UNIVERSITY OF NAIROBI

SCHOOL OF ENGINEERING

Department of Geospatial and Space Technology

MOBILE DEVICE WEB-BASED COMPLEMENTARY TEACHING OF GIS

Benjamin Munyao

F56/67517/2013

Project submitted for the Degree of Master of Science in GIS, in the Department of Geospatial and Space Technology of the University of Nairobi

June 2015

DECLARATION

I, Benjamin Munyao hereby declare that this project is my original work. To the best of my knowledge, the work presented here has not been presented for examination in any other university.

Signature……………………………… Date: …………………

Name: Benjamin .M. Munyao

Registration number: F56/67517/2013

This project has been submitted with my approval as university supervisor.

Signature……………………… Date: …………………

Professor. John Bosco Kyalo Kiema

Supervisor

Associate Professor, Department of Geospatial and Space Technology

University of Nairobi

ACKNOWLEDGEMENT

I would like to thank the entire team of Academic staff in the Department of Geospatial and Space Technology for their valuable advice in improving my project. In particular want to thank my Supervisor Professor Kiema for his advice on the way forward especially during the time I had health problems. To my family, you’ve always been there for me thank you so much.

ABSTRACT

The presence of mobile devices provides an opportunity to complement Geographical Information Systems (GIS) education. They contain internet browser application which can act as a thin client to Web-based Geographic Information Systems. This project aims to show that free and open source software (FOSS) provides a viable alternative to using proprietary software in terms of cost, ease of use, performance and conformance to open standards. The development of such an open source system clearly illustrates the logistics of open system architecture and the orchestration of the interaction between open source software's. The opportunities for such a system are immense. Students are no longer tied to desktop applications to carry out their GIS practical work. Institutions in Kenya have an opportunity to take advantage of a system with no installation on their part. Today's university students are "digital natives." They were born into the digital world, and web-based learning comes naturally to them. Web GIS can engage students in a variety of fields and improve teaching and learning effectiveness. Mobile devices initially seen as a distraction in class provide an opportunity to complement teaching GIS in ways not experienced before.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

CSS-Cascaded Style Sheet

CSV- Comma-separated values

DOM- Document Object Model

ESRI – The Economic and social research institute

FOSS - Free and open source software

GIS - Geographic Information System

HTML – Hypertext Markup Language

HTTP - Hypertext Transfer Protocol

OGC - Open Geospatial Consortium

OSM – OpenStreetMap

SLD - Styled Layer Descriptor

SRS - Spatial Reference System

SVG – Scalable Vector Graphics

TAB – Tablet

WBGIS – Web-Based Geographic Information System

WCS - Web Coverage Service

Webinar – Web-based seminar

WFS - Web Feature Service

WMS - Web Map Service

WWW - World Wide Web

XML - extensible Markup Language

CHAPTER 1: INTRODUCTION

1.1 Background to the study

The intersection of GIS and what is now cloud computing has been traditionally referred to as Web GIS. Consistent with cloud-based applications, in Web GIS one does not need to store large digital geographic data files on the local computer to view and analyze those data. Beyond the easing of barriers related to data acquisition, software licensing, data storage, and processing power, Web GIS and similar cloud-based applications offer teachers and learners a tool for producing and publishing user-generated geographic content.

Due to ease of use, accessibility, lower cost, and interactivity, Web GIS is much more likely to be embraced by the educational community than professional GIS software community. The potential for Web GIS may be even greater when opportunities for collaborative environments are paired with Web GIS and other cloud-based tools.

Web GIS provides a simple online interface that both GIS novices and experts can use to access GIS functionality. They can view data sets without requiring software or data on the user's computing device, thus moving beyond the mere posting of static maps of roads or geographic features on a web. For example, users can look at areas of interest then select which layers they wish to display. They can also query data sets, and even buffer features.

