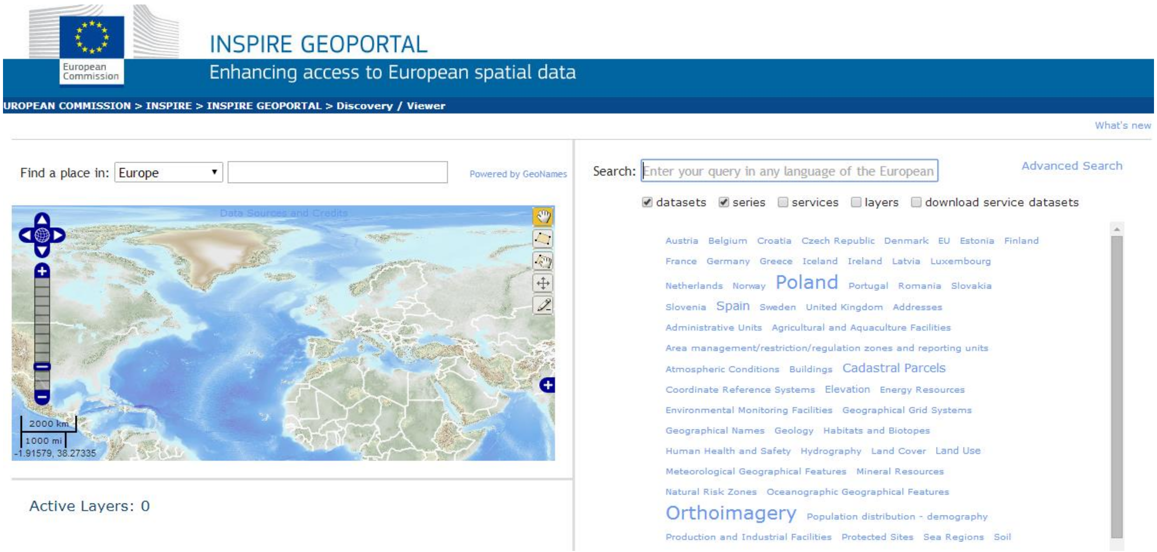

Figure 3: INSPIRE Geoportal

1.5.4.2 GeoGratis (Natural Resources Canada)

GeoGratis is an intuitive information service and highly successful Geoportal. This geoportal provide rich geospatial data at highest detail. The services and technologies it provides can easily be accessed and integrated. The geospatial data it provides it is absolutely free of cost. It provides almost all kind of data related to natural resources. It provides two kinds of data discovery by searching and by browsing the catalog. Both the methods are efficient for searching data.

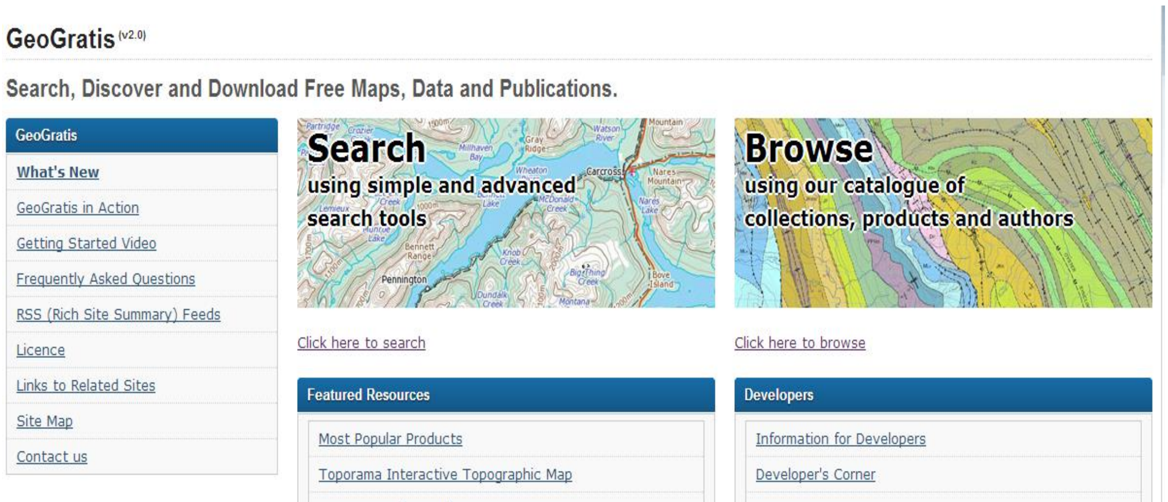

Figure 4: National Resources Canada (Canadian NSDI)

1.5.4.3 GeoBase

GeoBase is another Canadian Geoportal that provide free access to all Canadian geospatial data. GeoBase is federal, provincial and governmental initiative that is undertaken to provide free up to date and quality geospatial data for all of the Canada for free. The GeoBase project is developed by the collaboration between federal, provincial and terrestrial agencies. These partners are responsible for data creation and processing of geospatial data. In order to access GeoBase free data the user to first register online as registration is mandatory. It also provide access to free WMS (Web mapping services) compliant with OGC (Open Geospatial Consortium).

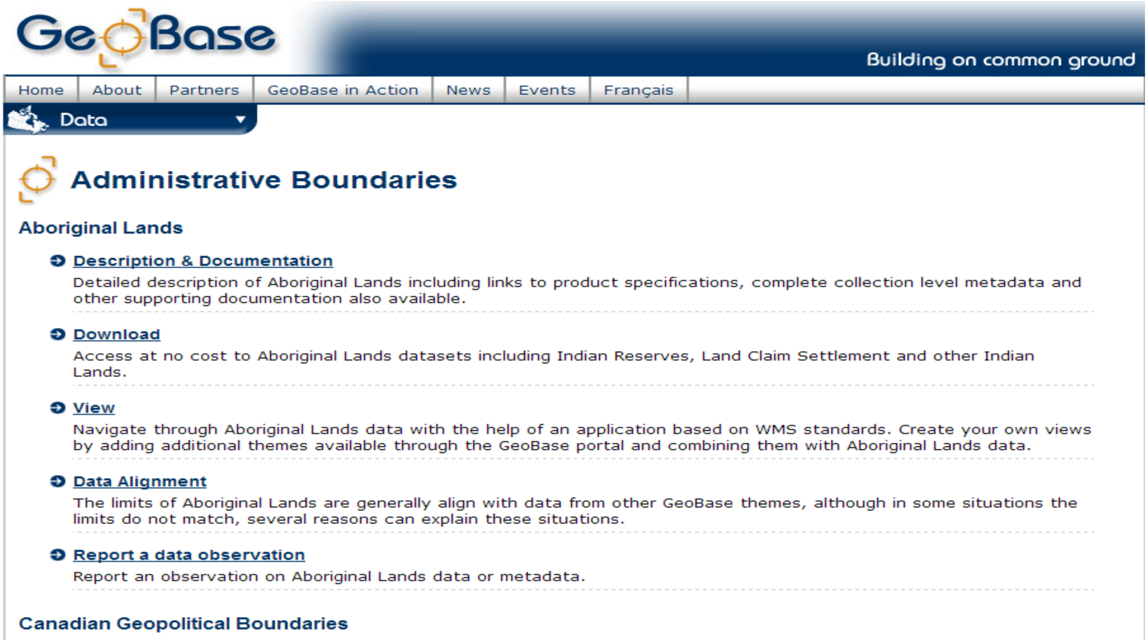

Figure 5: GeoBase Website Screenshot

1.5.4.4 NEDF (National Elevation Data Framework) Portal

The NEDF-portal is developed to establish virtual data repository and online portal that would server elevation data and the relevant metadata. Digital elevation data about the Australian landform and the seabed data is very important for evaluating the impact of climate changes, urban planning, water management etc. The portal is continuously populated with elevation data. One can search the elevation data from wide variety of options.

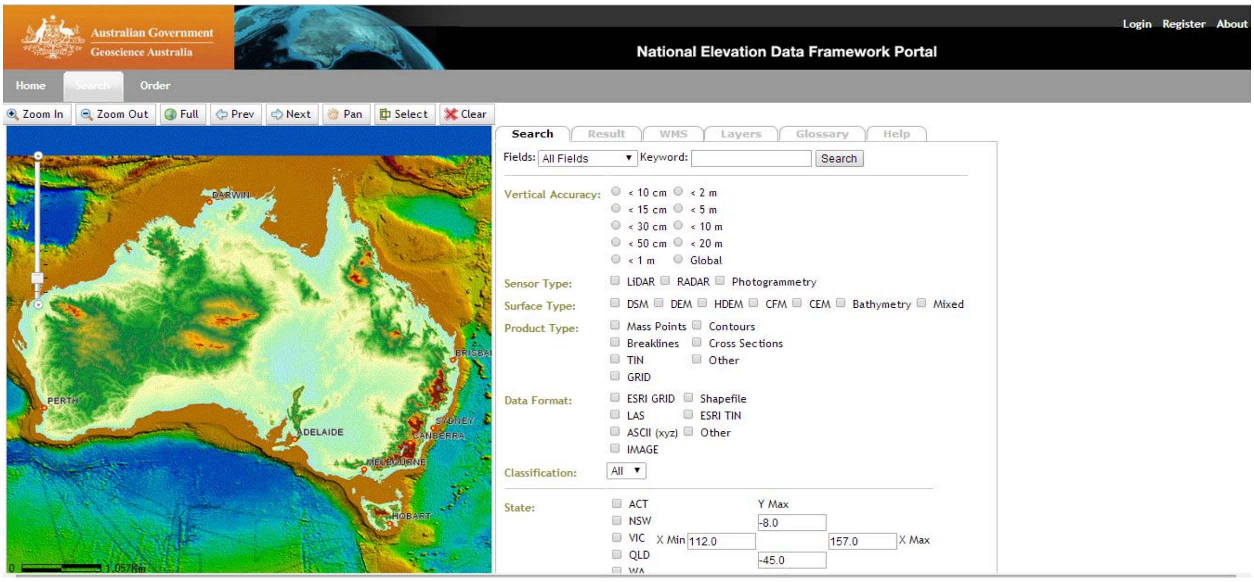

Figure 6: National Elevation Data Framework, Australia

1.5.5 Discussion Related To NSDI and Its Implementation in Pakistan

Our project comprises the development of an online spatial data repository system that is a component of National Spatial Data Infrastructure (NSDI). So it is important for us to investigate the procedures, techniques, frameworks and policy decisions involved in the successful implementation of NSDI. The development of an effective and efficient framework for the deployment of online spatial data repository is a critical step toward NSDI.

We have gathered some knowledge about the development, implementation, possibilities and problems regarding NSDI in Pakistan. It is a good thing to know that a bunch of people in Pakistan have done a lot of research in the field of SDI. While doing some internet research we came across that Mr. Asmat Ali (Assistant Director in Survey of Pakistan) had written handsome amount of research papers regarding problems and possibilities in the successful deployment of NSDI in Pakistan. Therefore we decided to meet Mr. Asmat Ali in person who has devoted his life in the field of SDI.

Mr. Asmat Ali have spent 25 years in spatial information production and management discipline as practitioner and educator. He has written more than 15 research papers and articles on GIS, Remote Sensing, SDI, E-governance, as well as Land Administration that have been published in various international conferences and renowned journals. He had been a part of the Faculty of Geo-Information Science and Earth Observation (ITC) of the University of Twente, Netherlands. He also did his Masters in Geo-Information from the same Institute. He did a Diploma in ArcCadastre/Cadaster from National Land Survey of Sweden. Currently he is an Assistant Director at Survey of Pakistan. Being foreign qualified and having years of experience in Geo-Information and SDIs we found that a formal meeting with him at Survey of Pakistan would be a great value for us to gain some insight regarding our project.

After some official recommendations and requests we were permitted to visit Survey of Pakistan (SOP) and meet Mr. Asmat Ali. On 12 November 2013, a warm welcome followed by a tour of SOP lead by Mr. Asmat Ali himself and his Assistant Mr. Munir. We visited the departments of photogrammetry, map making (cartography), surveying and leveling and at last printing department. After that we visited the Central DBA (IT section) which was of greater interest regarding our project. The Central DBA was further divided in two sections i.e. Server room and Control room. In control room we had been briefed about the local Geoportal which runs on the intranet of Survey of Pakistan. The geoportal has the facilities like view, search and read metadata of the datasets created at SOP. The framework is built on using PHP ArcServer, GeoNetwork and PostGIS as a Database engine.

At the end of the visit we came to the meeting room where we discussed some issues regarding the successful implementation of the NSDI. The first and the foremost issue is the Geospatial Data Sharing. Mr. Asmat highlighted some possible reasons that why people and organizations in Pakistan don’t want to share geospatial data like they didn’t create metadata information on their datasets, they fear that their data may inaccurate and not at proper standards, there are inaccuracies in data overlays which refers to incorrect geo-referencing, knowledge of people regarding Geo-Information etc. His assistant Mr. Munir also highlighted that Survey of Pakistan face some bandwidth issues and their current hardware does not support the deployment of NSDI. Similarly, there will be huge funding requirements if someone wants to deploy NSDI in Pakistan. Moreover, there were no established laws until 2014. Now there are defined laws under Surveying and Mapping Act 2014.

Mr. Asmat suggested to do some research to better understand the problems related to the development of NSDI. We were advised to visit other companies and government departments which are developing geospatial data. We had been advised to develop a questionnaire whether organizations are willing or not to share the data in the NSDI and if not then why? We were also advised that we should take sample spatial data from various organizations, overlay it and analyze their respective accuracy and the problems associated with it. We were also advised study how the spatial data from different institutions is represented at a single platform. This effort will contribute towards the development of NSDI in Pakistan.

1.5.6 Mashups

1.5.6.1 Geo Mashup

A mashup can be defined as an application that combine web services or resources dynamically from other different website and combine it to make a new application. A geo-mashup is a mashup which uses spatial geo-referenced services that could be one or more. (Pinde Fu, 2014).

The definition of Geo-mashup includes three important concepts:

Contents of web or its functions

These are actually the sources used to build the mashups. They can be contents such as data or maps, and they can also be functions which includes operations and processes. For example, when we talk about web services, where the input parameters are sent by users and the results are returned by the service using some specific logic. In this scenario, it is not just contents but the functions that are combined.

Dynamic

Dynamic means that the mashup should be generated automatically using the computer programs. It also keeps the links of the sources it used to a certain level. If the sources of the mashups are updated, the mashup application also becomes updated in a specified time frame. If we take a screen shot of a map manually, and stores it as a JPEG file and then upload it on the web site, and the information is overlaid on the top of it then the website is not considered a mashup, as we don’t have any dynamic link to the original source from where we pick the data.

Multiple Web Sites

One of the two sites is mashup application itself. If the pictures are overlaid on the map on a website from another website, the former website is a mashup.

A geo-mashup is a mashup in which at least one of the contents or functions is geo-referenced.

Geo-spatial mashup actually combine multiple data sources which is based on the same geographic locations. Mashups are capable of retrieving distributed data from the web and then overlay these layers dynamically. The concepts of mashups is related to web services, as to combine web services from different webservers.

There are two types of geo-mashups:

1.5.6.2 Server Side Mashup

Mashups that relied on web services, where the mashup processes are done on the server. The advantage of server side mashups are very clear that server has more powerful hardware and software, but these mashups also need complex server side programming and professional programming tools too, which is very complex when it comes to deployment and development. As they are complex, so these mashups are limited to professional programmers. Mashups didn't becomes famous when it just came because the web services are not as richer as it is now. As the technology grows with time, and enhance the mashups. (Pinde Fu, 2014)

This is the type of mashup in which the webserver sends request to different services or resources receives the responses and combine the services to make an application. This needs server side programming.

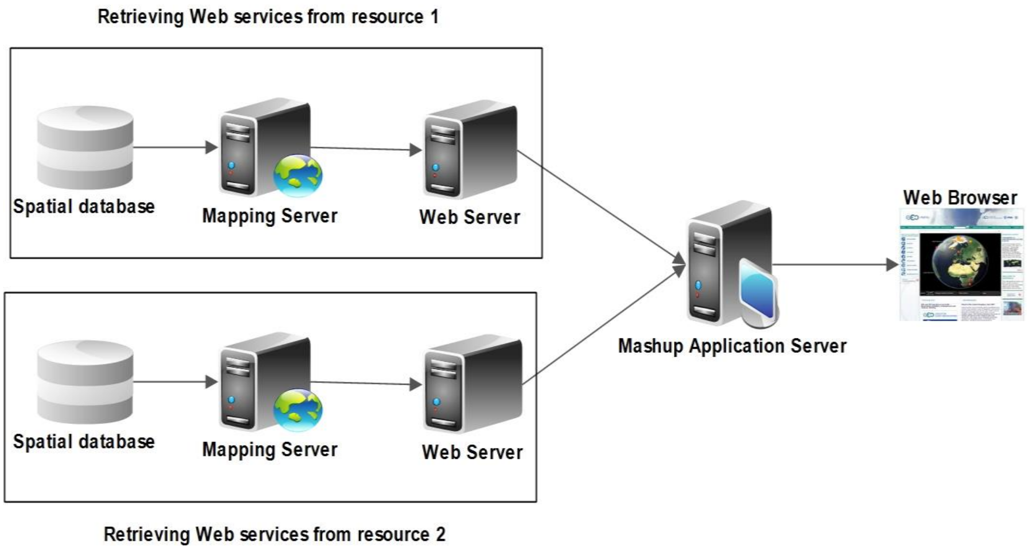

Figure 7: Server Side Mashup Architecture

1.5.6.3 Browser Side Mashup

Today's mashups are mostly done on the browser side. Their wide and rapid adoption is due to a rich contents and services that are available on the web and the popularity of AJAX and JavaScript. Most of the web mapping sites provide maps through JavaScript, which is not much complex as many users can learn it without any proper training.

The users can also view the sources of the scripts using the source function in the browser viewer. Therefore the developers that can use the code and make applications through it. Realizing the use of mashups and the number of customers they attract, the companies such as Google, Microsoft, Yahoo, and MapQuest officially announces their capabilities related to mapping through the JavaScript API.

Professional companies in GIS also use this approach as it is easy and very quick considering the development of web applications. Later on ESRI released ArcGIS API's for Flex, Silverlight and JavaScript. These API's reduce the complex server side programming and reduce the developer's effort needed to make complex applications with mashups.

In this type of mashups, the browser sends requests instead of servers for the services receive the responses and combine the result for new application. (Pinde Fu, 2014)

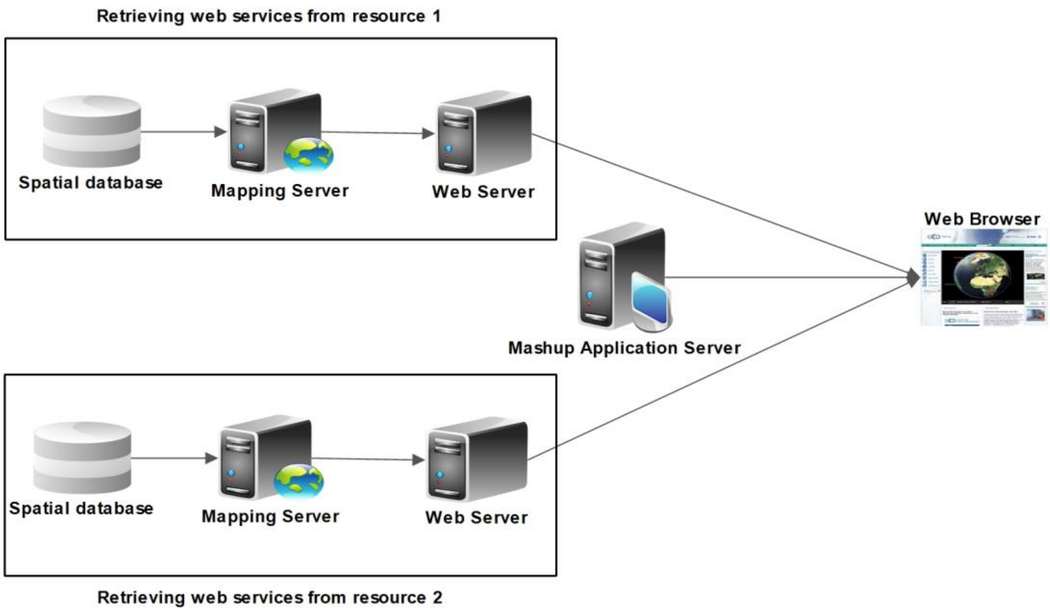

Figure 8: Browser Side Mashup Architecture

1.5.7 Web 2.0

Web 2.0 is defined as the sum of content generated by the user, using the web as a platform, and rich experience of the user. GIS and mapping companies follows the principle of web 2.0 and encouraging users to build their own applications using light weight API's. The examples of web 2.0 includes Google Earth, Google maps, Bing maps, Yahoo maps and Map Quest etc. They provide high resolution ground imageries of many regions around the globe.

Professional companies related to GIS adapt Web 2.0 principles and their patterns related to design in the field of Web GIS to facilitate communication, collaboration, sharing and the integration of geo-spatial information on the web.

The ESRI product line includes the following principles:

- Binding the collective intelligence and the data as the next "inside intel".
- Using the platform of the web.
- Programming style related to mashup.
- Mobile solutions.
- Rich user experience.

1.5.7.1 Distributed GIS

Distributed computing provide the fundamental standards and technology on which the internet and the Distributed GIS are built. The internet basically consist of standards related to Information Technology such as TCP/IP, HTTP, HTML, and XML including software, physical computing and the network infrastructure. The term SOA (Service Oriented Architecture) is used to refer to this technology of the internet. The framework provided by SOAs is very flexible and it supports range of applications. So the GIS industry exploit these kind of technologies and build the capabilities known as distributed GIS. Distributed GIS is actually a GIS technology that is built and deployed by using the software's and standards of the internet. (Tait, 2005)

1.5.7.2 Web GIS

Any GIS that uses the Web technology to communicate between the components. All the technologies related to Web as explained below are incorporated into the Web GIS. As the Mobile GIS also has connection with the Web, so we can say that it is the part of Web GIS.

1.5.8 Mobile GIS

The cell phones and other mobile devices, as many people uses them in this era, brought the Mobile GIS in a demanding stage. Mobile GIS is introduced in the context of Web GIS. In mid 1990s it is emerged as it fulfills the need of surveying work such as utility work. In early times, the mobile GIS works independently without connecting with the web. However, as time passes and the evolution of 3G networks enables us to connect the mobile devices with the Web and thus it becomes the part of Web GIS. By using the Mobile devices, we update the server information from the field.

The general public needs the Mobile GIS in a similar way as it is needed by the professionals and different organizations. Mapping applications that are built on the Mobile platform provides Location Based Services (LBS) regarding the geographic location of the concerned user.

1.5.8.1 Functions

Mobile GIS performs the following functions very well:

- Gathering of information and updating it on the web.

- Mobile devices are used by billions of people round the world, anyone can see and browse maps and routes from anywhere.

1.5.8.2 Advantages of Mobile GIS

Mobile GIS has many advantages over the Desktop GIS. These are:

- It reduces the paper based work, and manually entering of data on to the web server. This reduces the error chances and concurrency of data.
- Mobile devices are not attached to any wire or cable, as they are portable. And easily be held in the hand or pocket. Mobile GIS extends the data capturing process to areas where wired devices can't reach.
- As the volume of users are extensively large that use the Mobile devices. It broadens the GIS market in the context of Mobile GIS.
- Awareness of the location is very important thing as Mobile GIS provides. Major rescue services, survey, navigation and other LBS are the application of Mobile GIS. The ability to know where I am is the most important characteristic of the Mobile GIS.
- The other important and advantageous thing in Mobile GIS is the information. The information we get from Mobile devices and then updated on the web is almost real time, and it is very plus point of this technology.

1.5.8.3 Limitations to mobile GIS

Mobile GIS is relatively weak in some GIS functions. And these functions include:

- Mobile devices can't store large data, as they are not capable of it. The hardware is not as good as Desktop GIS has.
- As the computing power of Mobile devices are limited and it cannot provide the complex analysis. Mobile devise often have some specific purpose not used for the analysis.
- Another lack is the presentation in Mobile devices, as their screens are not as big as Desktop LCD's and the color combination.
- Limited internet bandwidth is also a problem as the Mobile devices have to transfer the real time data.

- Privacy of the user can be disturbed as the user's current location can be tracked easily.

1.5.9 Ubiquitous GIS

Due to extensive development of Mobile GIS, it contributes in many subjects and research areas, which includes Public Participation GIS, Augmented Reality, 4-D GIS and Dynamic Demography.

1.5.10 Public Participation GIS (PPGIS)

It actually describes the use of GIS in public domain. If we take every person having a cell phone as a sensor, and it captures information for some GIS application, it is worth using that information without any cost.

1.5.11 Augmented Reality

It basically combines the information in the database with the information from the senses. It provides the information regarding the subject area where the user with the Mobile device is present, from the GIS database as the user don't know where actually he is.

1.5.12 4-D GIS

The Mobile GIS when involves the 4^{th} dimension of time with the X, Y and Z dimensions, it is called as 4-D GIS. When we want to know where or when the geospatial events are taking place, Mobile GIS rapidly analyze the changing phenomena. This characteristic of Mobile GIS makes it 4-D GIS.

1.5.13 Dynamic Demography

As the population changes in a city from morning to evening depending on many factors. The census agencies capture the population information is static. As through Mobile GIS we can analyze the changing patterns of population within a city, which helps in mapping of real time traffic, and planning regarding transportation and in many other applications.

1.5.14 Technologies

The following section highlights some of the technologies that are used to develop and integrate frameworks for SDI's across the globe. Firstly a brief detail of different protocols used for establishing data communication links in SDI is given.

1.5.14.1 Protocols

Hypertext Transfer Protocol (HTTP) is a stateless protocol for communication between computers that are located at some distance. It defines how messages should be sent from one computer to another under certain standards. This kind of protocol is normally established between a client and a server or where a client server architecture exists between systems. In order to store data on servers POST request method is used by clients which have no limit on data length and are not stored in the browser as are not cached (httpmethods, 2014). In case of retrieving information from servers clients use GET request method which has limits on data length and remain in browser as can be cached.

File Transfer Protocol (FTP) is one of the simplest protocols established for exchanging files between computers on internet. It makes use of TCP/IP protocols. It allows the user simply to download an upload a file to server.