Postgraduate GIS students in this university come from different academic disciplines and with varying GIS practical skills. Time is required to plan for a full GIS installation and licensing of expensive software as well as powerful hardware that has a lot of storage space for the typically large GIS datasets. As a result, GIS Laboratory demos are done towards the end of the respective course units thereby putting students with little GIS background at a disadvantage due to lack of visible learning[1]. Web GISs have the potential of addressing time issues for educators. The simple interface and pre-collected data of a Web GIS site require less training for teachers to learn to use, as well as less expertise for them to teach in their classrooms. For example, students do not have to be in the laboratory to show their lecturer the progress of say digital cartography

[1] http://visiblelearningplus.com (last accessed 23rd Feb 2015)

assignment. This can be done from a mobile device in a lecture room along the corridors or staircases. Students would know whether they are on or off course in their assignment.

1.2 Problem Statement

We live in a world that that requires skilled workforce. These can be achieved through individualized teaching which is not feasible with large class numbers. Blanket extension of assignment hand in date for example, assumes students share the same problem that requires more time to solve. Mobile devices present an opportunity to solve some of these problems. The aim is to develop a web-based software system that makes it possible for students to carry out spatial data visualization on mobile devices. The system should provide templates that users would need to make minor changes to suit their data. The aim should be to ensure that data visualization is not restricted in a computer lab. Lecturers should be able to monitor progress of data visualization assignments in and out of formal class settings.

1.3 Overall Objective

The main objective of this study is to develop a Web GIS-based education software solution that can be used across mobile devices i.e. Tablets and Smart phones running on different operating systems.

1.4 Specific Objectives

1. Demonstrate Web GIS on mobile devices as way of teaching GIS principles and applications;
2. Demonstrate that Web GIS encourages important learning methodologies like Visible Learning.
3. Demonstrate that open source tools have come of age and can be used in developing Web GIS based teaching software.

Activities intended to achieve specific objectives

- Use online resources to create a map.
- Evaluate geographic data for use in a mapping project.
- Explore a GIS map and access information about map features.
- Visually analyze feature relationships and patterns on a map.
- Create queries to find and select features that meet specific criteria.

- Prepare a map to share information and present analysis results.

1.5 Justification of the Study

The technology that underlies GIS is permanently evolving towards the Internet. Software and hardware investment has been one of the biggest impediments to GIS uptake in Africa in general, and Kenya in particular. However Web GIS based on FOSS reduces barriers to entry. A sample of mobile devices used in this project is shown in figure 1.1. Opera mobile emulator software has been used to emulate the device software.