Simple Mail Transfer Protocol (SMTP) is a protocol for sending e-mails. It only works for outgoing messages. Its biggest advantage is its reliability and simplicity as either the message will be delivered to the recipient or an error will be generated for sender.

There are various scripting languages used in establishing an SDI amongst which some of them are server side languages while some are client side languages while some help in data transfer.

1.5.14.2 Languages

Hypertext Markup Language (HTML) is a client side language composed of tags. It is a description of the webpage and tells the web browser how the page will be displayed. An extension of HTML is HTML5 which gives web browser the ability to become a development platform (html5intro, 2014). It enables one to write cross platform and offline applications. Another scripting language is JavaScript which adds interactivity to webpages and can be embedded with HTML. It is an object oriented programming language.

Cascading style sheets (CSS) is used for the styling of webpages. Its main advantage is that it helps in keeping the styling layer separate from data transactions and access layer while CSS3 provides the user with opportunity to enhance styling with lightweight codes.

Java server side programming is a language that extends the ability of web servers and information systems as servlets which are made of java programming generate responses to requests from clients.

PHP is an open source server side programming language that helps in creating dynamic web pages. It is embedded with a HTML source document and can be used to design stand-alone graphical applications.

Python is a high level object-oriented programming language that helps in minimizing long tedious sequences of actions in performing a task. It has an easy to learn syntax which thus reduces the cost of maintenance. It helps in import of different modules such as “arcpy” which increases its reusability.

.NET is a proprietary programming framework model which helps you in building interactive and highly user friendly web applications. It has a large set of libraries and provides language interoperability which minimizes code writing and hence eases on the user to build applications.

Extensible Markup Language (XML) has been designed especially for web documents. It helps users in creating customized tags which helps in data transmission and interpretation between organizations. It is normally used over http POST and GET requests.

JavaScript Object Notation (JSON) is used an alternative to XML and uses normal readable text to transmit data and is a language independent data format. It is built on two structures which are following (json, 2014)

- A collection on value pairs
- On ordered list of values

1.5.14.3 Servers

After going through the languages we now see three most prominent database servers used in Geospatial Repository development which make use of the above language which are MySQL, POSTGIS, SQL SERVER.

MySQL is a database system that runs on web and can be used for both small and large applications. It is free, reliable and easy. PHP is normally used with it to provide cross platform compatibility. It is made by Oracle corp. SQL is used for querying in it

POSTGIS in an extension of an open source software PostgresSQL which has the ability of organization, storage and retrieval of spatial databases. It allows location queries to be run in SQL

MS SQL SERVER is a proprietary database server whose basic functionality is to store organize and retrieve data as requested by software’s through a relational database. It is the most suitable software to help in managing data in large organizations or through web

Apache web server is a secure, efficient and extensible server and provides HTTP services with pre-defined standards. It also supports modules for extensibility (apache, 2014). Tomcat is an application server established by Apache that executes Java servlets and renders Java webpages.

Jetty is another web server which uses Java and its API’s as programming languages. Developers can instantiate jetty as an object, instantly adding network and web connectivity to a stand-alone java app.

GeoServer is an open source GIS server written in java that allows you to write and edit geospatial data. It publishes data from any major data source under certain standards developed by OGC and W3C (GEOS, 2014).

Map server is flexible, reliable and is an open source environment for building web mapping applications. It has cross platform capability and supports many powerful languages (mapserver, 2014).

ArcGIS Server gives you the capability of sharing GIS resources across an organization or on web. It helps you in delivering spatial information as a service to the end users. It is proprietary and has been developed by ESRI.

1.5.14.4 Cataloging Services

The spatial data maintained and managed through different data and GIS servers are deployed across web as web services by different cataloging services

GeoNetwork is an open source spatial information management system that gives access to spatial data uploaded by variety of data providers. The data is uploaded with a descriptive metadata which helps and eases on different users to search specific datasets which may help in making decision making. The main goal of the software is to increase collaboration among GIS community and minimize data redundancy and give access to large resources of spatial data constructed by different organizations through one platform.

ESRI Geoportal Server is an open source product that helps in searching geospatial resources. It is a suite of different software packages that help in publishing data services and provides protocols for searching and managing of spatial data. It main component is a Geoportal which provides access to geospatial resources such as metadata records, catalogues, web services etc. (geoportal, 2014).

As the web arena has evolved so the methods of storing spatial data has also evolved and now web services are used.

Web Map Service (WMS) are used to for presenting spatial data as dynamic or interactive maps from a map server. They can be in the form of PNG, GIF or JPEG. They are commonly used by the GIS industry. It has two different request types (Sun, 2011)

- get capabilities: return parameters about WMS
- get map: returns a map image

Web Coverage Service (WCS) provides access to coverage data in a form that can be further used by clients in different models and systems. A WCS allows customers to select portion of data based on certain query. It provides data in its original form instead of pictures

Web Feature Service (WFS) is also known as the source code behind a map. It returns geographical features based on requests as compared to images which are delivered through WMS which cannot be edited or spatially analyzed. WFS allow you to create, update or delete features or get features based on certain constraints (Sun, 2011).

CHAPTER 2

MATERIALS AND METHODS

2.1 DESIGN AND ANALYSIS

After extensive literature review on various aspects of SDIs both technical and political as well as associated technologies required for successful implementation and deployment of SDI, we move forward to the design specifications of our Project. This will include system environment, web–architecture and application specific architecture. The next part of this chapter will focus on Unified Modelling Language (UML) Diagrams to understand the core working and flow of our SDI web application.

2.1.1 System environment

Our project is based on Open Source Development with all the technologies including servers, databases, cataloging services, internal/external apps, application engine and operating environment under Open Source Community License.

2.1.1.1 Linux

The Operating System (OS) used throughout the development of this project is Linux tough the final outcome (website/geoportal) can be accessed through any platform. Linux is one of the best examples of open source software collaboration. Its source code can be used, modified, and distributed commercially or non-commercially by anyone under General Public License (GNU). Linux is packaged in a format known as a Linux distribution for desktop and server use. Some mainstream Linux distributions include Debian, Ubuntu, Linux Mint, Fedora, Arch Linux, and the commercial Red Hat Enterprise Linux and SUSE Linux Enterprise Server.

2.1.1.2 Ubuntu 12.04 LTS

The Linux distribution used throughout this project is the latest Long Term Support (LTS) release of Ubuntu version 12.04. Long Term Support (LTS) release, indicating that such release is supported and receive updates for five years, with paid technical support also available from Canonical Ltd. The desktop version of LTS releases before 12.04 were supported for only three years. Non-LTS releases before 13.04 have typically been supported for 18 months, and have always been supported until at least the date of the next LTS release. 13.04 Non-LTS is supported for 9 months.

2.1.2 Web architecture

The web architecture followed in this project is a three-tier architecture with some specific modifications in the logical tier to allow spatial data processing and requests. Following are the tiers of web-architecture:

- Presentation Tier
- Logical Tier
- Data Tier

Following are the various components of web architecture used in this project.

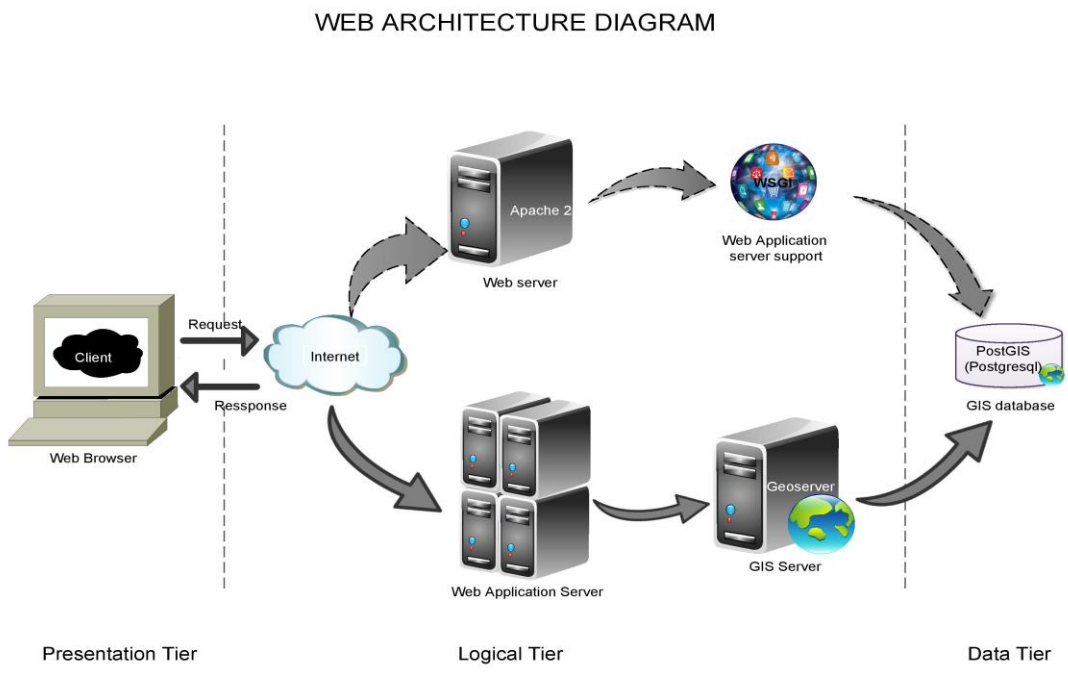

Figure 9: Web Architecture Diagram

2.1.2.1 Presentation Tier

Presentation tier is related to client side program whose main function is to display the final product. In this project web browser is the client side program which generates requests and receive responses and then display the results via internet. Web browser is independent of platform and operating system. The web portal can be accessed through laptop, PC, mobile etc. and so on any Operating System (OS). Popular Web Browsers can be Firefox, Chrome, Opera, and Internet Explorer etc.

2.1.2.2 Logical Tier

Logical Tier is linked with presentation tier on the front end and its main task is to process requests and generate responses. At the back end Logical tier is linked with data tier to record and store data/spatial data. As far as this project is concerned there are several components in the Logical Tier which are essential to process spatial and non-spatial data accompanied by the geoportal. Now we will discuss each component one by one.

Web Server

Web Server is the software (the computer application) that helps to deliver web content that can be accessed through the Internet. The most common use of web servers is to host websites, portals or geoportals. The key function of a web server is to store, process and route web pages to clients. The communication between client and server takes place using the Hypertext Transfer Protocol (HTTP) or (HTTPS). Web Pages appeared in Web Browsers are most frequently HTML documents, which may include images, style sheets, scripts and text content. In this project the web server used is Apache HTTP Server version 2.0. Apache Web Server is developed and maintained by an open source community. Most commonly it is used on a Unix-like system, like the one used in this project (Linux). Apache HTTP Server is available for a variety of operating systems (OS), including UNIX, FreeBSD, Linux, Solaris, Novell NetWare, OS X, and Microsoft Windows etc. Released under the Apache License, Apache is open-source software.

Web Application Server Support

This project is primarily based on PYTHON web application development and JAVA as secondary development language used for GIS Server management. Python web application frameworks are a problem for Python users because the choice of web framework would limit the choice of web servers. Python applications are often designed for only one of CGI, FastCGI, mod_python or custom API of a specific web-server. To overcome this, Web Server Gateway Interface (WSGI) provides universal interface between web servers and web applications or frameworks for the Python programming language. It was developed in December 2003. Since then it has been adopted as a standard for Python web application development. For this project, the details of this specification can be ignored.

Web Application Server

A web application server/framework is a software that is designed to support the development of dynamic websites, web applications, web portals, web services and web resources. The framework aims to ease several common activities performed in web development. For example, provide libraries for database access, template websites, security and permissions and session management, and they often encourage code reuse. As discussed earlier, this project is based on python web application development, so this project uses Django Application Framework as Web Application Server. Django is an open source web application framework, written in Python and follows a model–view–controller (MVC) architectural pattern. The primary goal of Django is to ease the creation of complex, database-driven websites. Django emphasizes reusability of components and rapid development. Python is used throughout, even for settings, files, and data models. Django provides an administrative interface to create, read, update and delete that is generated dynamically via admin models.

GIS Server

GIS Server is used for creating, managing and supporting GIS Web services, mapping applications, and spatial data. GIS Server is typically deployed in Logical Tier and is connected with application framework at the front end and with Data Tier at the back end. GIS Server aims to operate as a node within a Spatial Data Infrastructure. Just as the Apache HTTP Server has offered a free and open web server to publish HTML, GIS Server aims to do the same for geospatial data. The GIS Server used in this project is GeoServer which is an open-source server written in Java, allows users to share, process and edit geospatial data. It is designed for interoperability, it publishes data from any major spatial data source using open standards (OGC Standards). GeoServer functions as the reference application of the Open Geospatial Consortium Web Feature Service standard, and also apparatuses the Web Map Service, Web Coverage Service and Web Processing Service specifications.

2.1.2.3 Data Tier

In data tier the information is stored and retrieved from a database or a file system. The information is then pass back to the Logical Tier for processing and eventually back to the user. In this project the aim is to handle both spatial and non-spatial data. So the database used in this project should be spatially enabled (can store and retrieve spatial data). For this purpose we used PostgreSQL with its spatial extension known as PostGIS. PostgreSQL is an object-relational database management system (ORDBMS). As a database server, its primary function is to store data, securely and supporting best practices, and retrieve it later, as requested by other software application whether web based or desktop based, be it those on the same computer or those running on another computer across a network or through internet. It is an open source software, released under the PostgreSQL License. PostGIS is an open source software program that provide support for geographic objects to the PostgreSQL. PostGIS follows the Simple Features for SQL specification from the Open Geospatial Consortium (OGC).

2.1.3 System architecture

Application Architecture is based on a set of tools and libraries that provide the building blocks on which the IGISpatia is built. Following description and figure provide a basic understanding of each component critical to the successful development and implementation of IGISpatia.

Let’s look at each of these components and discuss how they are used within the IGISpatia application.

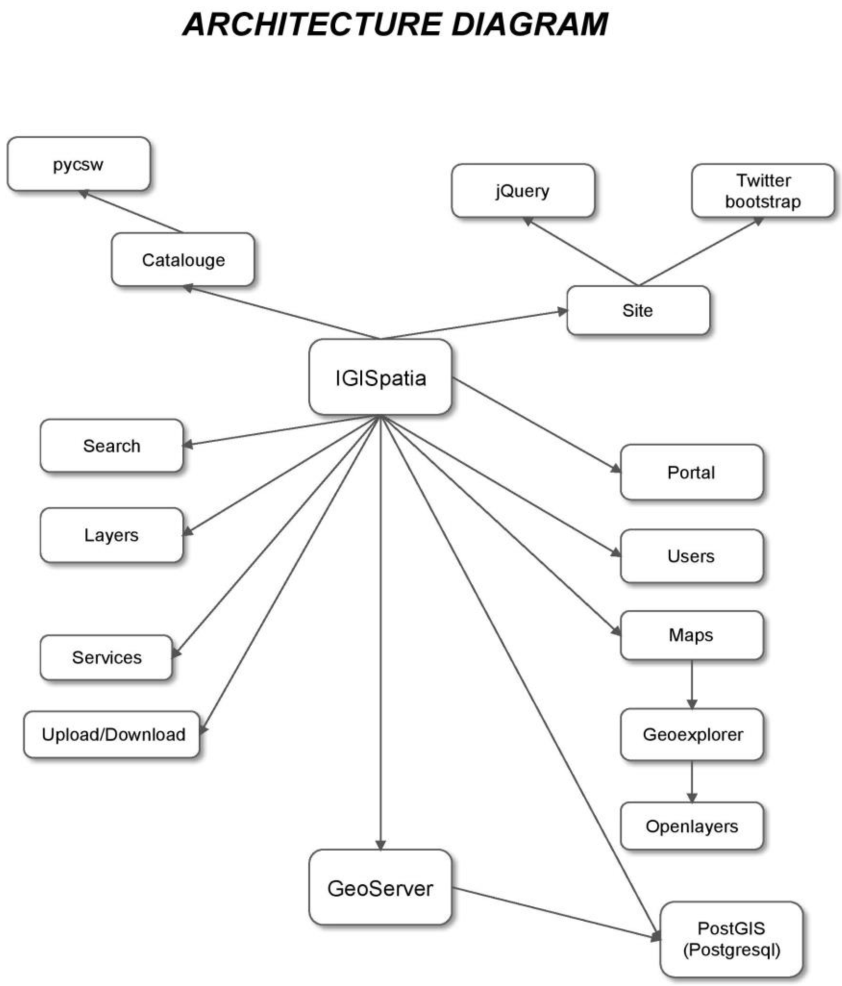

Figure 10: System Diagram

2.1.3.1 Django

IGISpatia is based on Django which is a high level Python web development framework that promotes rapid development and clean realistic design. Django is based on the Model View Controller (MVC) architecture pattern, IGISpatia models layers, maps and other modules with Django's Model module and these models are used via Django's ORM in views which contain the business logic of the IGISpatia and are used to enterprise HTML templates to display the web pages within the web portal.

2.1.3.2 GeoServer

GeoServer is an open source software server written in Java. It provides OGC compliant services which publish data from spatial data sources. GeoServer is used as the GIS component inside IGISpatia and is used to render the layers in IGISpatia, creating map tiles from the layers, providing functionalities for downloading those layers in various formats and to allow editing of those layers.

2.1.3.3 GeoExplorer

GeoExplorer is a web mapping application, based on the GeoExt framework, for creating and publishing web maps and other web based GIS Services. GeoExplorer is used inside IGISpatia to provide many of the GIS and cartography functions that are a core part of this mapping application.

2.1.3.4 PostgreSQL and PostGIS

PostgreSQL/PostGIS is the database component that stores and manages spatial data and non-spatial data for IGISpatia and the Django modules that it is composed of, pycsw (a metadata cataloging service) and GeoServer. All of these tables and data are stored within a database in PostgreSQL. GeoServer uses PostGIS to store and manage spatial vector data for each layer.

2.1.3.5 Pycsw

pycsw acts as a metadata cataloging server for spatial data. It is an OGC CSW server implementation written in Python. IGISpatia uses pycsw to provide an OGC standards-based CSW metadata and catalogue component for SDIs, supporting some of popular geospatial metadata standards such as Dublin Core, ISO 19115, FGDC and DIF.

2.1.3.6 Geospatial Python Libraries

IGISpatia uses several geospatial python libraries including gsconfig and OWSLib. gsconfig is used to communicate with GeoServer REST API to configure IGISpatia’s layers in GeoServer. OWSLib is used to communicate with GeoServer OGC services and to communicate with other OGC services.

2.1.3.7 Django Pluggables

IGISpatia uses a set of Django plugins referred as pluggables. Each of these pluggables delivers a particular set of functionality inside the application like Registration and Profiles, interactivity with external sites. These Django based plugins enables IGISpatia to take advantage of the large network of these pluggables out there while a specific set is included in IGISpatia.

2.1.3.8 jQuery

jQuery is a JavaScript library that is used within IGISpatia to provide an interactive and responsive user interface as part of the application. IGISpatia uses several jQuery plugins to offer specific pieces of functionality, and the new features can be added to the interface by adding extra plugins.

2.1.3.9 Bootstrap

Bootstrap is the front-end framework for laying out and styling the web pages that make up the outlook of IGISpatia. It is designed to ensure that the pages render and look and behave the same across all browsers and on all platforms.

2.1.4 UML Diagrams

The Unified Modeling Language (UML) is a modeling language in the field of software engineering, which is general purpose and designed to provide a standard way for the visualization of the design of a system.

It was developed in the mid-1990s as a collaborative effort by James Rumbaugh, Grady Booch and Ivar Jacobson, they developed their own symbols in the early 1990s. The 'U' in UML stands for 'unified', as the three developers combined the best features of the languages they had, each previously developed. The guardian of the UML standard is the ***Object Management Group (OMG)***. The OMG approved version 2.0 of UML in 2004.

It is a graphical visualization language and consists of a series of symbols and connectors that can be used to process and diagrams and is often used in modeling of computer programs and workflows.

UML is not just a set of notations but it is very effective for drawing diagrams, it has the following additional important and interesting features, these are:

- The diagrams we create with it are anticipated to be interconnected to make a unified model.
- It has a detailed semantics, and describe mathematically the meaning of every aspect of its notations.

- It also has an extension mechanisms, which allow its software designers to represent concepts that are not part of the core of UML.
- It also has an associated documented language called Object Constraint Language (OCL) that allows us to formally state the various facts about the elements of the diagrams.

The objective of UML diagrams is to assist in software development Life Cycle (SDLC). It is not a methodology, because it does not describe a step-by-step process of how we do and perform certain actions.

Some of the developers are successful in developing small software systems without using the diagrams and other features of modeling languages. However, when their systems become larger and larger and the data in their systems increases with time plus the increase in user requirements, such developers have very difficult time analyzing the 'big picture' and are likely to create poor designs and take too much time in their work.

Most software systems are then documented with the use of diagrams that explain the process efficiently and effectively. It also provides views of structure and functionality that would be difficult to hold by looking at code or the description in text alone. In simple words, diagrams provide abstraction.

A model is beyond an ordinary set of diagrams. A model contains an interrelated set of information about the system. A diagram is simply one dimensional view of that information. More than a few diagrams can present the same information by using slightly different approaches, either with different notations or with different levels of detail (grain of the data). I can delete any element from any diagram and keeping it in the model, if I delete an element from the model it should disappear from all the diagrams. (Laganiere, 2005)

2.1.5 Use case diagrams

Use case diagram explains the main functionality of the system. It actually cares about the systems behavior (what) not taking into account how it is done. A Use Case diagram is the best way to communicate to management, customers and other non-technical people that what a system will do when it is completed. (Laganiere, 2005)

Use Case diagram has following components:

2.1.5.1 *Actors*

Set of actors having the roles that a user can play in the system. In our Project the actors mainly are:

- Registered User
    - Administrator
    - Staff User
    - Active User
- Non-registered User

Actors are shown as:

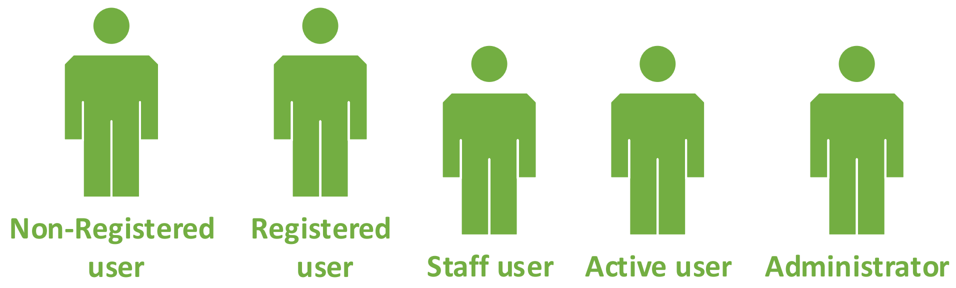

Figure 11: IGISpatia Users

2.1.5.2 *Use Cases*

Each Use Case explains a possible kind of interaction between an actor and the system. Basically they are the actions that a user can perform. It is represented by an ellipse like:

Figure 12: Use Case Example

2.1.5.3 Relationships

The relationships are between actors and the use cases. They are simply illustrated with a line which connects actors and use cases. It is represented as a single line.

The Use Case diagram for the IGISpatia portal is:

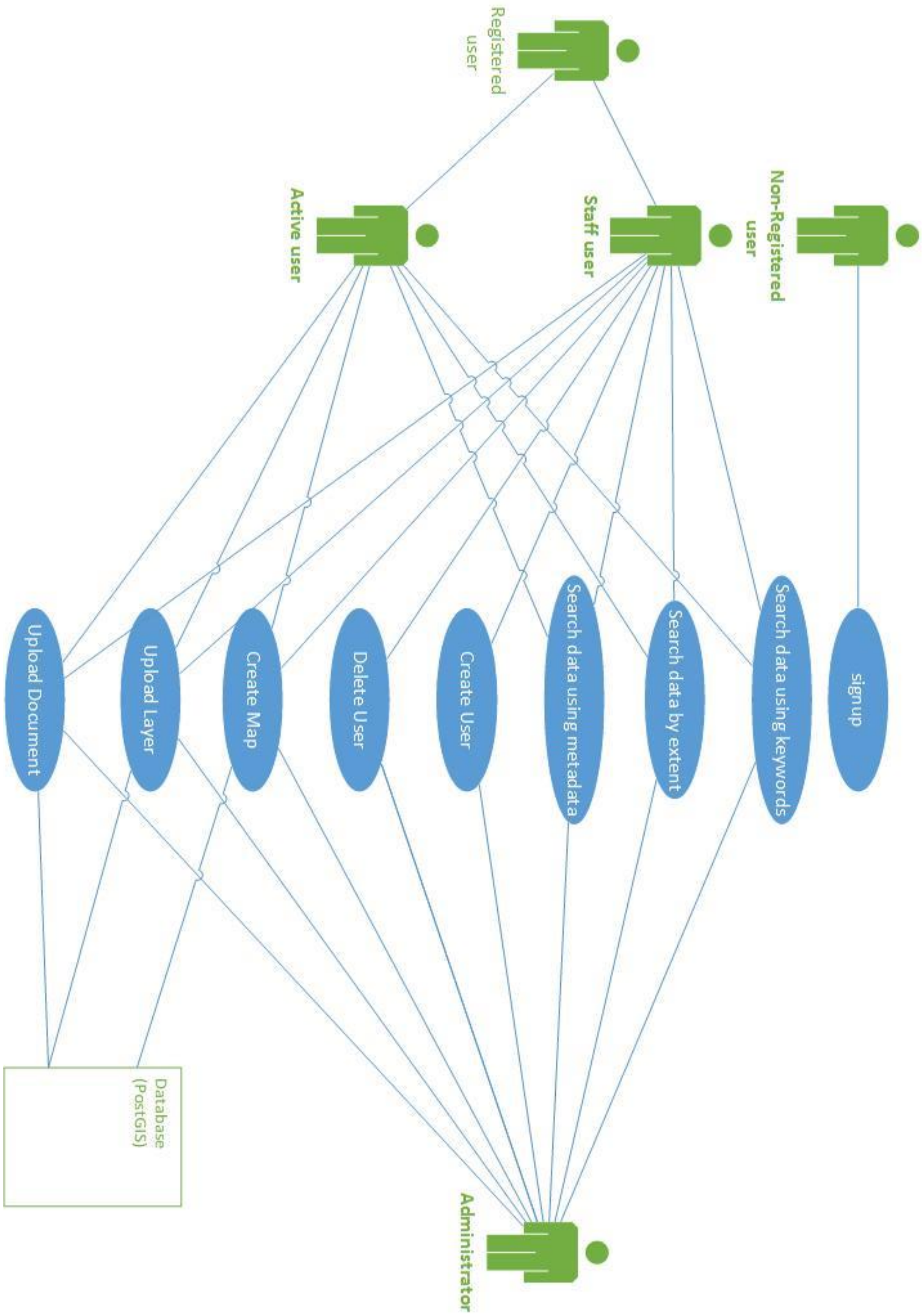

Figure 13: Use Case Diagram for Account Management and Search

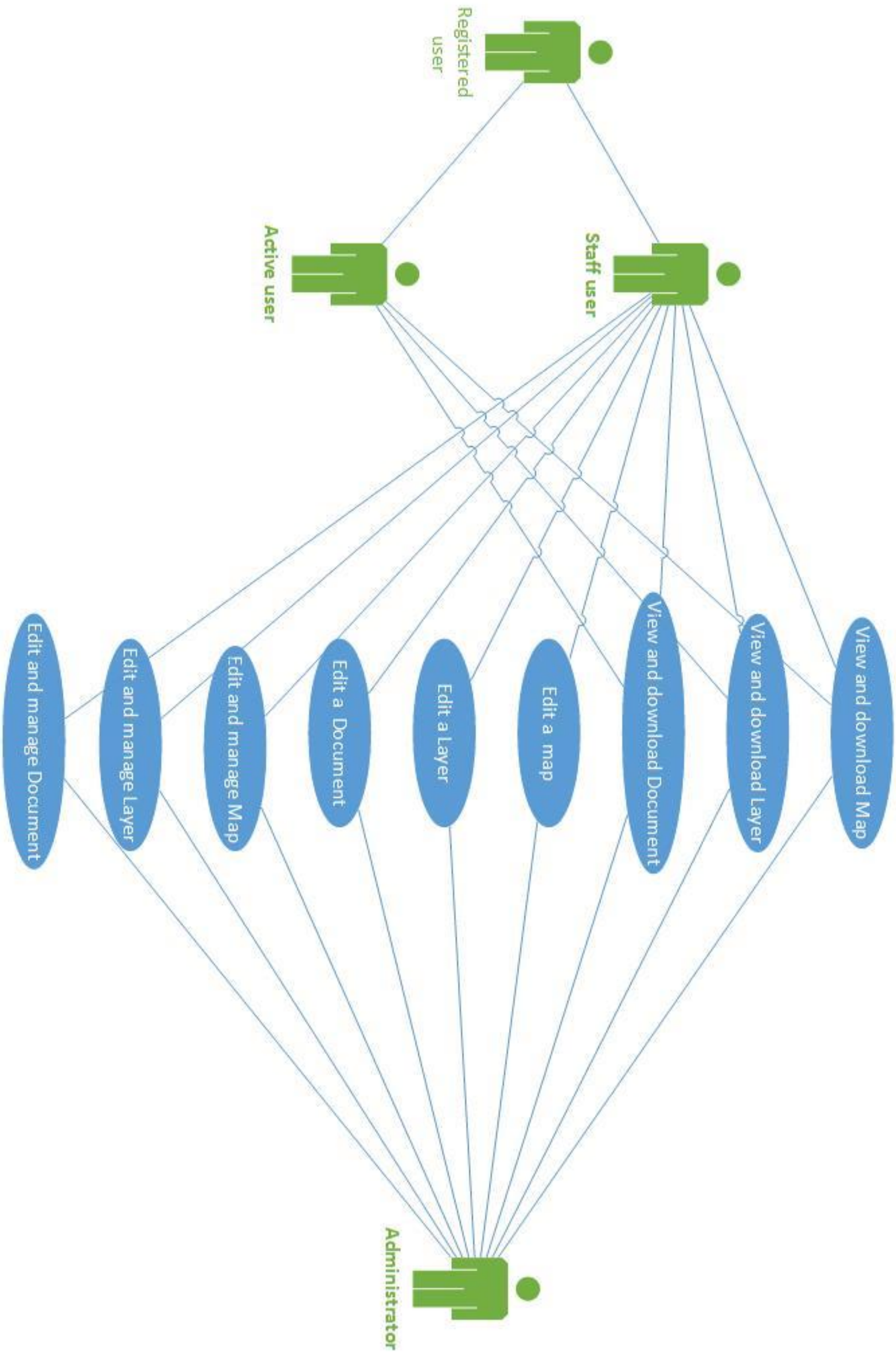

Figure 14: Use Case Diagram for View and Edit

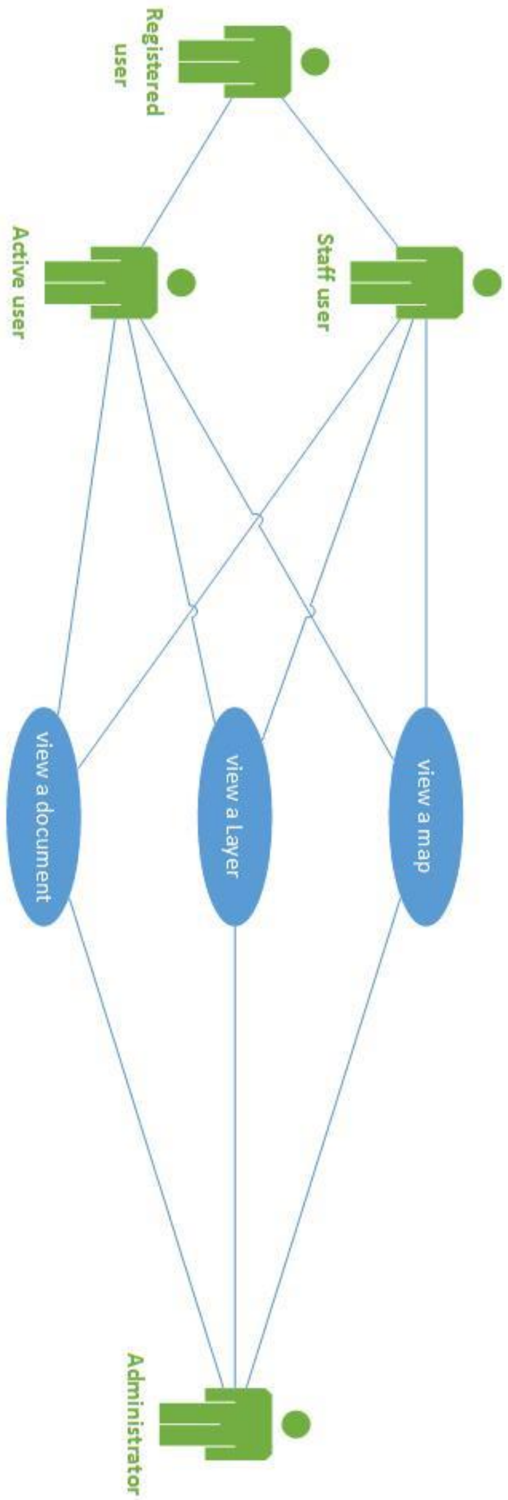

Figure 15: Use Case Diagram for View

2.1.6 Activity Diagrams

Activity diagrams are actually related to the program flow plans that are also called flowcharts, are used to explain the activities. In an external view, we use these diagrams for the description of software processes that describe the functionality of the software system.

Unlike the use case diagrams, in activity diagrams it is true whether actors can perform business use cases together or independently from one another.

These diagrams allow us to think more functionally. Traditionally, the object-oriented approach probably dislike this fact. We on the other side consider this fact as a great advantage, since users of object-oriented methods, as well as users of functional thinking patterns, we find a common and familiar display format, which is also a significant aid for business-process modeling.

In these diagrams, we just represent how the actions are performed in the IGISpatia portal. As these diagrams are helpful for the users to get the clear meaning of the flow of the actions that are performed in the portal for specific activity. (Laganiere, 2005)

Activity diagrams are made using a limited number of shapes which are connected with arrows. The most important and mostly used shapes are:

- Rounded rectangles from all four corners represent actions.
- Diamonds represents the decisions.
- Bars represent the split and end join of simultaneous activities.
- A shaded circle represents the start of the diagram.
- An encircled shaded circle represents the end.

The Activity diagrams we make are listed as follows:

- Sign-up
- Search Data
- Add User
- Delete User
- Create Map
- Upload Data (Layers and Documents)

- View and Download Data (Layers, Maps and Documents)

The diagrams are as follows:

Figure 16: SignUp

1- Sign-up

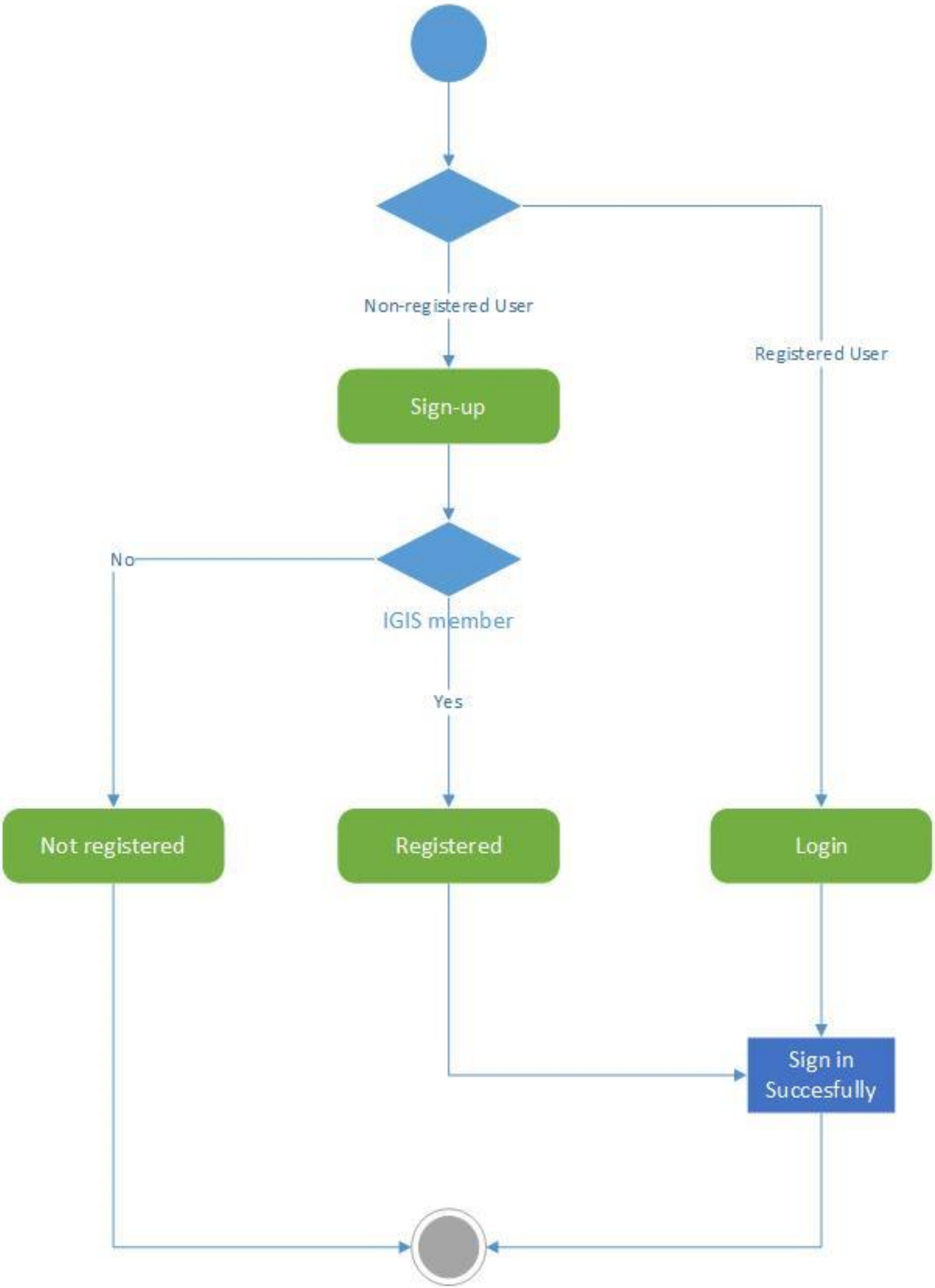

Figure 16: SignUp

2-Search data

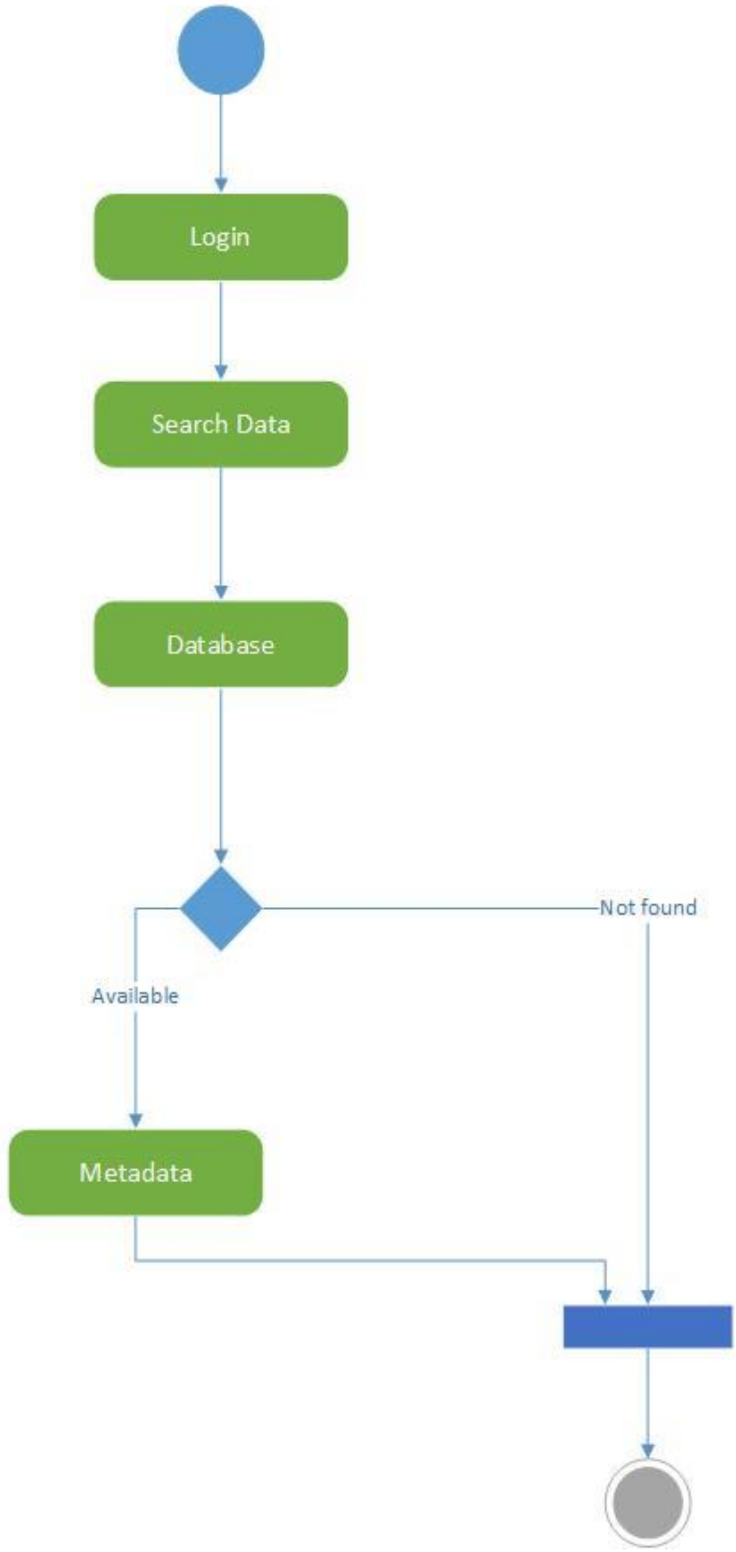

Figure 17: Search Data

3-Add user

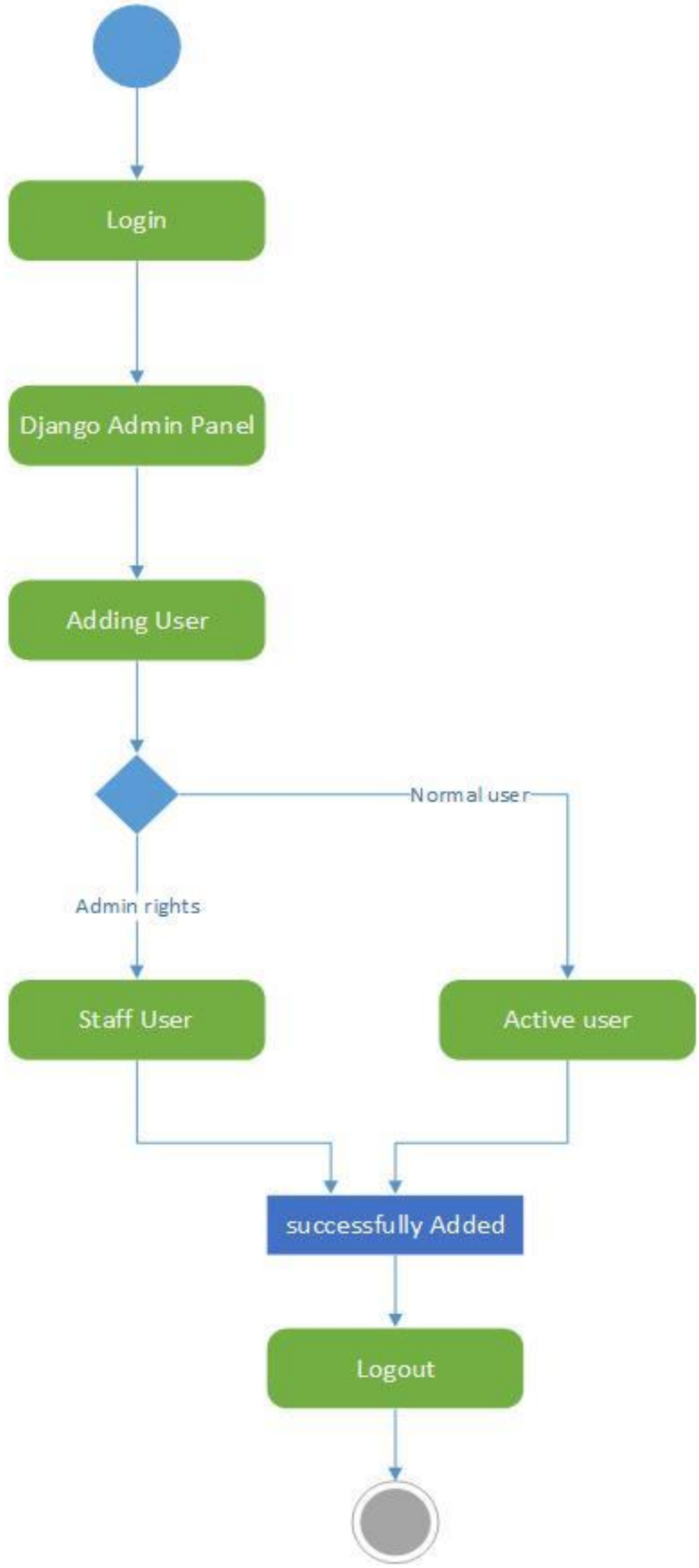

Figure 18: Add User

4-Delete user

Figure 19: Delete User

5-Create Map

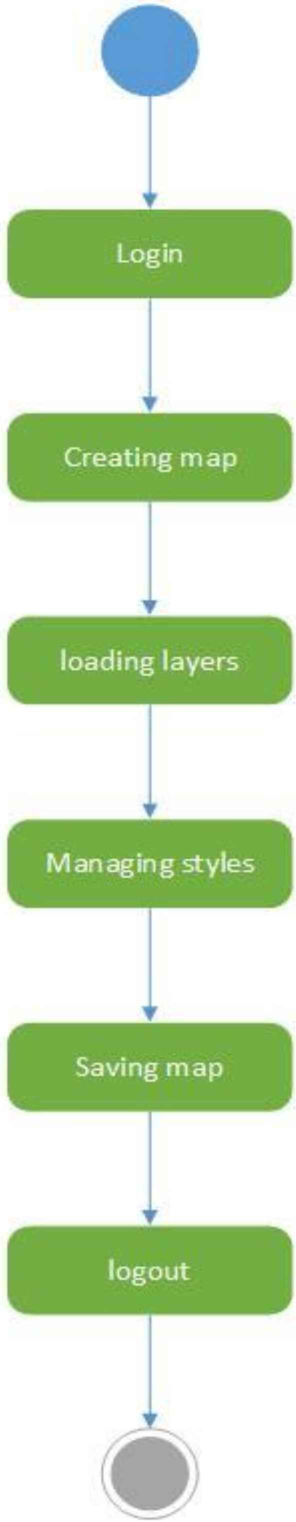

Figure 20: Create Map

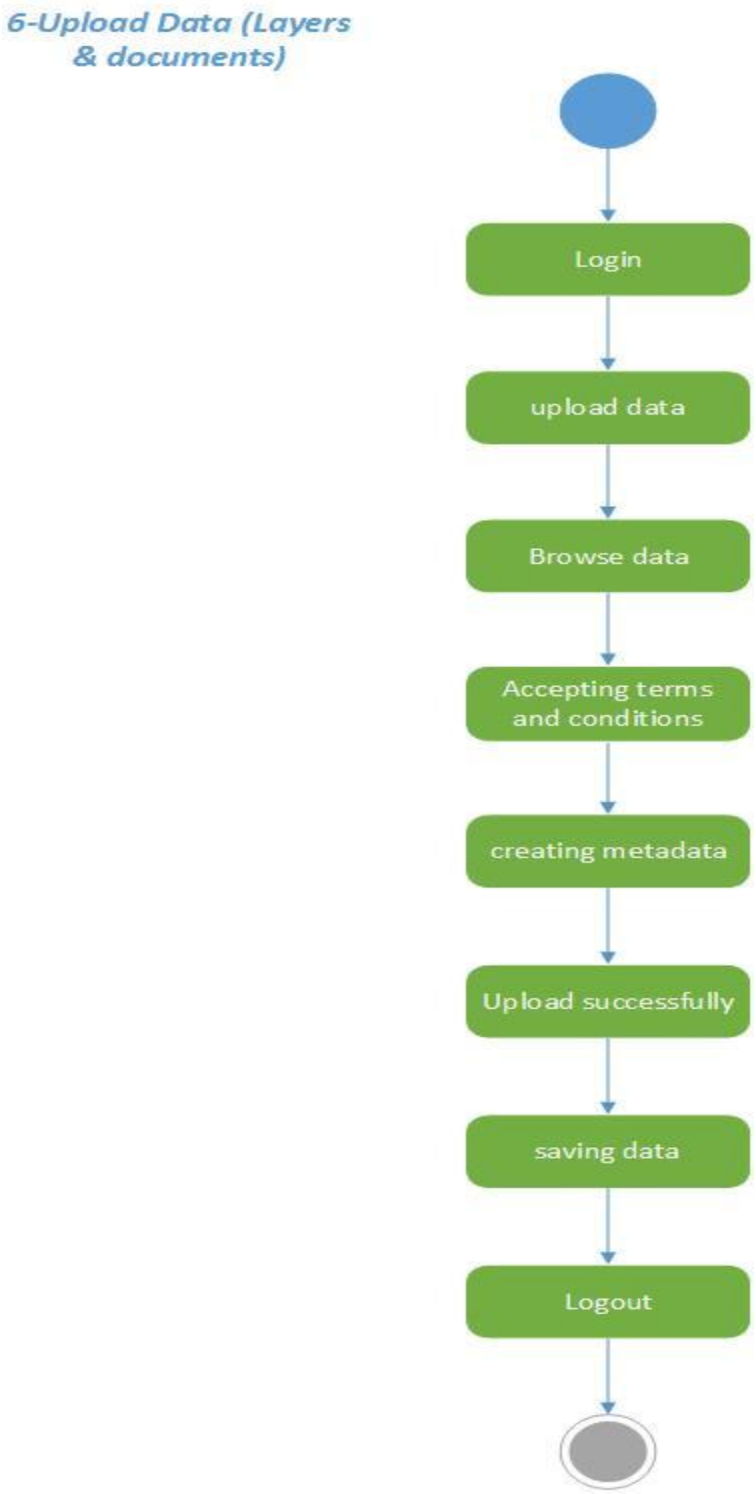

Figure 21: Upload Data (Layers and Documents)

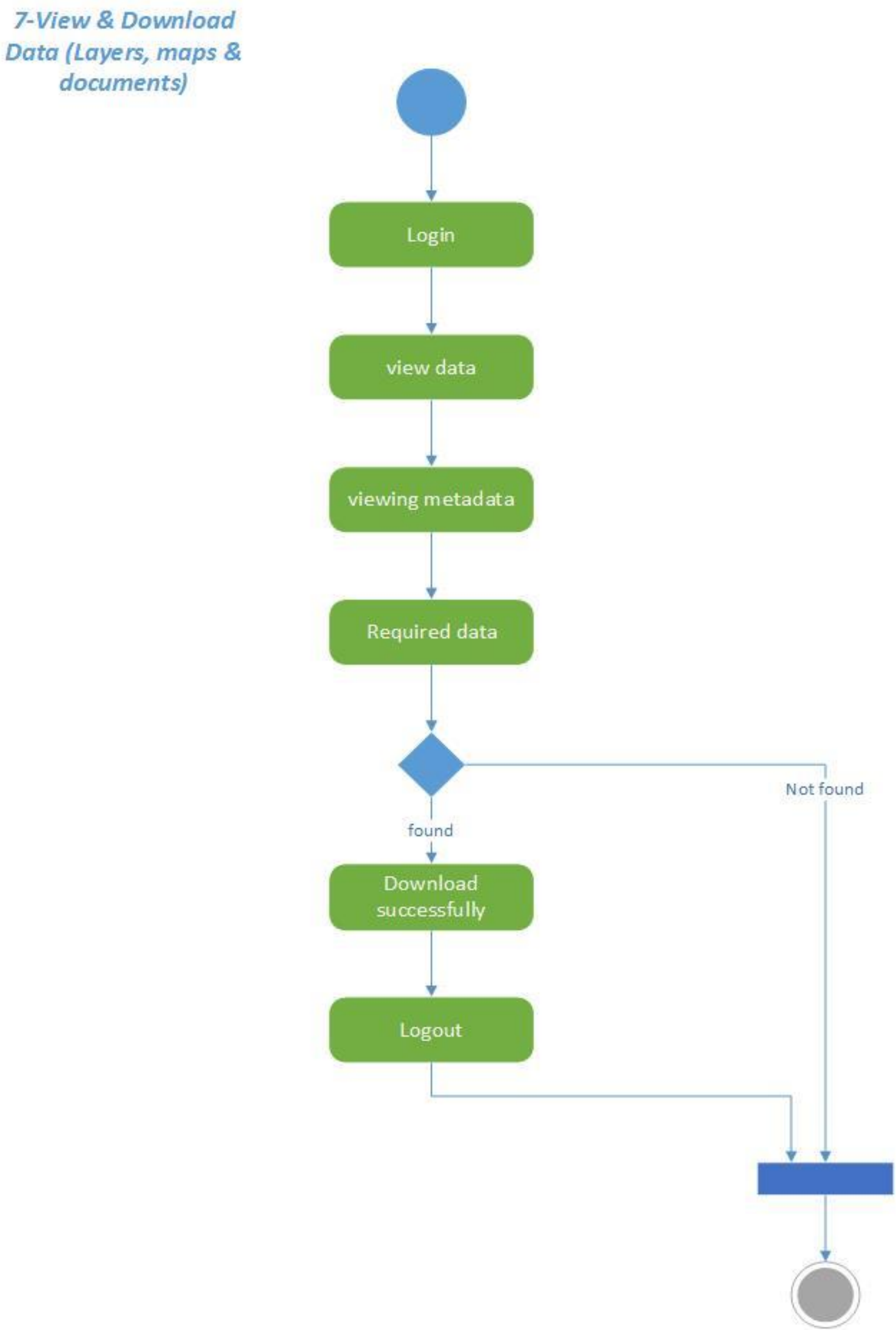

Figure 22: View and Download Layers, Maps and Documents

2.1.7 Sequence diagrams

A sequence diagram represents the sequence of messages or activities exchanged by the set of objects (and probably an actor) performing a certain task.