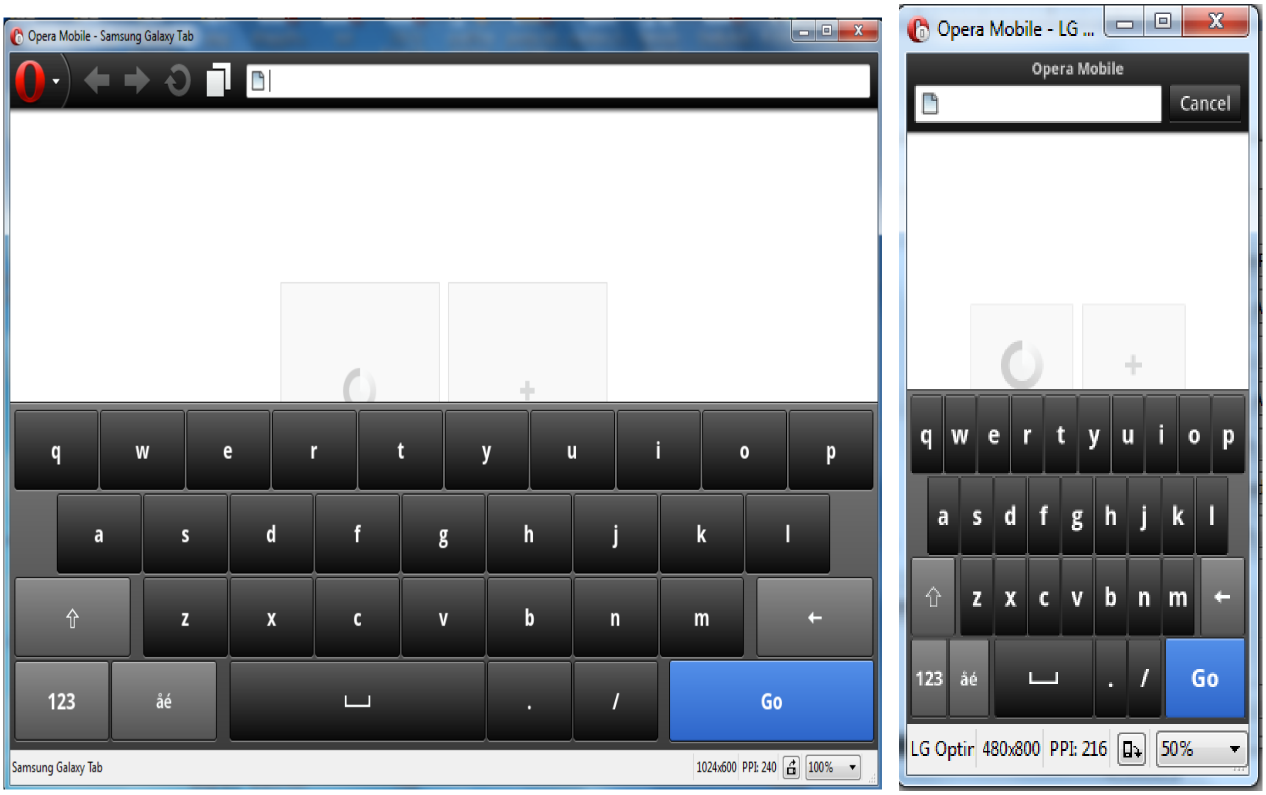


Figure 1.1: Samsung Galaxy Tablet and LG Optimus 3D smart phone

CHAPTER 2: LITERATURE REVIEW

2.1 Trends in GIS Education

Increasingly more classrooms are comprised of students with a variety of intellectual capabilities and preferred learning styles. Employing GIS in various courses provides students with the opportunity to interact with spatial data in ways that facilitate engagement with the material, as well as enhance overall learning. At more advanced levels, GIS is a tool for display, inquiry, and analysis widely used in undergraduate research projects.

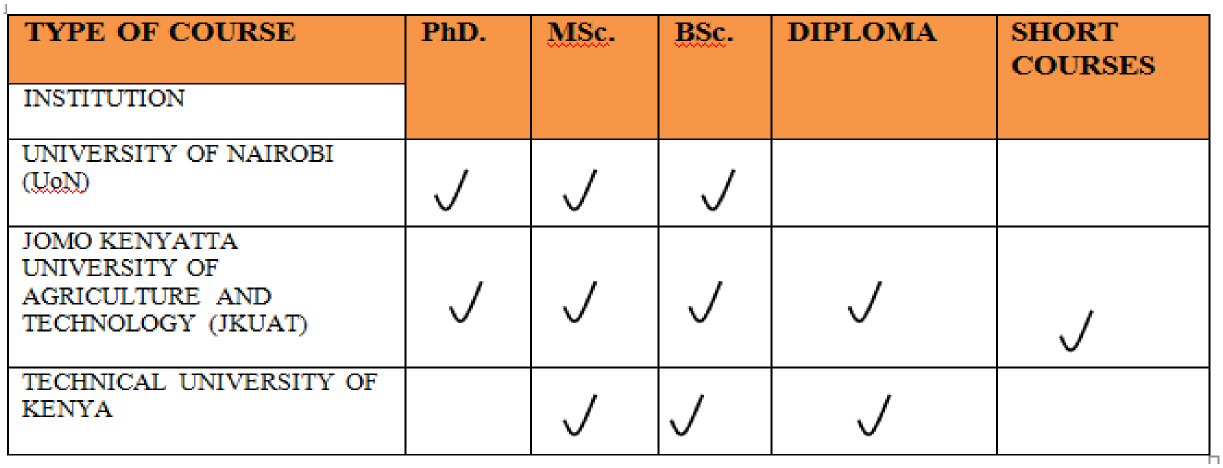
2.2 GIS in Elementary Education

While we generally think of GIS education as being concentrated largely in universities, it is now extending beyond that confine. OpenStreetMap (OSM)[2] has started a project with young school children from Kibera slums in Kenya to map it[3].