The objects are often arranged from left to right across the diagram at the top, an actor starts the interaction is often shown on the extreme left. The vertical dimension of the diagram shows time. The top of the diagram is the starting point and as the time passes downwards towards the bottom of the diagram. A vertical dashed line is called a lifeline which is attached to each object or actor in the diagram. The lifeline becomes a box which is called an activation box during the time in which the object is performing computations. The object is said to have live activation during these times.

The sequence diagram consist of Objects, Actors, Lifelines and the time attribute which varies in vertical direction. (Laganiere, 2005)

The sequence diagrams we made for IGISpatia are as follows:

- Sign-Up
- Search data
- Add User
- Delete User
- Create Map
- Upload Data (Layers & documents)
- View & Download Data (Layers, maps & documents)

The diagrams are given below:

1-Login

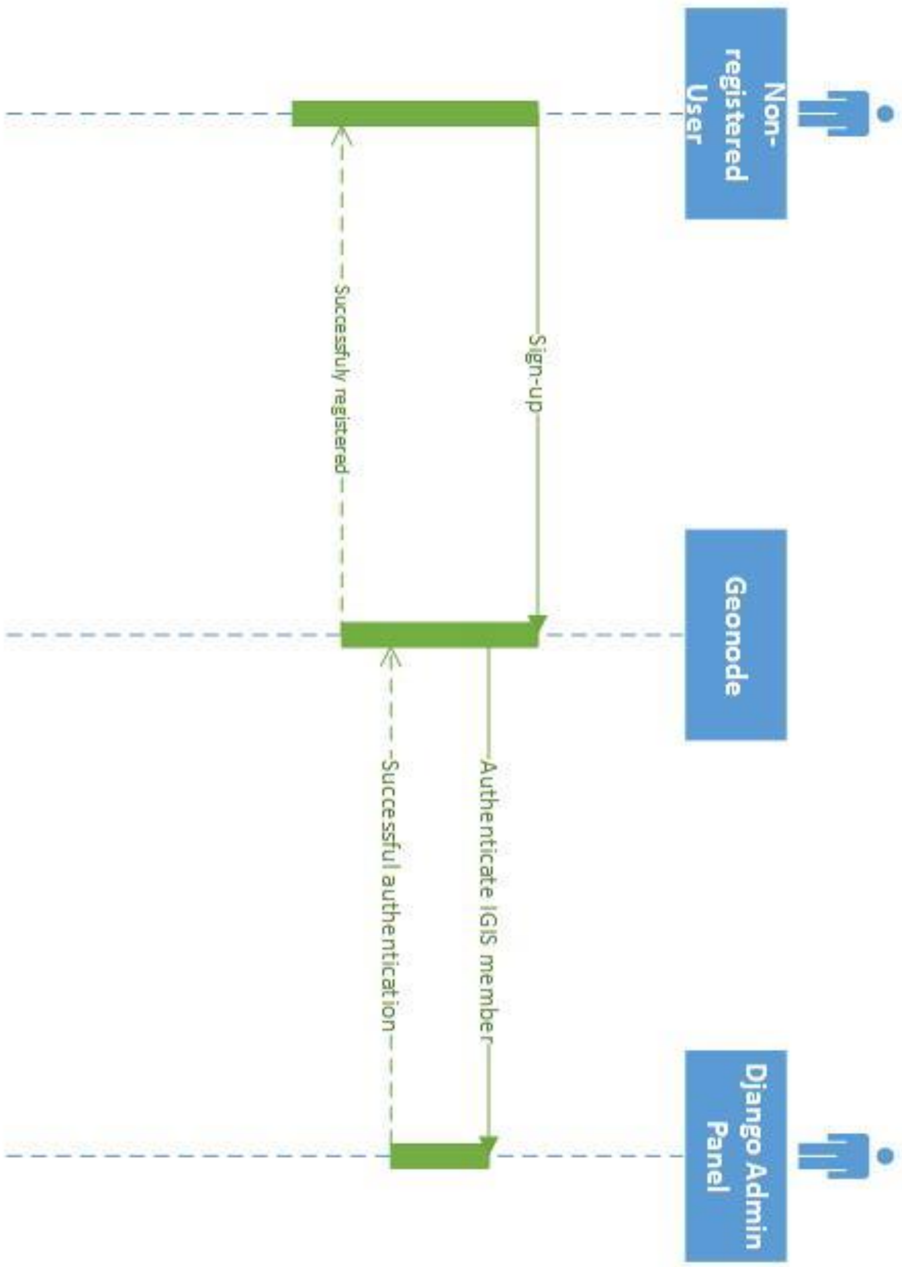

Figure 23: Login

2-Search data

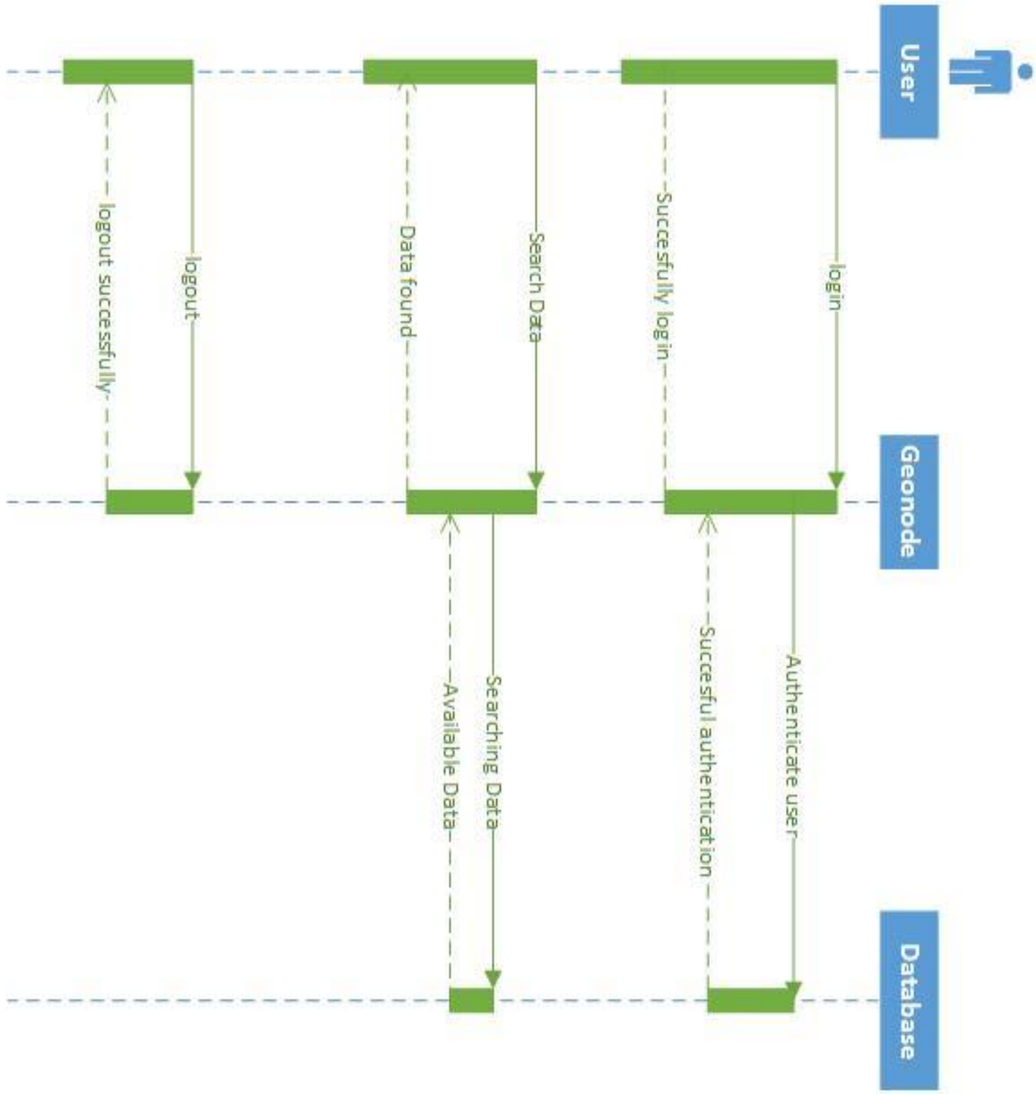

Figure 24: Search Data

3-Add user

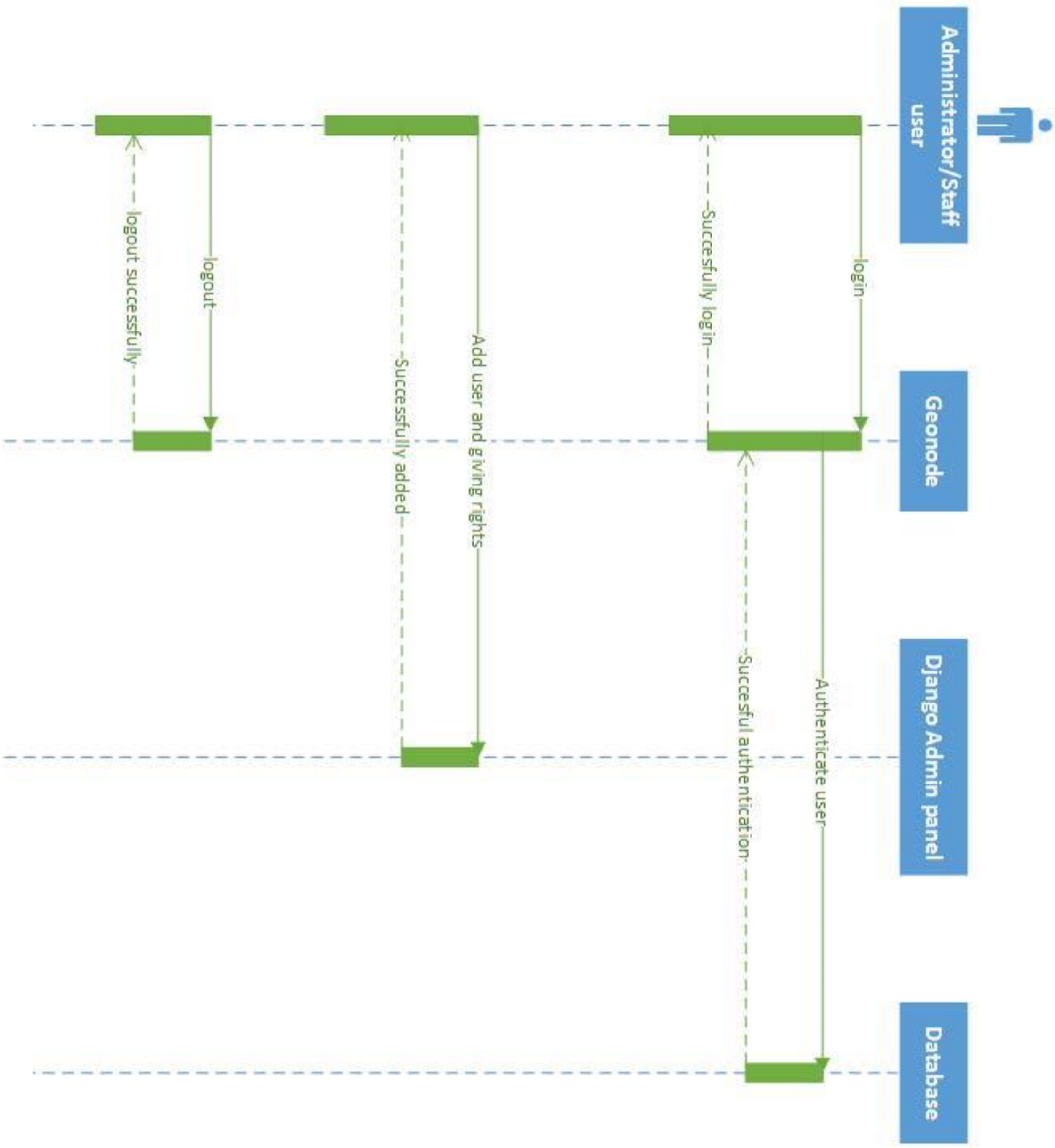

Figure 25: Add User

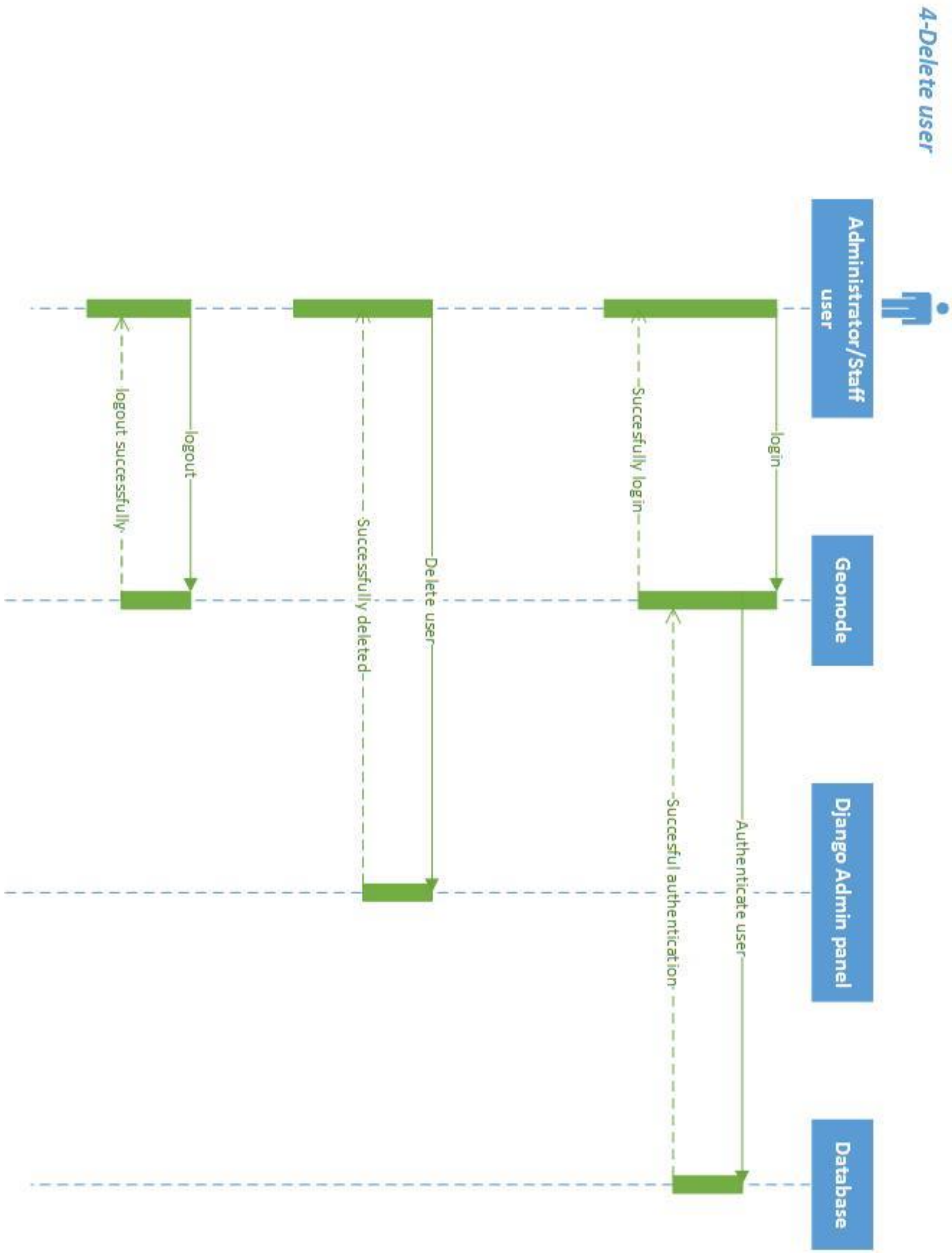

Figure 26: Delete User

5-Create Map

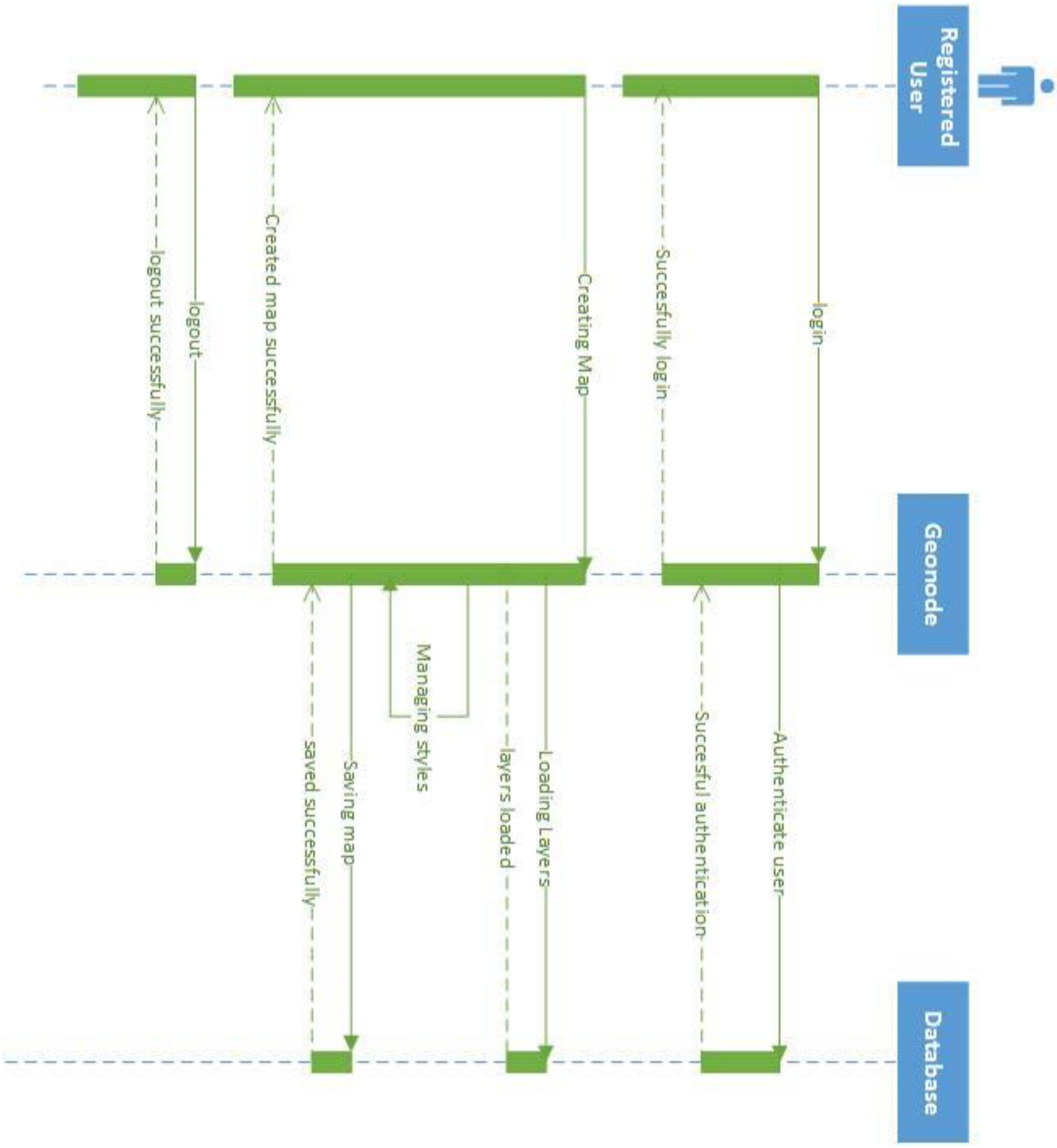

Figure 27: Create Map

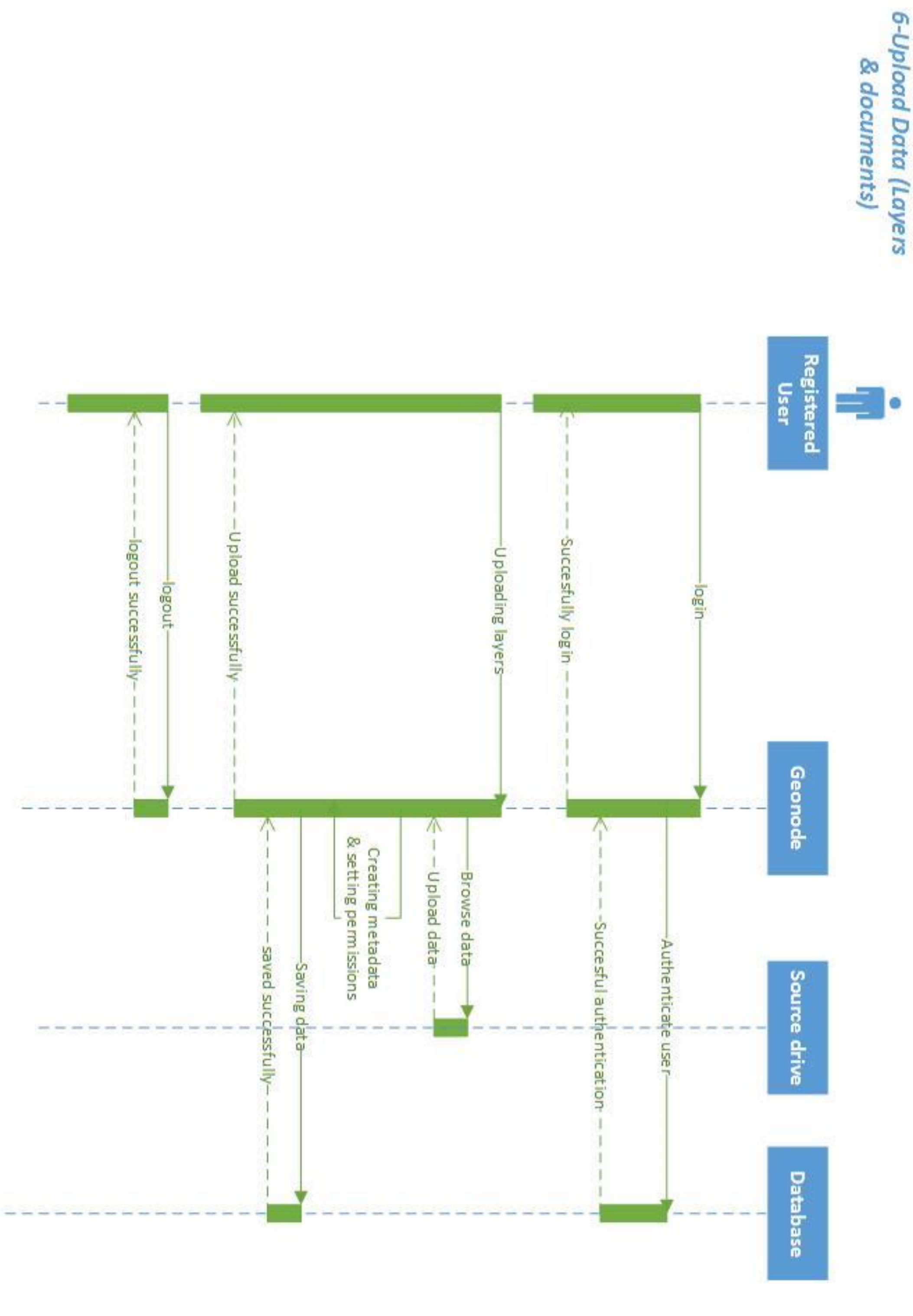

Figure 28: Upload Data Layers and Documents

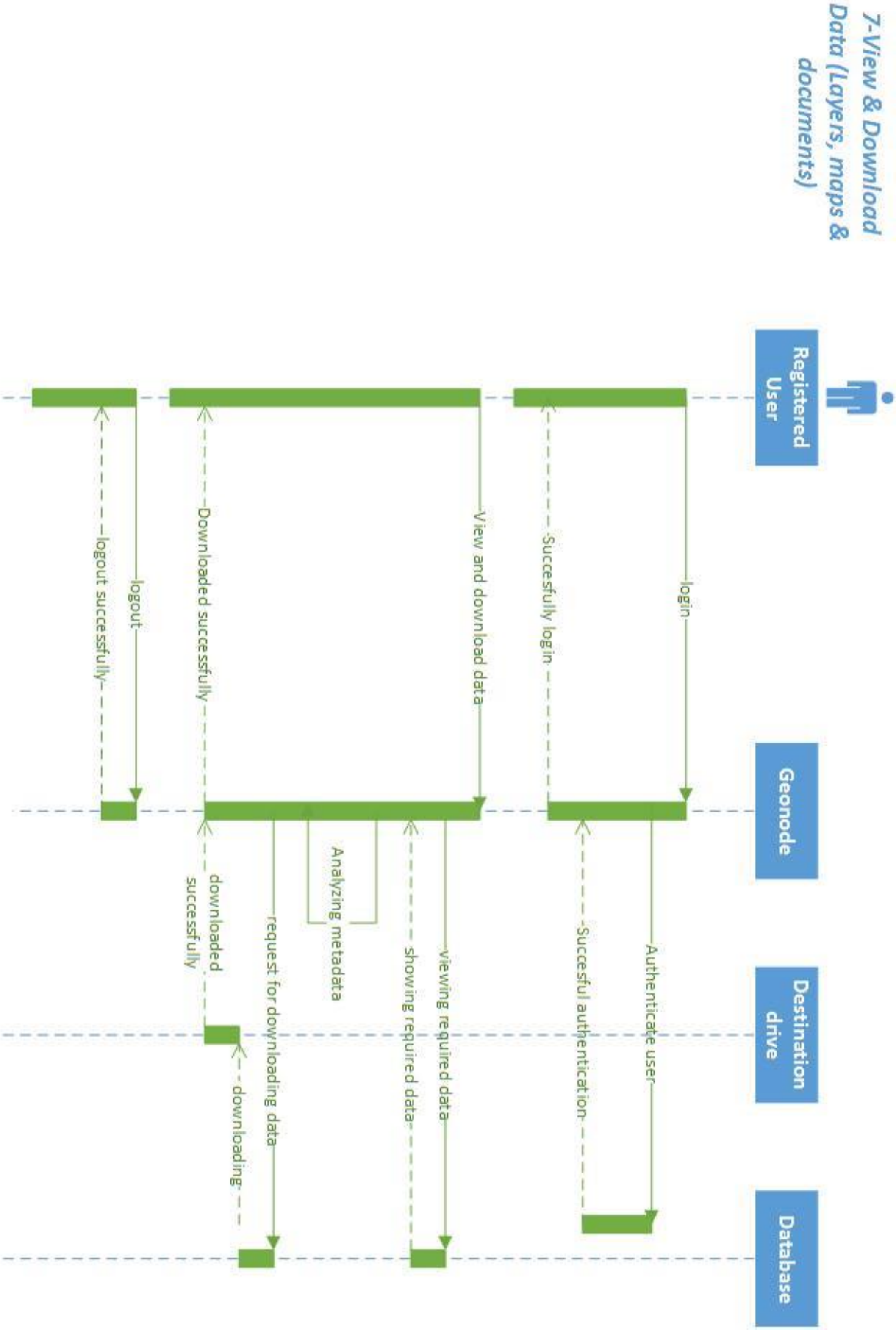

Figure 29: View and Download Layers, Maps and Documents

2.2 INTEGRATION AND IMPLEMENTATION

2.2.1 Setting the environment

After describing the overall design and architecture of the web application, it is required to practically implement the design strategy. To develop and deploy an application which is supposed to run on web and also using some desktop components, it is necessary that the overall architecture of all components is defined first. Later, one shall be able to implement each of the components and integrate then together to work as a complete web application. If we analyze on our requirements and objectives, we can find the desired end product and guess about what it should be like regarding its functionality and look and feel. In this case, at first it is needed to look at the overall architecture of our application. When we analyze our product, the overall architecture of our product is found and explained as below.

2.2.2 The System Architecture

The web architecture refers to the components which needs to be installed and deployed locally on the system and need to be integrated in order to work properly and in synchronization with what is desired to be as the final product. This web application follows the normal 3-tier web application architecture with an additional component of GIS. The basic components if this web architecture are described below.

- The presentation tier consists of the end user interface or front end which allows the user to interact with the web application. The user can input data or send different kinds of requests which are sent to the server via other tiers and get back with a response which again reaches the presentation tier. This serves as the result of that query which was invoked by the application user. Referring to our web application, the presentation layer is designed in HTML. CSS was used to apply styles to the HTML generated content. Further, JavaScript along with jQuery (JavaScript library) was used to make the content dynamic and user driven. It enables the end-user interface to be very interactive.
- The middle tier or the logical tier is the most important stage of this architecture. The reason is that all the activities that are taking place between the first and the last tier go through this tier. This tier is responsible for taking requests to the

database server where the query gets executed and a response is generated. This response is then sent back to the presentation tier again through this logical tier. In our web application, the logical tier consists of Apache2 as the default web server of Linux. It is responsible for moving the HTML documents along with all of its contents which may be textual data, image data, videos, HTML objects or geospatial data. The basic Django Web Application runs on this server. Further, MOD_WSGI acts as the web application server for the specific python functionality and holding its execution over the web. MOD_WSGI provides high performance for running python based applications. For the purpose of holding geospatial data in its native form and allowing the spatial queries to be executed on them, we definitely need a Web GIS Mapping Server which in our case is the GeoServer. GeoServer is an open-source application for sharing and hosting of geospatial data and is widely used in Web Mapping Applications and Web applications having other Location Based Services. It can pick up the geospatial data from any source database. It provides a high level interoperability. Tomcat 7 acts as a web application server for the hosting of GeoServer itself which is basically a Java based web application. So, in this case, we can see that we have two application servers over which two web applications are deployed both are based on two different technologies. The combination of Java based application and a python based application proves the final product to be at a higher level of interoperability. Apart from that, there is a Catalogue Service for Web (CSW) required in geo-portals which keeps all the metadata catalogs and stores them and keep a record of each type of metadata. PYCSW is a python based catalog service for web. It provides for the publishing and discovery of geospatial data. It is used to provide standards based metadata and cataloging service in spatial data infrastructures.

- The Database tier is the final and last tier in this web architecture. It contains the Database server which in case of our web application is the PostgreSQL with its spatial extension PostGIS enabled. It provides the storage and retrieval of geospatial data as well as non-spatial data. The request generated by the user at the presentation tier is ultimately reached at the database tier, where the relevant data is queried and the resultant data is returned back. The GeoServer publishes web services using the geospatial data contained by PostgreSQL. We can also say that a layer of GeoServer acts on top of the database to handle the behavior of geospatial data.

- The Simple Mail Transfer Protocol (SMTP) is used to set up a mailing server on a system. In our web application, we have used the Ubuntu's default Postfix SMTP server. Since our application needed to send emails on different scenarios, therefore a system's own SMTP server was deployed to handle this functionality instead of using any online SMTP server.

2.2.3 Security and Permissions

The web portal or the web application provides different levels of security and user permissions. It provides the facility to restrict the access to the geospatial data, non-spatial data, maps, documents, metadata and even the web pages. This functionality is added for providing better security over the data contained by the geo-portal. These permissions and access rights are actually based on the type of user who is viewing the website. Therefore, in order to better understand the accessibility and restrictions, we first need to have a look on the users and their access rights which are defined on this geo-portal.