2.3 GIS in Higher Education

Kenya in particular has several universities offering GIS at various levels. Before registering for my master's degree program in this university, I did a survey on GIS education in Nairobi and its environs and the results were as shown in Table 2.1.

Table 2.1: Institutions and type of courses in GIS



TYPE OF COURSE	PhD.	MSc.	BSc.	DIPLOMA	SHORT COURSES
INSTITUTION					
UNIVERSITY OF NAIROBI (UoN)	✓	✓	✓		
JOMO KENYATTA UNIVERSITY OF AGRICULTURE AND TECHNOLOGY (JKUAT)	✓	✓	✓	✓	✓
TECHNICAL UNIVERSITY OF KENYA		✓	✓	✓	

[2] OpenStreetMap Wiki, http://wiki.openstreetmap.org/wiki/Main_Page last accessed 2nd May 2015

3 http://kibera.ushahidi.com/ (last accessed 2nd May 2015)

2.4 Teaching with Internet-Based GIS

Despite the changes in GIS education over the past 20 years, its model of delivery has held remarkably constant over much of that time. While it is true that more spatial data has become available and accessible, and while hardware, software, and computer networks have certainly become more powerful and mobile throughout the period, the hardware-software model of GIS use has remained more-or-less the same. This model has essentially consisted of loading and running GIS as a desktop software application on a computer running UNIX, Windows, or Macintosh operating system, and loading and using spatial data from an outside source. At first, the source of this spatial data was largely drawn from physical media including magnetic tapes, ZIP drives, CD-ROMs, DVDs, or some other external device. Lately, spatial data has been increasingly downloaded from the Internet. However, the result was still the same with the data stored on a hard disk. The hard disk was either local, externally attached to a local computer, or on a local network.

The above model provided the foundation for 20 years of teaching with GIS and about GIS (Sui, 1995). However, since the advent of Internet-based GIS, teaching with GIS and teaching about GIS have both been slowly but inexorably changing. This project focuses on the changes that Internet-based GIS brings to teaching with GIS.

New tools present new opportunities for the educator and learner as well as new challenges. As never before, educators today have a wealth of geospatial tools at their fingertips, most of which require fewer hours of training than traditional GIS. These tools can hasten the ability of learners to think spatially. They can also support standards-based inquiry-driven methods of teaching and learning, while providing basic analysis tools for exploring geographic or scientific data (Baker, 2005).

2.5 Benefits and Challenges of Internet GIS in the Curriculum

The advantages of Internet-based GIS for the educator are numerous (Baker, 2005). No software is required beyond a standard web browser. This alone, especially for instructors who are faced with difficulties with the school technology staff in installing any software not viewed as standard in the school, is a significant benefit. Furthermore, most web-based GIS require little time to learn; drastically reducing the time it takes for educators to incorporate them into their curricula. Dynamic content, such as real-time wildfires or earthquakes, can be analyzed online in a way not possible in the desktop environment. A significant advantage of online GIS over desktop GIS particularly in countries where all data are licensed is that the data are being viewed online and therefore is accessible without cost or licensing restrictions.

Many web-based GIS tools can be effectively used to enhance teaching and learning in conjunction with traditional desktop-based GIS. Teachers seem better able to make the leap to using maps to analyze data rather than just looking at places when there is an intermediate technology that introduces them to spatial analysis. Most of these tools function in the same way within a desktop GIS environment that they do in the Internet-based environment. Furthermore, Internet-based GIS provides an excellent avenue to more advanced spatial analysis with desktop GIS (Baker, 2005).

2.5.1 Challenges to Web-Based GIS in Education

Challenges in the use of web-based GIS as an instructional tool exist just as they do in the desktop GIS environment. Most of these challenges are not easily solved with software and hardware, because they are embedded in the educational system itself (Kerski, 2003). Digital maps, whether on the Internet or as part of a desktop GIS package, present special challenges for interpretation and analysis (Cammack, 1999). For teachers who have previously relied on maps only for reference, and not for investigation, this represents an interpretation as well as a pedagogical shift. Another challenge is that web-based GIS services do not and may never contain all of the functions available in a desktop GIS. A recurring theme of teaching with online or desktop GIS is data quality. Maps are by definition only abstractions and representations of reality. With Internet-based GIS, much less of the data adheres to metadata standards, and therefore, varies more widely in terms of data quality.

A further challenge of web mapping services compared to desktop GIS is that because the maps and services are online, one cannot teach with them if the website or the Internet connection is down. A related disadvantage for Internet-based GIS use is that, unlike desktop GIS use, it is difficult to save one’s work and return to it during the next day’s class period. As web-based mapping tools have increased in power, many of them require a high-speed Internet connection, which although becoming more common in primary and secondary schools, are far from ubiquitous.

Another challenge is that some of these services link to advertisements and chat rooms that either cause the entire site to be blocked by school servers or else are considered undesirable by teachers and school information technology staff. Internet-based GIS are often less stable than desktop GIS, and they are often slower, since they are streaming data over the Internet rather than simply showing data on the local computer. Some web-based services become so captivating to many students that they have a difficult time concentrating on the actual problems assigned. A related concern is how much "control" an instructor feels is necessary for successful teaching. For many educators, using the Internet for anything besides research for term papers is venturing into uncomfortable instructional territory (Cammack, 1999).

To overcome these challenges a shift is required from user interfaces where users "click buttons" to writing code since web apps run from the server and not the user's device which only displays results. Trying a solution similar to desktop application would imply having a rich client. This would imply that users have to install software on their devices. We have mobile devices from different manufactures having different operating systems which can pose serious interoperability issues. Developing a version for every device out there is very expensive.

2.6 An Example of Teaching with Internet-GIS in the Curriculum

Arc Web Explorer is a web mapping service built by the Economic and social research institute (ESRI) that effectively illustrates the power of some of these new tools[4]. Arc Web Explorer was built to showcase the data available through Arc Web Services, a library of information and tools compiled by ESRI that contains over 100 different sources of spatial content such as streets, imagery, and even live data such as traffic. Arc Web Services are a collection of tools that enables users to integrate mapping, routing, and geographic reporting into their web sites, wireless devices, or desktop applications. With Arc Web Services, neither the content nor the software has to be resident on a student or teacher's local computers. Arc Web Services allows users to perform a variety of spatial query and proximity searches on a host of datasets provided by some of the GIS industry's leading data providers or by other GIS users. With the routes, base maps, thematic maps, and reports, the goal in industry or in education is to solve geographic problems. Students can use Arc Web Explorer to investigate over 100 demographic variables down to the neighborhood level, such as total population, median income, education, age, and

[4] http://arcwebservices.com/awx (last accessed 22nd Feb 2015)

ethnicity. They can compare urban core neighborhoods to first ring suburbs to ex-urban to rural, or compare their own neighborhood to those nearby or across town[5].

2.7 Mobile Devices

Technology is a tool that can assist in the transitions needed for 21st century learners. The ability to incorporate the devices that students already use into the educational programs is critical to the future direction and success of schools. Making school buildings accessible to these devices is as critically important as training teachers and staff to be able to incorporate the devices. The prospect of a wireless device in every student's hand with real-time assessment and feedback presents the potential for a sweeping paradigm shift to learner-centered education (Caverly et al., 2009).

Educators are challenged by traditional classroom structures. Often we present the same lesson to a classroom of students offering them the same amount of time and resources to learn the material. However, this teaching approach is based on the premise that all students learn the same way and at the same rate. And although we maintain high expectations with an intense curriculum, this is not always an ideal learning environment. In a student-centered classroom, the teacher begins with mastery of the core standards in mind. However, the methods and pace in which students reach and exceed mastery look very different based on their educational needs and interests. Learning becomes the constant; time becomes the variable; and students leave our classrooms with the most crucial lesson we can present: how to keep learning. (Mahmoud 2008)

Firstly, the mobility component means that students are free to move within, beyond and between multiple environments and between topics and disciplinary contents and contexts. Secondly, use of the portable devices means that learning is not confined to formal educational contexts; learning is extended within informal opportunities such as home and work. Thirdly, mobile learning is not a one-way discourse/exchange from lecturer to students, but constructs understanding within participating communities. The fact that all this devices are touch screen makes it easier to pan, zoom in and out with one's fingers as use of a mouse has been fairly cumbersome in many applications. The nearly universally presented rationale for mobile learning

[5] http://arcwebservices.com/awx (last accessed 22nd Feb 2015)

is that the current and incoming tech-savvy 21st century learners, many of whom, have “grown up digital” expect it (Andone et al., 2009).