2.2.3.1 Users

There are two main types of users which are as follows.

- Unregistered users (anonymous)
- Registered users

Unregistered users are the ones who have not signed themselves up on the portal. They are anonymous visitors of the site who have not uploaded any data on the site or created any map. When a user signs-up and is validated and registered in the database, it becomes a registered user. Registered users may be one of the Staff, Active or Super user. A super user is created through the Linux terminal with all the access and grants. A super user created through the terminal has the status staff and active as well. Such a user has all the permissions without explicitly assigning them to it. A super user hence acts as an administrator who knows every information and has access to everything.

A staff user has the access to the Django Admin Interface. The Django Admin Interface allows the management of website and its resources from an administrator's point of view. An active user is a normal user which is meant to be the end user who will be using the web portal for accessing geospatial data and uploading and sharing with others as well as downloading and using the geospatial web services. If an administrator needs to delete an active user, it does not necessarily need to be deleted. Instead, an administrator can only unset its active status and it will not show up anymore.

For creating a new user, there are generally three options. Firstly, a user can be created using the Linux terminal interface. It is normally used to create super users. Secondly, a user can be created using the Django Admin Panel. Such cases usually occur when an administrator or staff wants to create any user from the user interface of the web application. Using this Admin Panel, one can also change the current status or permissions of any existing user. The third option is that a user goes for the sign-up process from the ‘Register’ option in the geo-portal. It will create a normal active user. The following snapshots show the process of creating all the three types of users.

Creating a super user using the terminal:

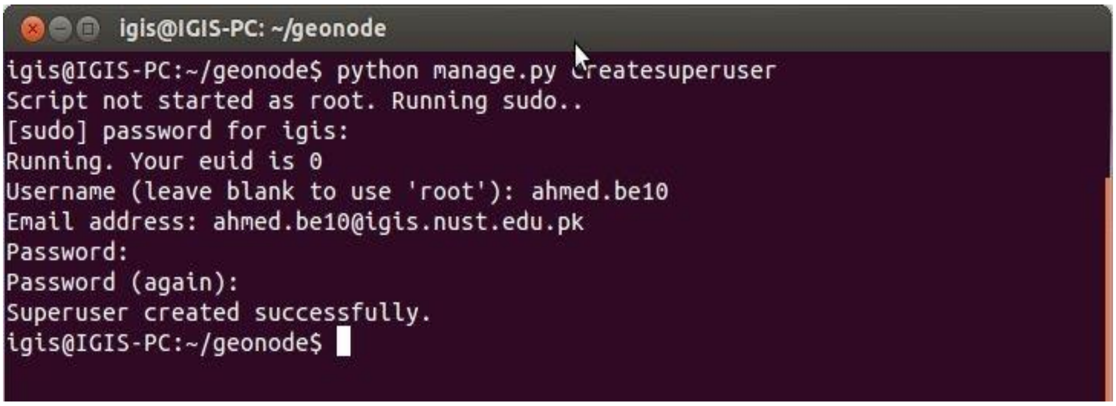

Figure 30: SuperUser

Creating a user using the Django Admin Interface:

Figure 31: Creating User using Admin Panel

Creating an active user through the sign-up process:

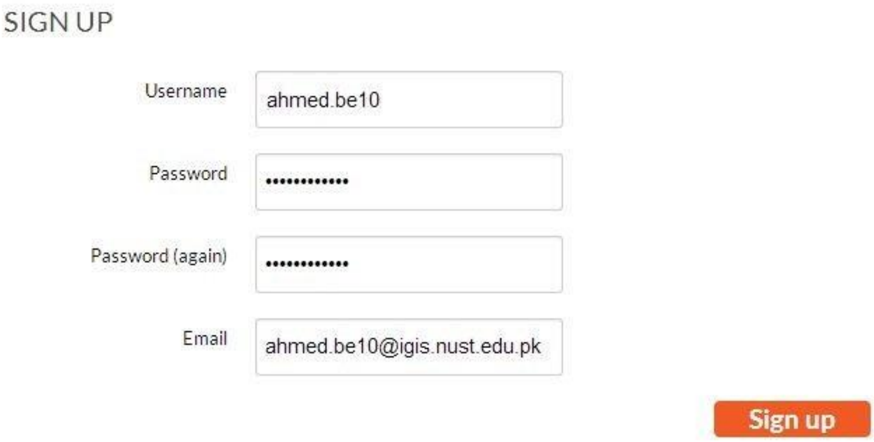

Figure 32: SignUp

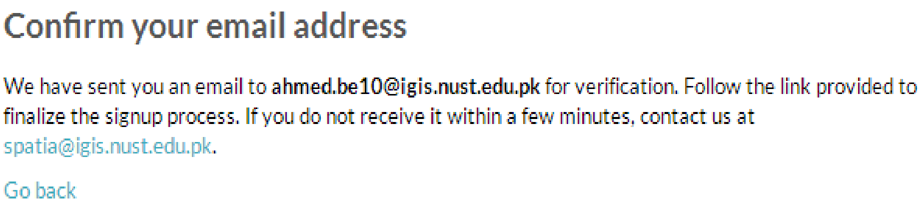

Figure 33: Confirmation

2.2.3.2 Geospatial Data Layers

A super user has the access to all the uploaded geospatial data layers in the geo-portal. But, a super user can set the permissions associated with any data layer to limit the access of that particular layer to the users. The IGISpatia portal has the following set of possible permissions.

- Permissions on who can view and download the spatial data layer
- Permissions on who can edit the spatial data layer
- Permissions on who can manage and edit the spatial data layer.

For allowing permissions to the users on these data layers, there are two options. Either set the permission to 'Any registered user' or set the permission to only a certain user or a certain group of users by listing their names. The following snapshot show the possible permission options.

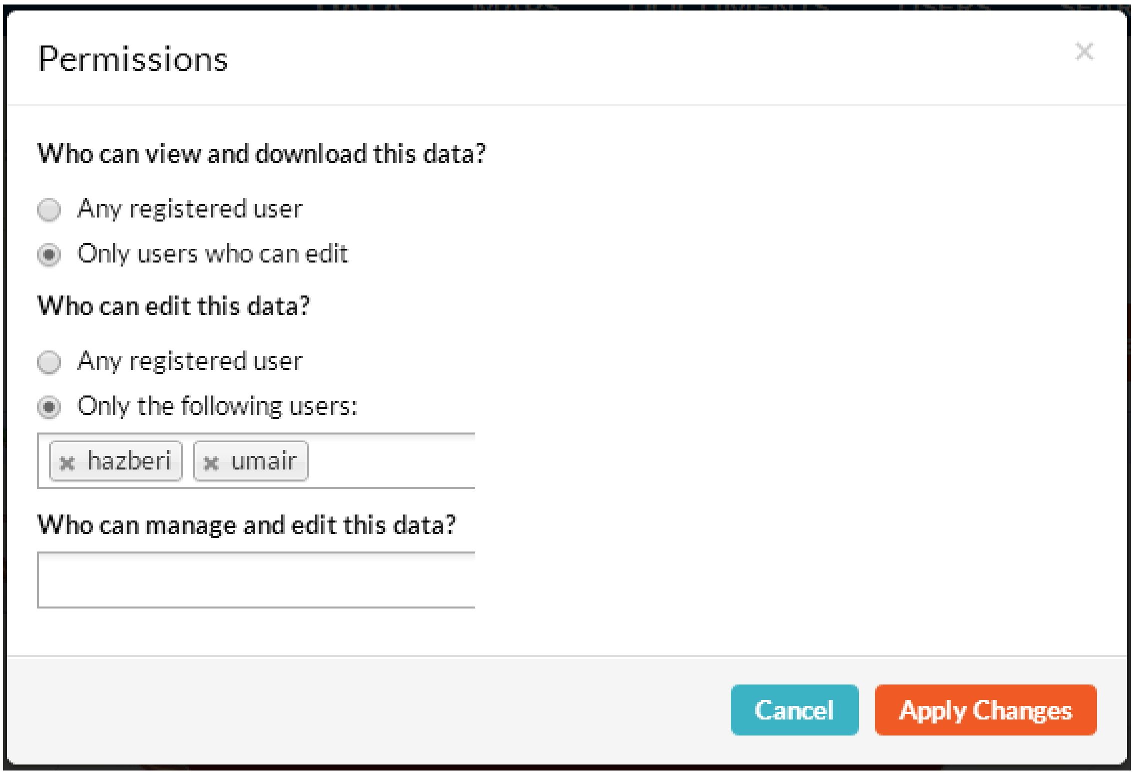

Figure 34: Permissions

When a certain user is allowed the permission to edit a data layer, the user actually gets the permission for following actions.

- Editing the metadata of that spatial data layer
- Editing the styles associated with that data layer

- Manage the styles of that data layer
- Replace the data layer with another one
- Remove the data layer

The following snapshot shows these permissions assigned to a user who gets the permission to edit a spatial data layer.

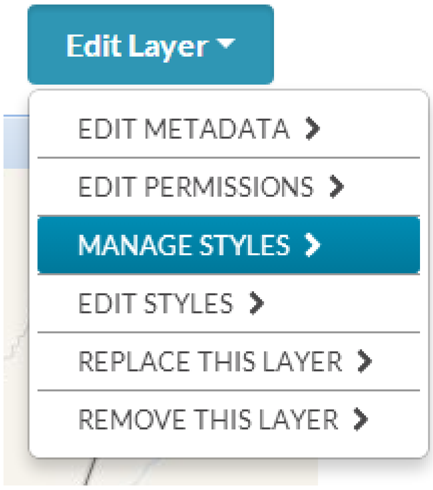

Figure 35: Styles

When the user clicks on the ‘Edit Metadata’ section, there is a list of metadata fields which can be changed by the user. These metadata fields are as follows.

- Owner
- Title
- Date
- Data type
- Edition
- Abstract
- Purpose
- Maintenance frequency
- Keywords region
- Restrictions
- Restrictions other
- Language
- Category
- Spatial representation type
- Temporal extent start

- Temporal extent end
- Supplemental information
- Distributed URL
- Distribution description
- Data quality statement
- Keywords
- Point of contact
- Metadata author
- Attributes

This snapshot shows the some of the metadata fields which the user is allowed to change them.

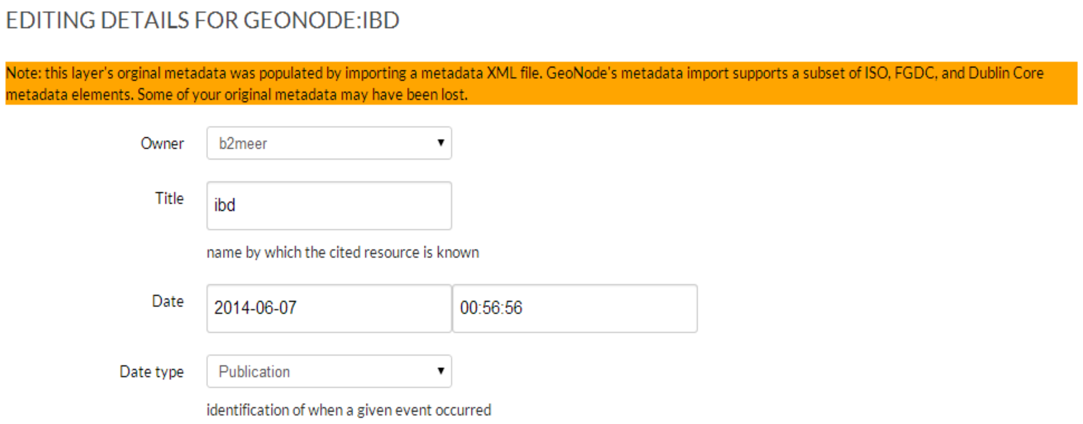

Figure 36: Metadata Fields

2.2.3.3 Maps

The set of permissions and the options available for the users for maps are the same as was for the spatial data layers. But, for a user who is allowed to 'Edit the Map' has fewer set of options to edit the map. These options are as follows.

- Edit map metadata
- Set map thumbnail
- Remove the map

The following snapshot depicts the same.

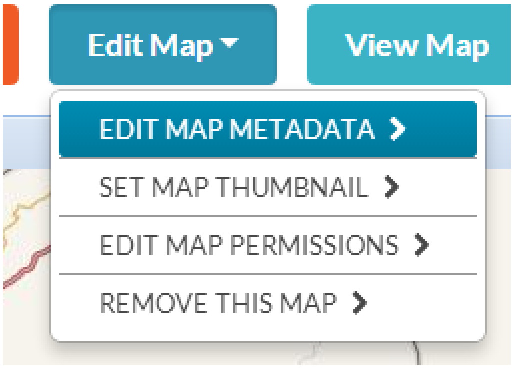

Figure 37: Edit Metadata

The 'Edit Map Metadata' section is the same as that of spatial data layers, just that it has two more options available which are as follows.

- Metadata XML
- Thumbnail

2.2.3.4 Documents

The set of permissions and user options are again the same as for the spatial data layers or maps. There is also an option to 'Edit Metadata'. Further, the option to replace or remove the document is also available.

2.2.4 Configuration

After the successful installations of individual components and their integration, the next step is to provide proper settings and configuration to all the files and paths so that everything works in a systematic way. For the purpose of defining settings, the Django application contains a file named 'settings.py' and another file named 'local_settings.py' in the installed application folder. The path to these files may look like as follows.

/home/IGISpatia/IGISpatia/settings.py

/home/IGISpatia/IGISpatia/local_settings.py

The file *settings.py* contains all the required settings related to the web application and it is used for deployment purposes. The file *local_settings.py* overrides the *settings.py* and it contains fewer information as compared to *settings.py*. Both these files contain configurations about the following main settings.

- GeoServer settings

- Database configuration
- SMTP settings
- Data size settings
- Domain and hosts
- Specific content on web pages (like Social buttons, Registration open or not etc.)
- Catalogue service settings
- Base layers for creating maps
- Installed Django Apps
- Document types allowed
- Debug mode
- Time zone
- Various Boolean variables
- Path to static files and paths to various other file types
- Metadata formats
- Vector data formats
- Raster data formats
- URL to the deployed website
- Deployment settings
- PYCSW settings
- And many other settings and variables

Following is the actual python code shared from the file *settings.py* of IGISpatia. One can go through it and have an idea as to how the configurations related to the targeted application component.

```
# Django settings for the project.
import os

#
# General Django development settings
#

# Defines the directory that contains the settings file as the PROJECT_ROOT
# It is used for relative settings elsewhere.
PROJECT_ROOT = os.path.abspath(os.path.dirname(__file__))

# Setting debug to true makes Django serve static media and
```

```
# present pretty error pages.
DEBUG = TEMPLATE_DEBUG = False

# Set to True to load non-minified versions of (static) client dependencies
# Requires to set-up Node and tools that are required for static development
# otherwise it will raise errors for the missing non-minified dependencies
DEBUG_STATIC = False

# This is needed for integration tests, they require
# geonode to be listening for GeoServer auth requests.
os.environ['DJANGO_LIVE_TEST_SERVER_ADDRESS'] = 'localhost:8000'

# Defines settings for development
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(PROJECT_ROOT, 'development.db'),
    },
    # vector datastore for uploads
    #'datastore' : {
    #    'ENGINE': 'django.contrib.gis.db.backends.postgis',
    #    'NAME': '',
    #    'USER' : '',
    #    'PASSWORD' : '',
    #    'HOST' : '',
    #    'PORT' : '',
    #}
}

# Local time zone for this installation. Choices can be found here:
# http://en.wikipedia.org/wiki/List_of_tz_zones_by_name
# although not all choices may be available on all operating systems.
# If running in a Windows environment this must be set to the same as your
# system time zone.
TIME_ZONE = 'Asia/Karachi'

# Language code for this installation. All choices can be found here:
# http://www.i18nguy.com/unicode/language-identifiers.html
LANGUAGE_CODE = 'en'

LANGUAGES = (
    ('en', 'English'),
    ('es', 'Español'),
    ('it', 'Italiano'),
    ('fr', 'Français'),
    ('de', 'Deutsch'),
    ('el', 'Ελληνικά'),
    ('id', 'Bahasa Indonesia'),
    ('zh-cn', '中文'),
    ('ja', '日本語'),
    ('fa', 'Persian'),
    ('pt', 'Portuguese'),
    ('ru', 'Russian'),
    ('vi', 'Vietnamese'),
    #('fil', 'Filipino'),

)
```

```
# If you set this to False, Django will make some optimizations so as not
# to load the internationalization machinery.
USE_I18N = True

# Absolute path to the directory that holds media.
# Example: "/home/media/media.lawrence.com/"
MEDIA_ROOT = os.path.join(PROJECT_ROOT, "uploaded")

# URL that handles the media served from MEDIA_ROOT. Make sure to use a
# trailing slash if there is a path component (optional in other cases).
# Examples: "http://media.lawrence.com", "http://example.com/media/"
MEDIA_URL = "/uploaded/"

# Absolute path to the directory that holds static files like app media.
# Example: "/home/media/media.lawrence.com/apps/"
STATIC_ROOT = os.path.join(PROJECT_ROOT, "static_root")

# URL that handles the static files like app media.
# Example: "http://media.lawrence.com"
STATIC_URL = "/static/"

# Additional directories which hold static files
STATICFILES_DIRS = [
    os.path.join(PROJECT_ROOT, "static"),
]

# List of finder classes that know how to find static files in
# various locations.
STATICFILES_FINDERS = (
    'django.contrib.staticfiles.finders.FileSystemFinder',
    'django.contrib.staticfiles.finders.AppDirectoriesFinder',
#    'django.contrib.staticfiles.finders.DefaultStorageFinder',
)

# Note that Django automatically includes the "templates" dir in all the
# INSTALLED_APPS, se there is no need to add maps/templates or admin/templates
TEMPLATE_DIRS = (
    os.path.join(PROJECT_ROOT, "templates"),
)

# Location of translation files
LOCALE_PATHS = (
    os.path.join(PROJECT_ROOT, "locale"),
)

# Make this unique, and don't share it with anybody.
SECRET_KEY = 'myv-y4#7j-d*p-__@j#*3z@!y24fz8%^z2v6atuy4bo9vqr1_a'

# Location of url mappings
ROOT_URLCONF = 'geonode.urls'

# Site id in the Django sites framework
SITE_ID = 5

# Login and logout urls override
LOGIN_URL = '/account/login/'
LOGOUT_URL = '/account/logout/'
```

```
# Documents application
ALLOWED_DOCUMENT_TYPES = [
    'doc', 'docx','gif', 'jpg', 'jpeg', 'ods', 'odt', 'pdf', 'png', 'ppt',
    'rar', 'tif', 'tiff', 'txt', 'xls', 'xlsx', 'xml', 'zip',
]
MAX_DOCUMENT_SIZE = 25 # MB


EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'localhost'
EMAIL_HOST_USER = ''
EMAIL_HOST_PASSWORD = ''
EMAIL_PORT = 587
EMAIL_USE_TLS = True
DEFAULT_FROM_EMAIL = 'IGISpatia <noreply@spatia.igis.nust.edu.pk>'

ACCOUNT_EMAIL_UNIQUE = True
ACCOUNT_EMAIL_CONFIRMATION_REQUIRED = True

INSTALLED_APPS = (

    # Apps bundled with Django
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'django.contrib.admin',
    'django.contrib.sitemaps',
    'django.contrib.staticfiles',
    'django.contrib.messages',
    'django.contrib.humanize',

    # Third party apps

    # Utility
    'pagination',
    'taggit',
    'taggit_templatetags',
    'south',
    'friendlytagloader',
    'geoexplorer',
    'django_extensions',

    # Theme
    "pinax_theme_bootstrap_account",
    "pinax_theme_bootstrap",
    'django_forms_bootstrap',

    # Social
    'account',
    'avatar',
    'dialogos',
    'agon_ratings',
    'notification',
    'announcements',
    'actstream',
    'user_messages',
```

```
    # GeoNode internal apps
    'geonode.people',
    'geonode.base',
    'geonode.layers',
    'geonode.upload',
    'geonode.maps',
    'geonode.proxy',
    'geonode.security',
    'geonode.search',
    'geonode.social',
    'geonode.catalogue',
    'geonode.documents',
)

LOGGING = {
    'version': 1,
    'disable_existing_loggers': True,
    'formatters': {
        'verbose': {
            'format':  '%(levelname)s  %(asctime)s  %(module)s  %(process)d
%(thread)d %(message)s'
        },
        'simple': {
            'format': '%(message)s',        },
    },
    'filters': {
        'require_debug_false': {
            '()': 'django.utils.log.RequireDebugFalse'
     }
    },
    'handlers': {
        'null': {
            'level':'ERROR',
            'class':'django.utils.log.NullHandler',
        },
        'console':{
            'level':'ERROR',
            'class':'logging.StreamHandler',
            'formatter': 'simple'
        },
        'mail_admins': {
            'level': 'ERROR',
            'filters': ['require_debug_false'],
            'class': 'django.utils.log.AdminEmailHandler',
        }
    },
    "loggers": {
        "django": {
            "handlers": ["console"],
            "level": "ERROR",
        },
        "geonode": {
            "handlers": ["console"],
            "level": "ERROR",
        },

        "gsconfig.catalog": {
            "handlers": ["console"],
```

```
            "level": "ERROR",
        },
        "owslib": {
            "handlers": ["console"],
            "level": "ERROR",
        },
        "pycsw": {
            "handlers": ["console"],
            "level": "ERROR",
        },
        'south': {
            "handlers": ["console"],
            "level": "ERROR",
        },
    },
}

#
# Customizations to built in Django settings required by GeoNode
#


TEMPLATE_CONTEXT_PROCESSORS = (
    'django.contrib.auth.context_processors.auth',
    'django.core.context_processors.debug',
    'django.core.context_processors.i18n',
    "django.core.context_processors.tz",
    'django.core.context_processors.media',
    "django.core.context_processors.static",
    'django.core.context_processors.request',
    'django.contrib.messages.context_processors.messages',
    'account.context_processors.account',
    'pinax_theme_bootstrap_account.context_processors.theme',
    # The context processor below adds things like SITEURL
    # and GEOSERVER_BASE_URL to all pages that use a RequestContext
    'geonode.context_processors.resource_urls',
)

MIDDLEWARE_CLASSES = (
    'django.middleware.common.CommonMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    # The setting below makes it possible to serve different languages per
    # user depending on things like headers in HTTP requests.
    'django.middleware.locale.LocaleMiddleware',
    'pagination.middleware.PaginationMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    # This middleware allows to print private layers for the users that have
    # the permissions to view them.
    # It sets temporary the involved layers as public before restoring the
permissions.
    # Beware that for few seconds the involved layers are public there could
be risks.
    #'geonode.middleware.PrintProxyMiddleware',
)
```

```
# Replacement of default authentication backend in order to support
# permissions per object.
AUTHENTICATION_BACKENDS = ('geonode.security.auth.GranularBackend',)

def get_user_url(u):
    return u.profile.get_absolute_url()


ABSOLUTE_URL_OVERRIDES = {
    'auth.user': get_user_url
}

# Redirects to home page after login
# FIXME(Ariel): I do not know why this setting is needed,
# it would be best to use the ?next= parameter
LOGIN_REDIRECT_URL = "/"

#
# Settings for default search size
#
DEFAULT_SEARCH_SIZE = 10


#
# Settings for third party apps
#

# Agon Ratings
AGON_RATINGS_CATEGORY_CHOICES = {
    "maps.Map": {
        "map": "How good is this map?"
    },
    "layers.Layer": {
        "layer": "How good is this layer?"
    },
    "documents.Document": {
        "document": "How good is this document?"
    }
}

# Activity Stream
ACTSTREAM_SETTINGS = {
    'MODELS': ('auth.user', 'layers.layer', 'maps.map', 'dialogos.comment',
'documents.document'),
    'FETCH_RELATIONS': True,
    'USE_PREFETCH': False,
    'USE_JSONFIELD': True,
    'GFK_FETCH_DEPTH': 1,
}

# For South migrations
SOUTH_MIGRATION_MODULES = {
    'avatar': 'geonode.migrations.avatar',
}
SOUTH_TESTS_MIGRATE=False

# Settings for Social Apps
AUTH_PROFILE_MODULE = 'people.Profile'
```

```
REGISTRATION_OPEN = True

# Email for users to contact admins.
THEME_ACCOUNT_CONTACT_EMAIL = 'spatia@igis.nust.edu.pk'

#
# Test Settings
#

# Setting a custom test runner to avoid running the tests for
# some problematic 3rd party apps
TEST_RUNNER = 'django_nose.NoseTestSuiteRunner'

# Arguments for the test runner
NOSE_ARGS = [
      '--nocapture',
      '--detailed-errors',
      ]

#
# IGISpatia specific settings
#

SITEURL = "http://spatia.igis.nust.edu.pk/"

# Default TopicCategory to be used for resources. Use the slug field here
DEFAULT_TOPICCATEGORY = 'location'

# Topic Categories list should not be modified (they are ISO). In case you
# absolutely need it set to True this variable
MODIFY_TOPICCATEGORY = False

MISSING_THUMBNAIL = 'geonode/img/missing_thumb.png'

# Search Snippet Cache Time in Seconds
CACHE_TIME=0

# OGC (WMS/WFS/WCS) Server Settings
OGC_SERVER = {
    'default' : {
        'BACKEND' : 'geonode.geoserver',
        'LOCATION' : 'http://spatia.igis.nust.edu.pk:8080/geoserver/',
        # PUBLIC_LOCATION needs to be kept like this because in dev mode
        # the proxy won't work and the integration tests will fail
        # the entire block has to be overridden in the local_settings
        'PUBLIC_LOCATION'                                                :
'http://spatia.igis.nust.edu.pk:8080/geoserver/',
        'USER' : 'admin',
        'PASSWORD' : 'geoserver',
        'MAPFISH_PRINT_ENABLED' : True,
        'PRINTNG_ENABLED' : True,
        'GEONODE_SECURITY_ENABLED' : True,
        'GEOGIT_ENABLED' : False,
        'WMST_ENABLED' : False,
        'BACKEND_WRITE_ENABLED': True,
        'WPS_ENABLED' : True,
        # Set to name of database in DATABASES dictionary to enable
        'DATASTORE': '', #'datastore',
```

```
        'TIMEOUT': 10  # number of seconds to allow for HTTP requests
    }
}

# Uploader Settings
UPLOADER = {
    'BACKEND' : 'geonode.rest',
    'OPTIONS' : {
        'TIME_ENABLED': False,
        'GEOGIT_ENABLED': False,
    }
}

# CSW settings
CATALOGUE = {
    'default': {
        # The underlying CSW implementation
        # default is pycsw in local mode (tied directly to GeoNode Django
DB)
        'ENGINE': 'geonode.catalogue.backends.pycsw_local',
        # pycsw in non-local mode
        #'ENGINE': 'geonode.catalogue.backends.pycsw_http',
        # GeoNetwork opensource
        #'ENGINE': 'geonode.catalogue.backends.geonetwork',
        # deegree and others
        #'ENGINE': 'geonode.catalogue.backends.generic',

        # The FULLY QUALIFIED base url to the CSW instance for this GeoNode
        'URL': '%scatalogue/csw' % SITEURL,
        #'URL': 'http://localhost:8080/geonetwork/srv/en/csw',
        #'URL': 'http://localhost:8080/deegree-csw-demo-3.0.4/services',

        # login credentials (for GeoNetwork)
        'USER': 'admin',
        'PASSWORD': 'admin',
    }
}

# pycsw settings
PYCSW = {
    # pycsw configuration
    'CONFIGURATION': {
        'metadata:main': {
            'identification_title': 'GeoNode Catalogue',
            'identification_abstract': 'GeoNode is an open source platform
that facilitates the creation, sharing, and collaborative use of geospatial
data',
            'identification_keywords':
'sdi,catalogue,discovery,metadata,GeoNode',
            'identification_keywords_type': 'theme',
            'identification_fees': 'None',
            'identification_accessconstraints': 'None',
            'provider_name': 'Organization Name',
            'provider_url': SITEURL,
            'contact_name': 'Lastname, Firstname',
            'contact_position': 'Position Title',
            'contact_address': 'Mailing Address',
            'contact_city': 'City',
```

```
            'contact_stateorprovince': 'Administrative Area',
            'contact_postalcode': 'Zip or Postal Code',
            'contact_country': 'Country',
            'contact_phone': '+xx-xxx-xxx-xxxx',
            'contact_fax': '+xx-xxx-xxx-xxxx',
            'contact_email': 'Email Address',
            'contact_url': 'Contact URL',
            'contact_hours': 'Hours of Service',
            'contact_instructions': 'During hours of service. Off on
weekends.',
            'contact_role': 'pointOfContact',
        },
        'metadata:inspire': {
            'enabled': 'true',
            'languages_supported': 'eng,gre',
            'default_language': 'eng',
            'date': 'YYYY-MM-DD',
            'gemet_keywords': 'Utility and governmental services',
            'conformity_service': 'notEvaluated',
            'contact_name': 'Organization Name',
            'contact_email': 'Email Address',
            'temp_extent': 'YYYY-MM-DD/YYYY-MM-DD',
        }
    }
}

# GeoNode javascript client configuration

# Where should newly created maps be focused?
DEFAULT_MAP_CENTER = (0, 0)

# How tightly zoomed should newly created maps be?
# 0 = entire world;
# maximum zoom is between 12 and 15 (for Google Maps, coverage varies by
area)
DEFAULT_MAP_ZOOM = 0

MAP_BASELAYERS = [{
    "source": {
        "ptype": "gxp_wmscsource",
        "url": OGC_SERVER['default']['PUBLIC_LOCATION'] + "wms",
        "restUrl": "/gs/rest"
     }
  },{
    "source": {"ptype": "gxp_olsource"},
    "type":"OpenLayers.Layer",
    "args":["No background"],
    "visibility": False,
    "fixed": True,
    "group":"background"
  }, {
    "source": {"ptype": "gxp_osmsource"},
    "type":"OpenLayers.Layer.OSM",
    "name":"mapnik",
    "visibility": False,
    "fixed": True,
    "group":"background"
  }, {
```

```
    "source": {"ptype": "gxp_mapquestsource"},
    "name":"osm",
    "group":"background",
    "visibility": True
  }, {
    "source": {"ptype": "gxp_mapquestsource"},
    "name":"naip",
    "group":"background",
    "visibility": False
  }, {
    "source": {"ptype": "gxp_bingsource"},
    "name": "AerialWithLabels",
    "fixed": True,
    "visibility": False,
    "group":"background"
  },{
    "source": {"ptype": "gxp_mapboxsource"},
  }, {
    "source": {"ptype": "gxp_olsource"},
    "type":"OpenLayers.Layer.WMS",
    "group":"background",
    "visibility": False,
    "fixed": True,
    "args":[
      "bluemarble",
      "http://maps.opengeo.org/geowebcache/service/wms",
      {
        "layers":["bluemarble"],
        "format":"image/png",
        "tiled": True,
        "tilesOrigin": [-20037508.34, -20037508.34]
      },
      {"buffer": 0}
    ]

}]

LEAFLET_CONFIG = {
    'TILES_URL':
'http://{s}.tile2.opencyclemap.org/transport/{z}/{x}/{y}.png'
}

SOCIAL_BUTTONS = True

# Require users to authenticate before using Geonode
LOCKDOWN_GEONODE = False

# Add additional paths (as regular expressions) that don't require
authentication.
AUTH_EXEMPT_URLS = ()

if LOCKDOWN_GEONODE:
    MIDDLEWARE_CLASSES = MIDDLEWARE_CLASSES +
('geonode.security.middleware.LoginRequiredMiddleware',)


# A tuple of hosts the proxy can send requests to.
```

```
PROXY_ALLOWED_HOSTS                                                        =
('spatia.igis.nust.edu.pk','www.spatia.igis.nust.edu.pk','115.186.131.88','
localhost',)

# The proxy to use when making cross origin requests.
PROXY_URL = '/proxy/?url='


# Load more settings from a file called local_settings.py if it exists
try:
    from local_settings import *
except ImportError:
    pass

# Available download formats
DOWNLOAD_FORMATS_METADATA = [
    'Atom', 'DIF', 'Dublin Core', 'ebRIM', 'FGDC', 'TC211',
]
DOWNLOAD_FORMATS_VECTOR = [
    'JPEG', 'PDF', 'PNG', 'Zipped Shapefile', 'GML 2.0', 'GML 3.1.1', 'CSV',
    'Excel', 'GeoJSON', 'KML', 'View in Google Earth', 'Tiles',
]
DOWNLOAD_FORMATS_RASTER = [
    'JPEG', 'PDF', 'PNG', 'ArcGrid', 'GeoTIFF', 'Gtopo30', 'ImageMosaic',
'KML',
    'View in Google Earth', 'Tiles',
]
```

2.2.5 Deployment

Initially, the web application was setup locally on a system. When all the required components were installed and configured, then they were tested for their performance. This step was necessary to know whether every individual component is working perfectly or not. After ensuring that all the components that are integrated are working perfectly fine and in synchronization, the next step was to implement the same instance of the web application geo-portal over the World Wide Web. This refers to the actual deployment of the web application instance over a system which could then be made available over the web. For this purpose, a computer system with the following specifications was availed.

- Dell Branded system
- 12 GB of RAM
- 1 TB of Hard drive
- Intel core i5 processor
- Network card

After acquiring this CPU, the web application was locally installed on the system. The next step was to hand over this system to an internet service provider whose duty was to deploy it as a web server and assign a global IP address to this system. The global IP 115.186.131.88 was assigned to the system. Therefore, at that stage of deployment, the global instance of our web application was available at the URL *http://115.186.131.88* . After this, the next step was to assign a public and user friendly Domain Name Server (DNS) against this IP address. This step was also initiated and as a result we obtained a final DNS for our application which was *spatia.igis.nust.edu.pk.* Hence, then onwards, the geo-portal application was globally accessible at the URL *http://spatia.igis.nust.edu.pk*

Please note that in the previous section where the python code of the file *settings.py* is shared, the deployment settings are the latest and updated ones which means that they refer to the site through the URL *http://spatia.igis.nust.edu.pk* .

CHAPTER 3

RESULTS AND DISCUSSION

3.1 TESTING

Usability testing is an important step to check the efficiency of an application whether web or desktop. It also checks how much effective the application is. It also check whether it follow certain standards. For example to check for the color contrast and color convention, like to see that red color is used for the closing button etc.

In this study we are going to test the usability of a hosted geoportal "IGISpatia". A geoportal is single point access to geospatial data. In the usability of the IGISpatia we are going to test for the following features:

- Navigation between pages
- Color codes
- User friendliness

This testing was conducted by taking 5 users who had some experience of how to use geoportal. The first thing to test is the navigation between pages.

The first task is quite simple the users were asked to navigate between different pages of the application. Users performed the task and navigated between different pages with great ease.

Results of the first task show navigation of different pages

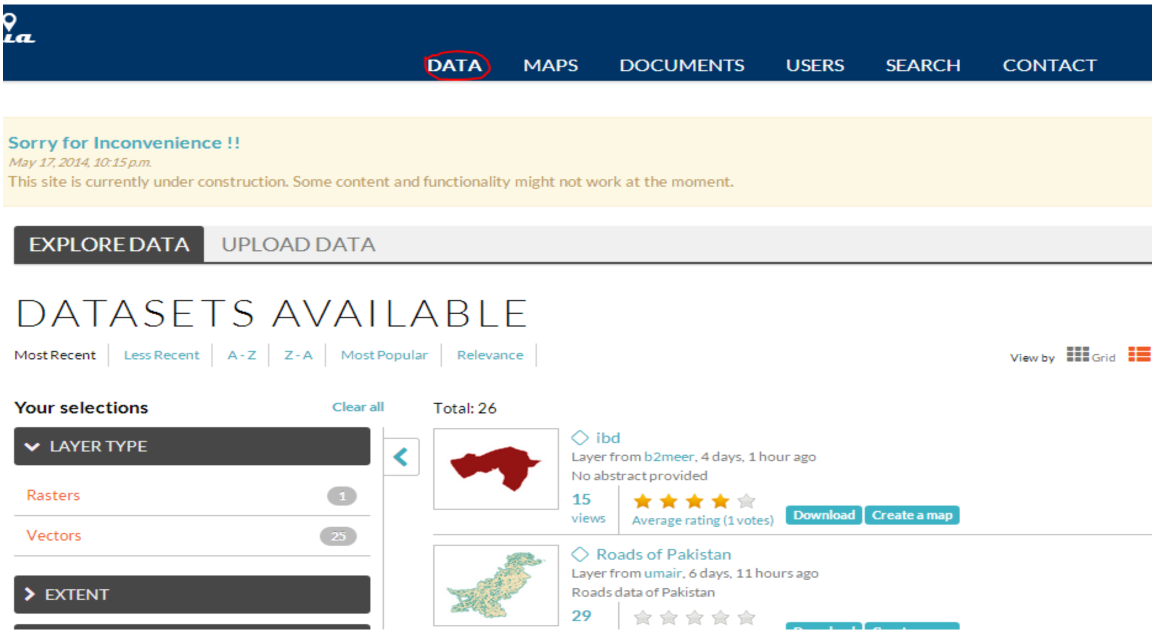

Figure 38: Explore Data

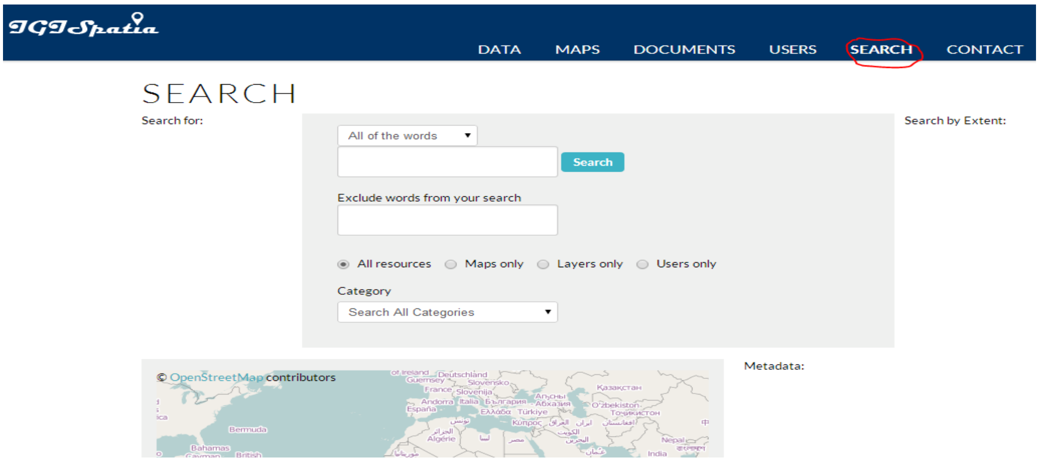

Figure 39: Spatial Search

The next task the user performed was to check for the color codes, the contrast, color appeal and the colors of different buttons whether it complies with the standards. According to the users response the color were appealing to eyes and there were no contrast issues and the color codes were according to the standards.

Results

The users were very satisfied with contrast matching of all the pages. The color of the button and background was quite clear.

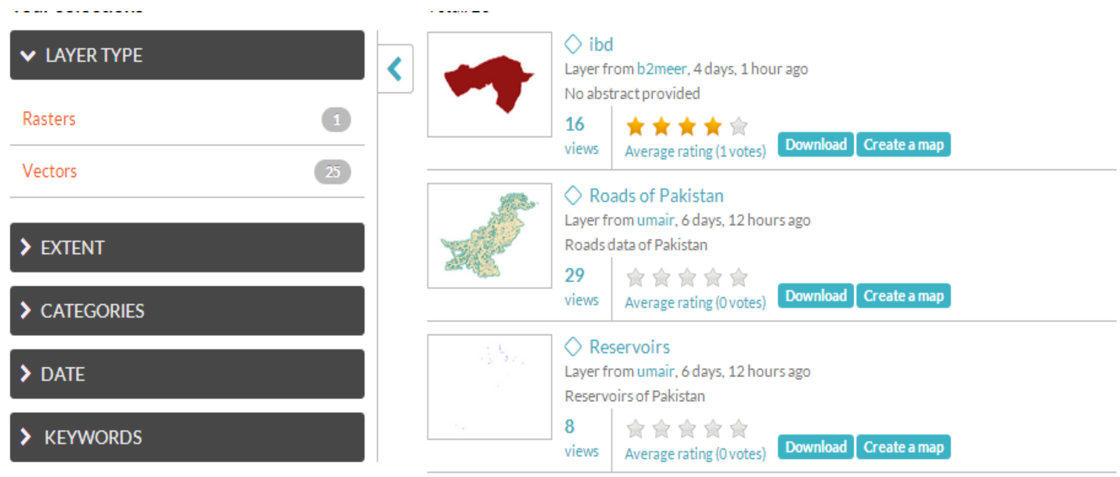

Figure 40: Layer Types

The logout button is quite clear from the background and follow convention.

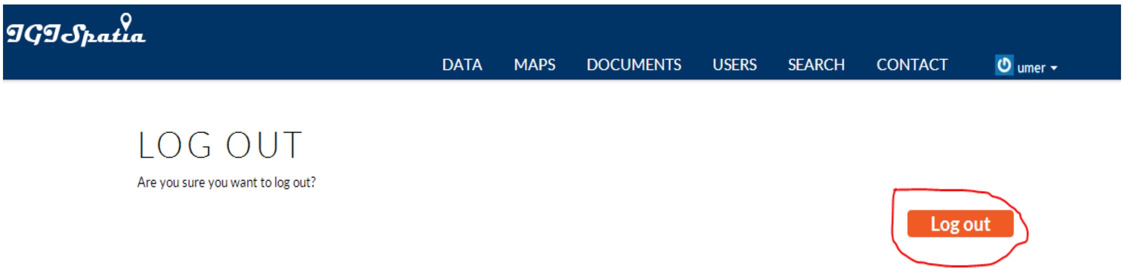

Figure 41: Logout

The next task was to check the user friendliness. The user were provided with different tasks like searching the results, commenting the layers, and user ratings, uploading layers etc. The users performed their tasks quickly as the found the application very user friendly.

Results

All the user were satisfied and they reported the user interface was very friendly and they found no difficulties in performing their tasks. Two of the five users were not satisfied with map creation method. As they said that the map creation process was not user friendly as somehow the map creation method is hidden and also the styling mechanism for the layers was not clear.

The create map section snapshots:

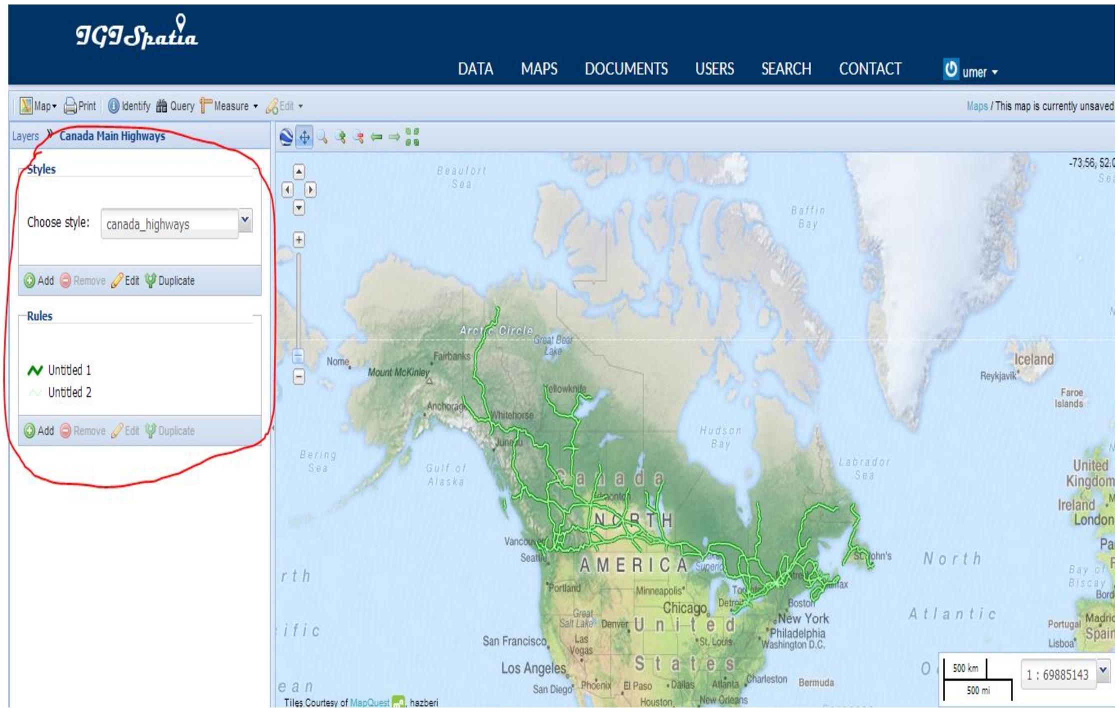

Figure 42: Layer Overlay

The highlighted section as reported by the user was not user friendly as the styling of layers is not very helpful.

3.1.1 Security Testing

Security of the application is an important thing. In case of IGISpatia the security is of prime importance as we cannot compromise the integrity of the geospatial data it contains. Security leaks can produce huge problems as it would cause data leaks from our system.

To test the security we asked few users to perform several tasks to ensure that our application has the proper and first class security. For this purpose we created users with different roles. The first task was to check that only registered user have the access to the data. First we asked some the non-registered user to access the data layers from the IGISpatia. They tried but they could not see the layers.

Results:

No layers access for non-registered.

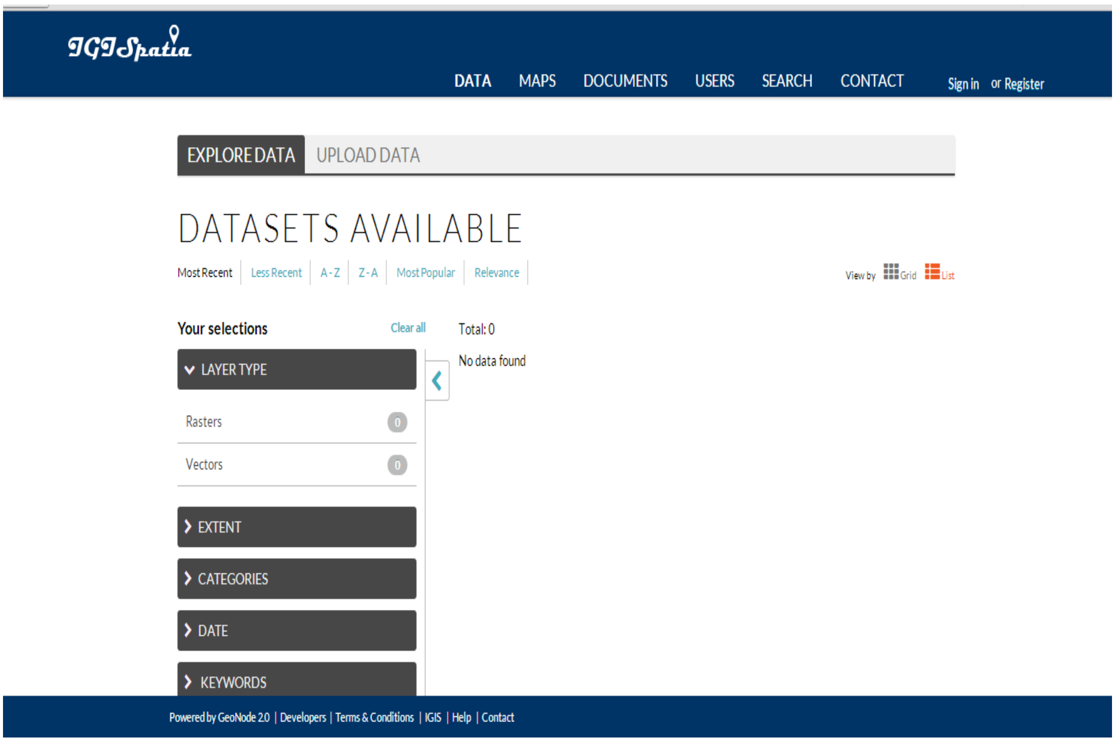

Figure 43: Non Registered User View

As we can see for the non-registered user there are no layers available.

Now we asked the non-registered user to access WMS from desktop client like QGIS. As there are no username and password provided as in the following snapshot.

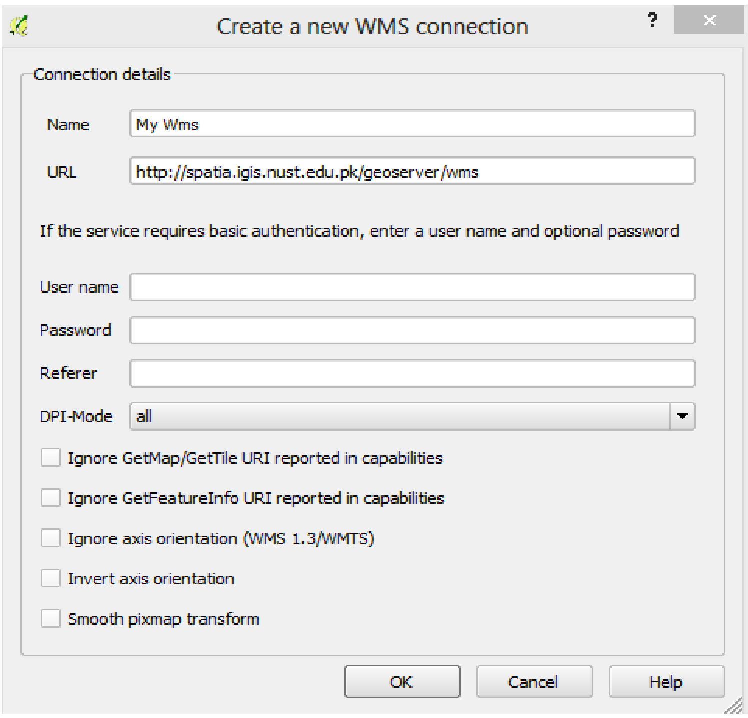

Figure 44: WMS Access in QGIS

When the non-registered user pressed ok and pressed the connect button to connect to server the WMS client showed a pop up box asking for password and user name which means the non-registered has no access to OWS services.

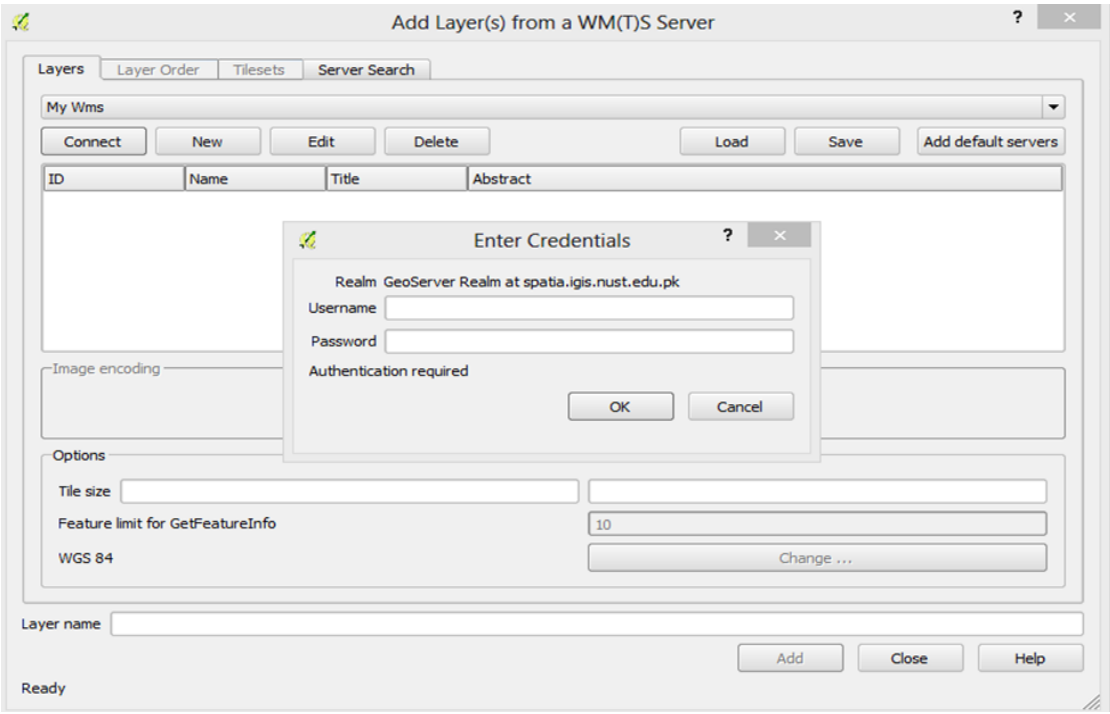

Figure 45: WMS Login in QGIS

This proved that non-registered users could not access any layers.

The same test was performed for registered users. First of all some users were created and they were given some basic rights like viewing and downloading data. When the users logged in to the IGISpatia they were able to see all the data on which they were given access.

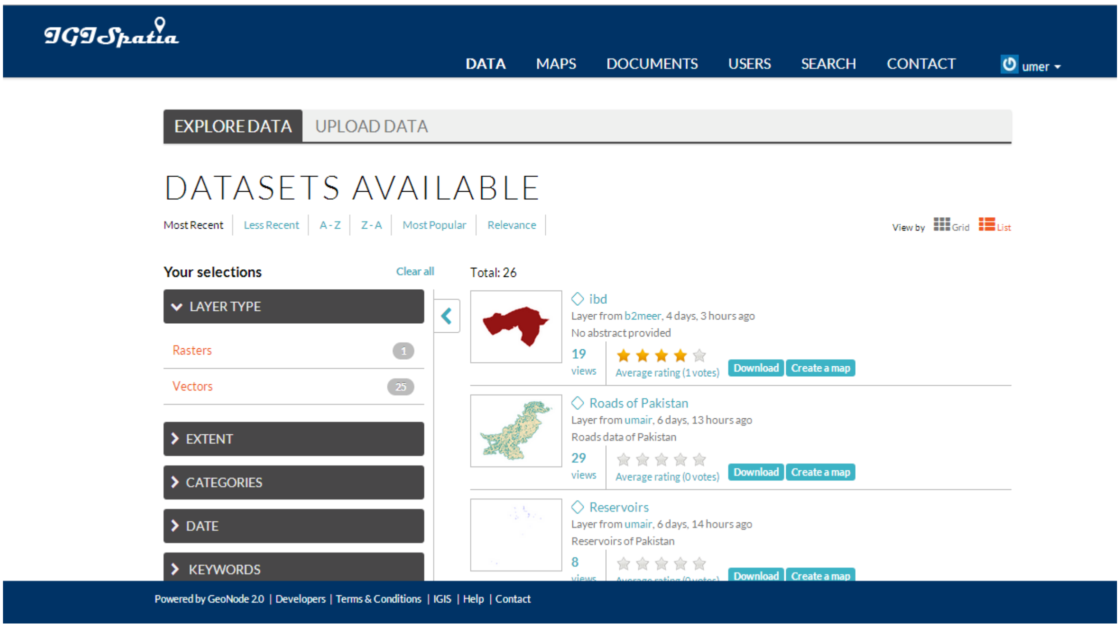

Figure 46: Data Layers

Now another thing to check for the WMS services within desktop client like QGIS.

This time the registered users provided username and password and all the WMS services was accessible within QGIS.

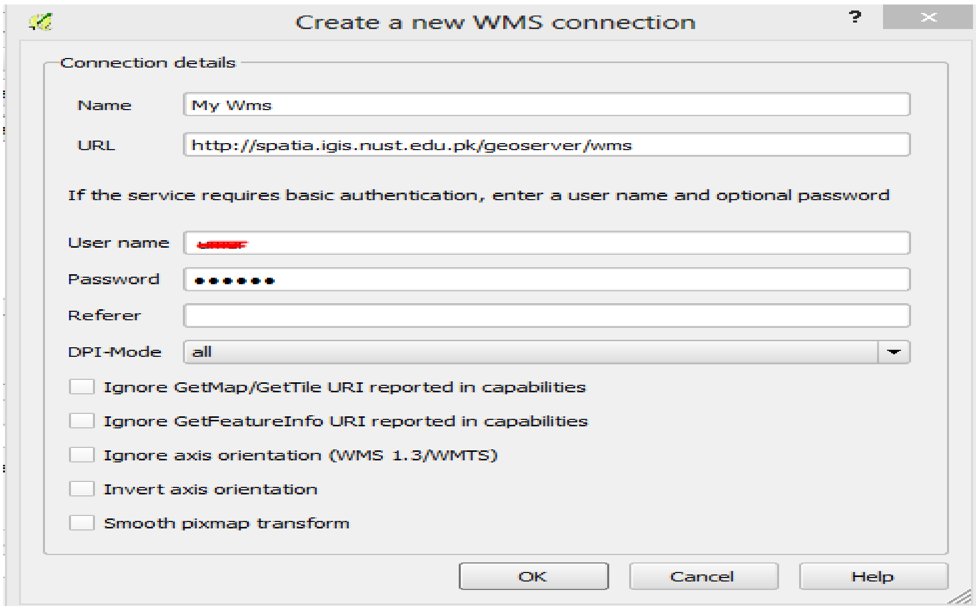

Figure 47: WMS Connection

Now after authentication all the layers were available to the registered users.

All the users registered and non-registered reported that the security was first class and strong as they concluded that there was no way to breach or compromise its security.

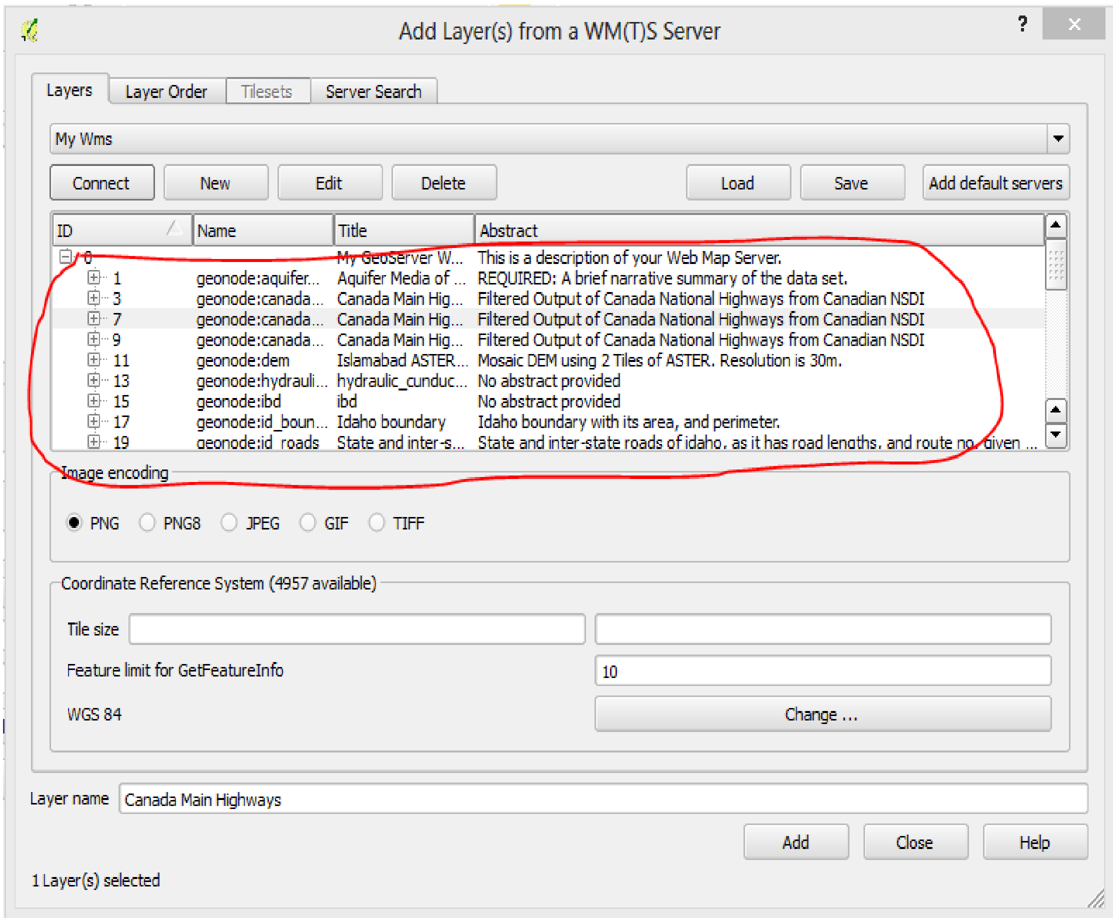

Figure 48: WMS Layers in QGIS

The highlighted portion shows all the layers that are available to the registered users.

3.1.2 User Acceptance Testing

User acceptance testing also known as end user testing is one of the phases of web application testing in which product is tested in real world by intended audience. The experience of the users are fed back to the developers which make final changes before releasing the product. Its basic objective is to meet the expectations of the user. UAT consists of five steps which are following:

3.1.2.1 Planning

The whole planning process is chalked out and the strategies for carrying out the testing are designed.

3.1.2.2 Designing Test Cases

Test cases are designed which cover all the essential functional scenarios of the spatial data repository in real world usage.

3.1.2.3 Selection of Testing Team

A team of 5-10 individuals is selected which are real world end users.

3.1.2.4 Executing Test Cases and Documenting

The testing team executes all the test cases designed and specified and documents all errors and successful results in a separate document with comments.

3.1.2.5 Bug Fixing

The developers then fix the bugs and make final adjustments to the product to make it available for market.

For testing our geospatial data repository, the same five steps were followed and the whole process was started by designing out the strategies for carrying out testing using designed test cases. For this purpose, it was planned, that different users a total of 5 would be asked to upload data from different browsers (e.g. Mozilla, Chrome) and they would be asked to give their comments regarding how acceptable the repository's overall look is to them based on which the developers will make the final changes before making the repository available for everyone.

The designing of test cases were carried out by discussing all possible scenarios in which repository can be used and finally the following test cases were designed:

- Browser compatibility
- Make sure that mandatory fields are given data in forms
- Multiple options are given for keywords
- Multiple options are given for base map when creating map

Browser compatibility was tested by asking different users to use repository on different browsers to verify that it is compatible with all browsers. After that users were asked to upload data and edit its metadata and verify that all mandatory fields required for filling up metadata were given. While filling up metadata it was also assessed that a list of keywords were given from which the user can choose the most appropriate one. This test case can also be tested by using the search tab where keyword searching results in required resources. The fourth test case is tested by asking the member of the testing team to create a map by overlaying the resource over the base map of his own choice and documenting the success and failure he/she had in all the test cases. The testing team was selected by randomly choosing 5 students from department and asking them to perform the designed test cases and document the success and failure which were to be submitted to the repository developers so that they can make the final changes. The results that were obtained from the execution of test cases were that browser compatibility had been ensured while making it as repository was running on all browsers. The mandatory fields were also given in most of the forms which were asked to be filled such as metadata or registration.

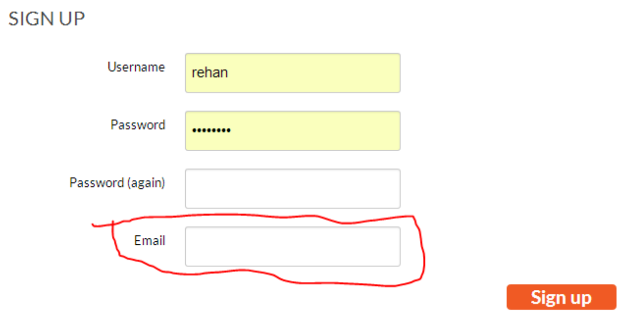

Figure 49: Email for SignUp

In the registration form "E-mail" field is mandatory as without filling it one cannot register and it has been provided beforehand. Similarly the form given for filling metadata already had mandatory fields. The options of base maps that were provided to the users were more than enough when creating maps. They had the options of choosing either the Bing Maps or the Open Street Maps. The only problem that the testers faced in these four test cases were there was no options available for keywords from which they can choose from and for this developers came out with solution that they displayed the list of keywords in selection bar that had been previously used by other users while uploading the data so that new user can select a list of keyword from that list for his data.

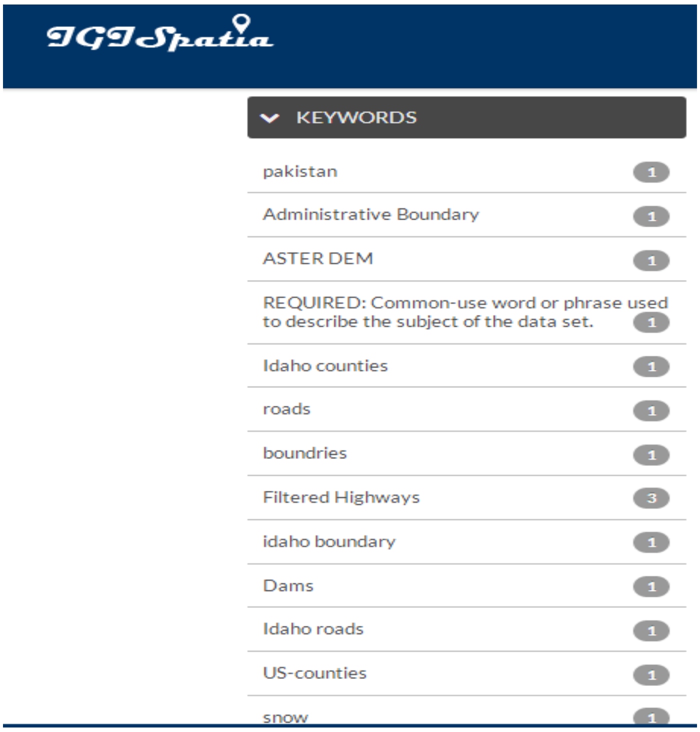

Figure 50: Keywords

Once all the bugs were fixed the product was ready to be used by the users within the department.

3.1.3 Functional Testing

Functional testing is a type of web application testing process in which web application is tested to ensure that it has all the required functionality specified within its functional requirements. It is a type of quality assurance process in which the output produced is examined on the basis of input given and to verify that it is the same output which is expected by the end-user. To carry out this testing on geospatial repository the following five steps were executed:

- Identification of upload, download, search and map creation as functions that are to be tested
- Creation of input data specific to these functions
- Determination of output based on these functions
- Executing a test case with the input data prepared
- Comparison between actual and expected outputs

In the first step the main functions that were to be tested were identified and as the four functions which are stated above are the core components of the repository so the whole testing has been carried out on them. In the next step, the creation of input data has been done separately for each function. For this purpose a sample input data of "Lahore District Boundary" was created but was used differently in each of the cases. Firstly the upload function was tested by uploading the input data and the data was uploaded successfully except when the complete files were not given so this functions outputs were quite similar to the expected outputs as the error only occurred if the user has made a big blunder while uploading. The second function to be tested was download and for this the input data was the same and when we downloaded it, the output was quite similar to what we had expected. The third function to be tested was search and when the test case was executed on this function for the input data we had defined for all functions, the outputs came out as expected for two techniques which were keyword and temporal searching but for geographical searching it did not produced the expected outcomes.

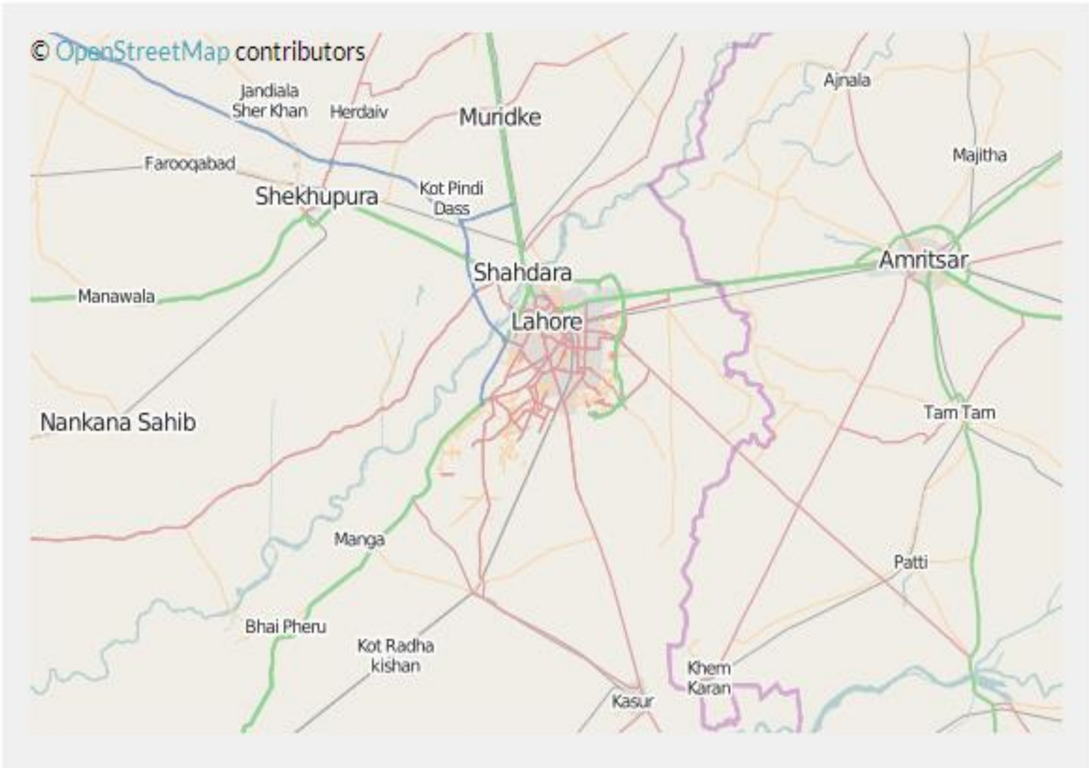

Figure 51: Data View

As can be seen in the above image that areal extent has been specified as input for selection of "Lahore District Boundary" resource but the outputs are contrary to the selection. This is mainly due to improper zooming. If proper and careful zooming in of the specified are is done then expected outcomes can be achieved.

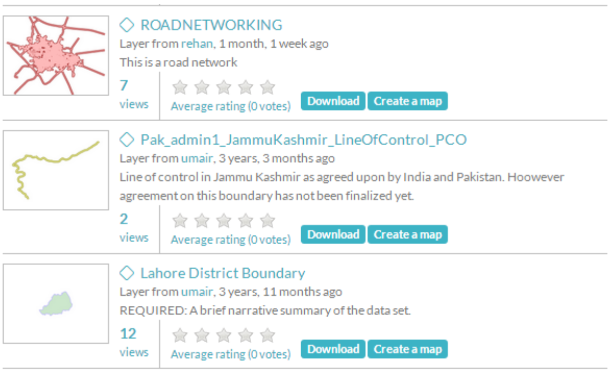

Figure 52: Layers

The above image shows the results that were obtained as a result of the above selection process.

The last function to be tested was the map creation. The same input was used to create a map and it gave the expected outcomes as layers were could be accessed from GeoServer. It also gave the additional capability of styling and embedding the map in other websites.

3.2 DISCUSSION

We as the students of Geoinformatics have gained a lot of learning and developing experience through this project. We have been learning a set of new tools and technologies by the virtue of this project. We actually feel very satisfied after the completion of this project. It seems to be an achievement for all of us. We have to learn to work in the open-source environment using open-source technologies. In this way, we were able to understand the flexibility and interoperability of these open-source technologies. From basic web development skills like HTML, CSS, JavaScript, jQuery, Open Layers to server side development in Django Python Framework, we have gained much experience. We have been introduced towards the Geospatial Content Management Systems and their deployment. A very new and exciting experience was to learn about different methods and concepts of networking which we had to come across during the implementation and web deployment stage. We have learnt to assess the quality of geospatial data, the data cleaning process, the reference information with it. We have learnt to assess different alternatives and the choice of one from among all of them for the sake of development. In short, throughout the completion of this project, we have been passing through different stages which include

- The research about Spatial Data Infrastructures
- Information about such existing systems
- Tools and Technologies used to develop these systems
- Choose the cost-effective and flexible set of technologies and environment
- Design a web application structure and implement in some technology
- Data collection and gathering
- Spatial data quality and data cleaning
- Networking and Web Deployment matters
- Documenting the work accomplished in an efficient way

Apart from all the good and nice experiences that we have encountered during the development of our product, there were also some limitations and bottle neck situations that were faced at many stages of the execution. These limitations refer to the issues related to the hardware as well as the software components of our product. These limitations are listed as follows.

- The system acting as server has the minimum required hardware specifications. For this geo-portal to run actually for the whole institute and handle all the data from students, the hardware setup needs to be upgraded to a very high level which will definitely involve many costs too.
- On the software side, there are some limitations such as only shapefiles (.shp) are currently supported as the vector data format. This is actually due to the limitation of the GeoServer (Java based-server) to synchronize with a python-based web application i.e. IGISpatia developed in Django. The extension of GeoServer to support python integration has some native bugs which will be fixed in its next versions.
- The GeoServer has already some issues related to the Web Coverage Services (WCS). This causes some problems with the use of uploaded raster data on the geo-portal. Again, the solution of this problem lies in the development of GeoServer and removal of its native bugs.
- The step of data collection and data gathering leave you with lots of geospatial data. The problem lies that all the data is not always in its purest form. In most of the cases, the data needed to be cleaned in terms of spatial reference, scale and metadata.

The above mentioned limitations are to be considered for their solution in the future. Each one of them can be solved in some way or the other.

CHAPTER 4

CONCLUSION AND RECOMMENDATIONS

4.1 CONCLUSION

The end product as one can see is a very interactive, user-friendly and professional type of web application. It was intended to be more or less the same product as it is. The basic purpose of this project has already been achieved which is related to the management of geospatial data resources. The students of IGIS-NUST can register at the geo-portal and have their own accounts. They can upload their data if they have any, download from the available data and even create online maps for different purposes. Further, for their study or research purposes, the geospatial web services allow them to use these data resources directly through the web. Other than data sharing or map creation, the site also provides a way to interact between different users through inbox. It helps in maintaining the quality of uploaded data via comments and rating options. In short the overall objective of the project has been achieved and is also available to extend its functionality further.

4.2 RECOMMENDATIONS

This geo-portal is definitely a milestone for the overall management of geospatial data resources for our institute. It needs to be in running condition and always available to the targeted users i.e. the students of the institute. It will benefit the students in their projects, assignments and other work to save their time from the hectic activity of repetitive data preparation. We see such a product to be a success in our institute. It is necessary that some initial or basic training to be provided to the students and others about using the geo-portal. The various processes like user sign-up, data upload/download and the use of geospatial web services should be demonstrated. This will enable the end users to gain maximum benefit from this product which is actually intended to be used by them. Further, this product can also be advanced to some new directions as well. As it will progress, it will be containing lots of geospatial data along with its attribute information. It can be used to setup a spatial data warehouse for some specific purpose of the institute and apply some statistical methods and spatial data mining techniques over it to find patterns based on locational information. In this way, this product opens a gateway towards the research and development side of Geoinformatics especially related to the geospatial data management and quality.

REFERENCES

Apache. (2014, 1 19). Retrieved from apache: http://www.apache.org/

David J. Maguire, P. A. (2014). The emergence of geoportals and their role. *Computers, Environment and Urban Systems*.

Fu, P., & Sun, j. (2011). *WEB GIS.* Redlands, California: ESRI Press.

Geoportal. (2014, 1 19). Retrieved from esri: http://www.esri.com/software/arcgis/geoportal

GEOS. (2014, 1 19). Retrieved from geoserver: http://geoserver.org/display/GEOS/Welcome

Halil AKINCI, Ç. C. (2014). Geoportals and their role in spatial data .

Html5intro. (2014, 1 19). Retrieved from w3schools: http://www.w3schools.com/html/html5_intro.asp

Httpmethods. (2014, 1 19). Retrieved from w3schools: http://www.w3schools.com/tags/ref_httpmethods.asp

Json. (2014, 1 19). Retrieved from json: http://www.json.org/

Laganiere, T. C. (2005). *Object Oriented Software Engineering (Practical Software Development using UML and JAVA).* Berkshire: Mc Graw Hill.

Mapserver. (2014, 1 19). Retrieved from osgeo: http://www.osgeo.org/mapserver

Pinde Fu, J. S. (2014). *Web GIS.* ESRI Press Redlands, California.

Sun, J. (2011). *WebGIS.* ESRI.

Tait, M. G. (2005). Implementing geoportals: applications of Distributed GIS. *Computers, Environment and Urban Systems*, 33-47.

The National Spatial Data Infrastructure. FGDC. (2005).

APPENDICES

APPENDIX A - REGISTRATION AND MANAGEMENT

In order to register as a new user, open the spatial data repository in a web browser by entering its web address (http://spatia.igis.nust.edu.pk.) and user will see its home page which has a "register" button on its top right corner.

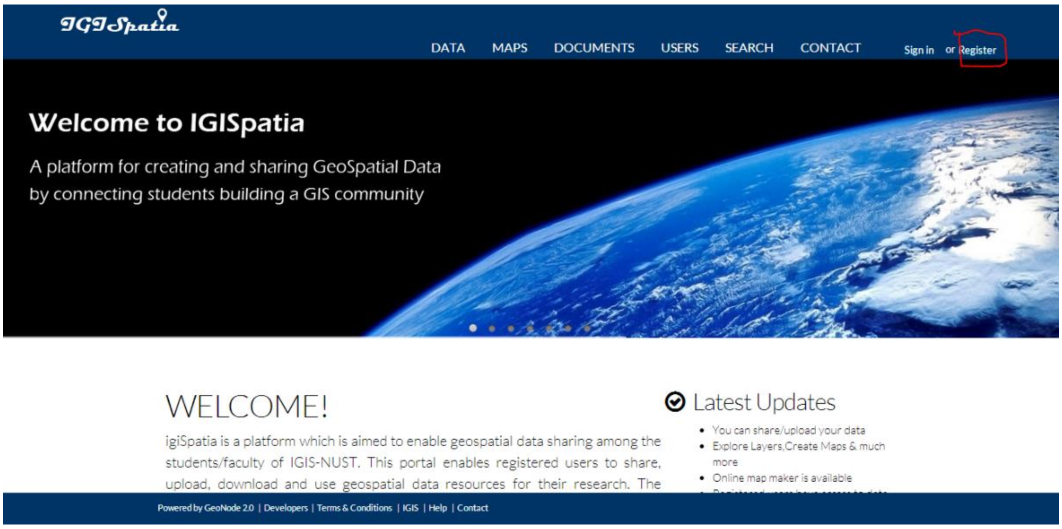

Figure 53: Home Page

Click on the register button and it will display a form with certain fields which once filled, submitted and authenticated by the system administrator will allow you to log in with the username and password which new user entered in the register form. The email address should be the one provided by the department otherwise registration would fail.

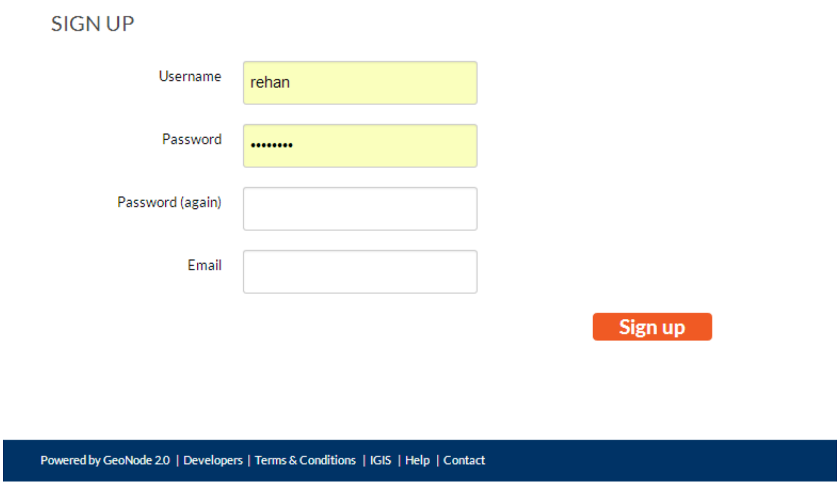

Figure 54: Sign Up

For management of user account, user must be logged in to manage his personal information, to view other user accounts or to interact with them or to manage his layers (discussed in next section).

For managing profile information click on username appearing on top right of screen and a drop down will appear which includes options such as **Profile**, **Inbox**, **Recent Activity** and **Logout.**

The Profile tab opens up a new page under which user can edit complete profile information including profile picture, password, email etc. It also offers the opportunity to add information about organization, position. It also shows all the layers maps and documents that have been uploaded by user. The logged in user can also view the activities of other users under the **Users** tab where he can see the number of layers, documents and maps uploaded by all the users or he can click on "My activities" button to see the recent activities of any specific user.

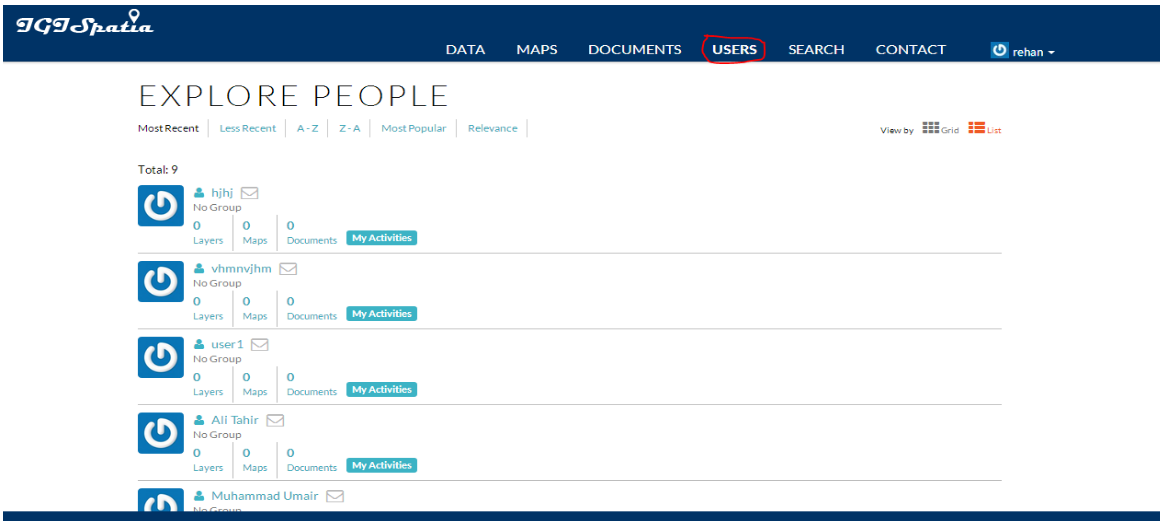

Figure 55: Users

One can also interact with the users by either clicking on the mail icon that appears next to each user which send an e-mail through Outlook or he can send a short message to the user he is interested to talk with by clicking on "**Inbox**" in the drop down menu as discussed before. Inbox opens up a new page in which user can view all the messages that has been sent to him by different users and can also create a new message which in fields are quite similar to an e-mail.

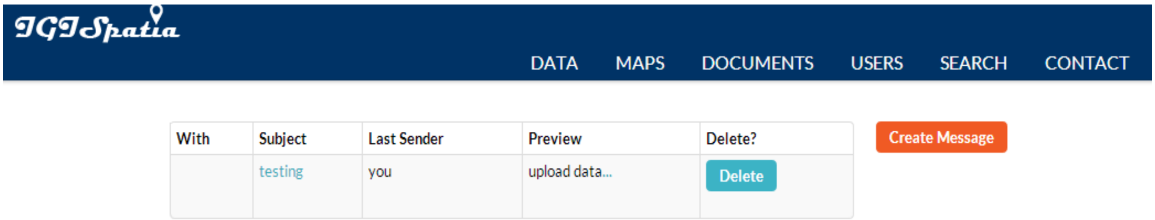

Figure 56: Inbox

The "Recent Activity" tab is quite similar to "My Activities" –the only difference being that under it you can view the recent activities of all the users while the latter allows you to see the activities of only the selected user.

APPENDIX B - UPLOADING AND MANAGING LAYERS

Layers which can be uploaded in two main formats which are shape file and GeoTiff are the basic resource in this geospatial data repository. Layers can be uploaded under the Data tab which gives user two options of exploring and uploading data. By clicking on uploading data, user is able to see a page in which he can browse for the files which he want to upload as a layer. The set and the combination of files should be complete while uploading otherwise the layer would not be uploaded successfully. The user can also set permissions on this page for that specific layer which he is uploading as to who can edit, who can view or download and who can manage that specific data which is being uploaded. In order to limit the access on any specific layer, user can specify under any one of the three options the name of users to whom the layer will be accessible. Once the data has been selected and permissions set then user will tick the check box for agreeing to the terms & conditions which will open up a pop up displaying key points of terms & conditions. Once user closes it and clicks on upload the upload will start.

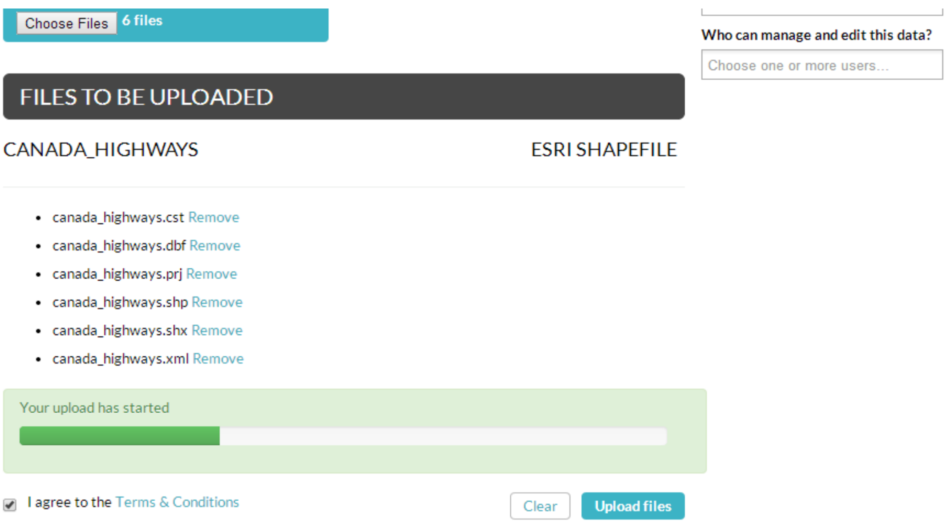

Figure 57: Uploading

Once the upload is complete, user sees three options in place of upload status bar which are “Layer Info”, “Edit Metadata”, and “Manage Styles”. By clicking on Layer Info button user will simply be taken to new layer uploaded displayed over a base map but because its metadata is yet to be filled and it should be assigned styles to make it prominent over base map so user should first click on “**Edit Metadata**” button in the drop down that will appear when user will click on “Edit Layer” tab in layer Info page.

This will take him to a new page where a whole form has been created based on ISO 19115 and ISO 19139:2007 standards for filling in metadata. Some of the fields will be prefilled if you have imported an xml file with the layer while the rest of the fields have to be filled. Once the fields have been filled, click on update. Some of the important fields are Keywords, temporal Extent, and Category as on the basis of these fields layers are searched and maintained into different categories. Once the metadata has been assigned then user can assign different styles to the layer by going to the edit styles option in same drop down menu. Or by clicking on the layer styles tool which is in the toolbar appearing at the top of map area. Some other features that "Layer Info" page offers is the option of rating the layer or giving comments regarding the layer. In addition to this user can also share the layer on Facebook or twitter or one can inquire about its attributes under the attributes tab.

The repository also gives the facility of uploading documents which are related to different map or layer resources in pdf, txt, docx, xls, jpeg, png, tiff format. They be linked to the resources to which they correspond. Permissions can also be set for documents in the way similar to which they are set up when uploading layers. The "link to" option shown in the picture below is used to link documents to layers.

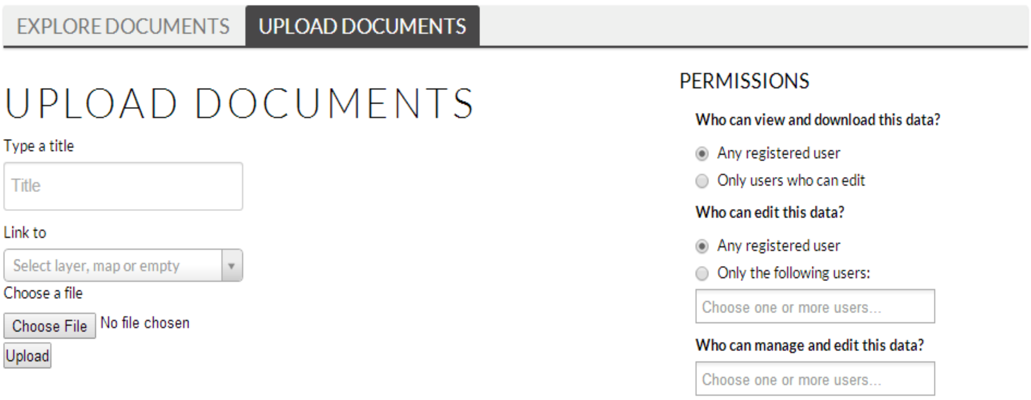

Figure 58: Documents Uploading

Another facility of the repository which can be used in context of layers and users is the **Search** option. The search tab allows the user to do three basic type of searching which are keyword searching, areal or geographical searching and temporal searching.

Keyword searching is based on the keywords which have been assigned to layers within their metadata. For keyword searching, enter the words on the basis of which one wants to search the layers in the box adjacent to Search button. Then user must select the resources which should be searched e.g. maps, users or all of them after which the category for searching the resource should be selected.

For areal (geographic) searching, user should zoom in to the area of which resources he wants to find and he should try to keep his zooming level as accurate as possible to the area to be searched otherwise irrelevant results may be obtained. For temporal searching, user should give the dates for “since this date” and “until this date” fields which will define the temporal extent of the resource which has been mentioned at the time of its metadata creation. The repository will search resources based on this extent and will return matching results. These three searching techniques can be used collectively or individually depending on the user needs.

APPENDIX C - CREATING AND SHARING MAPS

The layers in spatial data repository can be converted to maps by using the **Create Maps** option available with each layer. By clicking on this button a new page opens in which user can see his resource overlaid over the base map with various tools available to him. The user can change the base map according to his preferences. There have been multiple options made available to him which include OSM map, BING maps, MapQuest imagery and Blue Marble. The toolbar at the top left corner offers multiple tools as shown below

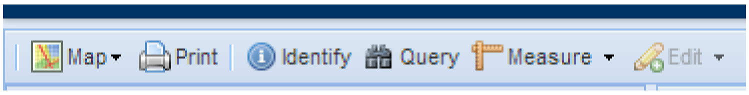

Figure 59: GeoExplorer Menu

The first tool starting from left allows user either to publish map or to save the map within the repository. The publish map option gives user an HTML code after saving the map through which he can embed the map in any other website. The dimensions of the map can also be changed according to that website

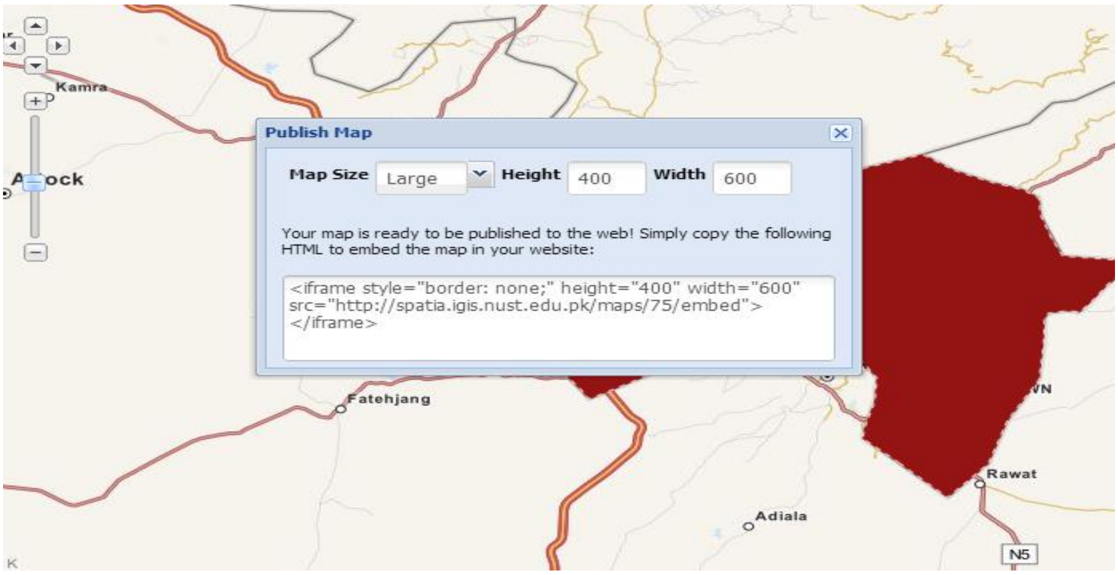

Figure 60: Map Publishing

The second tool simply allows the user to print the map by selecting the paper size and the resolution. A4 paper size and 75, 150 and 300 dpi resolutions have been provided for printing purposes. The third tool lets user to inquire about different features of the resource by simply clicking on them and opens up its attributes. It displays all the information about a feature held in an attribute tale in dbf file in a pop up window.

The fourth tool in the toolbar is the query tool which allows user to query the selected resource attributes and select and filter out its features depending on the criteria which he sets in the query tool. By clicking on the query tool, a query panel and a tables panel of the selected resource opens.

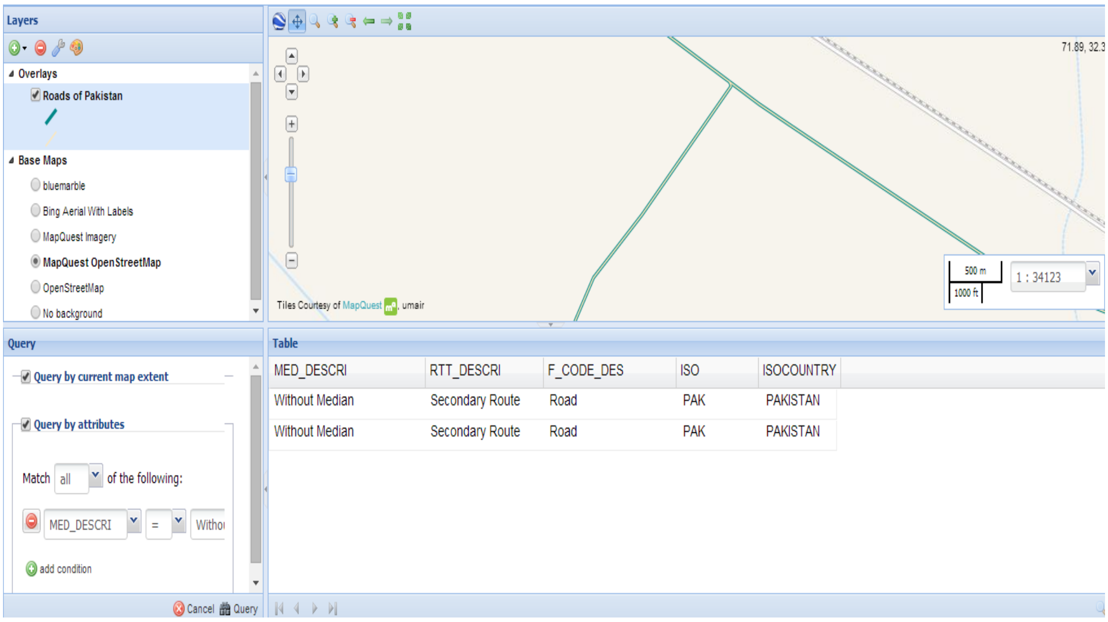

Figure 61: Attribute Query

In query panel the first field allows the user to select whether he wants the record to completely match with his query (ALL) or a part of it (ANY) while the second field allows the user to set up his query by first selecting the field then selecting the operator and finally selecting the values. Queries can be run both by map extent and by attributes. Once the query has been run, the selected results of the table are shown in the tables' panel while the features are shown in the map panel.

The fifth tool is the Measure tool which helps user in measuring the parameters of current resource. It offers two options of which one is two measure length and the other is to measure area. Both are simple to use and similar in a way in which measurement tools are made available in other GIS software (e.g. Arc Map). The last tool is the edit tool which helps user in editing the style of the layer.

The contact tab can be used to view the developers of the spatial data repository and to access their contact information. In order to make the repository user friendly searching or selection options have been provided under all three tabs i.e. Data, Maps and Documents and resources can be selected on the same three techniques as discussed above in the searching section. The links to IGIS official website, detailed terms & conditions and further help have been provided in the footer.