2.8 FOSS and its use in web mapping

Free and open source software (FOSS) includes a number of ideas that can invoke complex and even emotional discussions within the technical community[6]. Given the value of software, it's no surprise that commercial businesses invest millions in researching, developing, and selling proprietary software. They protect it through patent and copyright laws. They obfuscate the code to make it harder to copy or reverse engineer. Commercial software development has become a successful enterprise whose resulting tools have shaped our work and home environments. FOSS is “free” in the sense that it is open and amenable to use and modification.

If FOSS is usually available without cost, why is it such a successful concept? And how do people make a living from coding with it? First, it's important to mention that many developers contribute to FOSS solely for personal enjoyment and for the desire to donate their skills to a project offering wide societal benefits. They enjoy working in virtual teams and facing the challenges of rigorous software development that their ordinary “day jobs” might not offer. Other people may not contribute frequently to FOSS projects, but can still make a profit selling support services for FOSS software. When one buys proprietary software, one has access to a technical support package that allows one to call and talk to an analyst if is needed. Because FOSS does not offer these official support systems, enterprising individuals have stepped in to fill this void[7].

2.9 Open Geospatial Consortium

The Open Geospatial Consortium (OGC)[8] was formed in 1994 from the Open GRASS Foundation (OGF). It is a non-profit organization working in collaboration with more than 480 companies (by 2013) government organizations and universities all over the world. The major objective of the OGC is to develop international standards for geospatial data and to set standards for public use. OGC standards come with a technical documentation and encodings.

[6] https://www.e-education.psu.edu/ Understanding FOSS and its use in web mapping _ GEOG 585 Web uMapping.htm

[7] https://www.e-education.psu.edu/ Understanding FOSS and its use in web mapping _ GEOG 585 Web uMapping.htm

[8] http://www.opengeospatial.org/resources/ (last accessed 2nd March 2015)

OGC® and OpenGIS® are the trademarks of OGC standards. The standards allow the developers to create complex spatial information and make it accessible to public and can be useful in different kinds of applications.

2.9.1 Web Map Service

One fundamental component of the web map (and the simplest to understand) is the map image. The Web Map Service (WMS) is a standard protocol for serving georeferenced map images generated by a map server[9]. In short, WMS is a way for a client to request map tiles from a server. The client sends a request to a map server, and then the map server generates an image based on parameters passed to the server in the request and finally returns an image.

2.9.2 Web Feature Service

A web map service (WFS) returns the actual geographic data that comprise the map images. One can think of the geographic data as the "source code" of the map. This allows users to create their own maps and applications from the data, to convert data between certain formats, and be able to do raw geographic analysis of served data. The protocol used to return geographic feature data is called Web Feature Service[10].

2.10 Web Coverage Service

The Web Coverage Service (WCS) is a service that enables access to the underlying raster (or "coverage") data. In a sense, WCS is the raster analog to WFS, where one can access the actual raster data stored on a server[11].

2.11 Web Processing Service

The Web Processing Service (WPS) is a service for the publishing of geospatial processes, algorithms, and calculations[12]. WPS extends the web mapping server to provide geospatial analysis. WPS in GeoServer allows for direct integration with other GeoServer services and the

[9] http://docs.geoserver.org/2.6.x/en/user/index.html (last accessed 3rd Feb 2015)

[10] http://docs.geoserver.org/2.6.x/en/user/index.html (last accessed 3rd Feb 2015)

[11] http://docs.geoserver.org/2.6.x/en/user/index.html (last accessed 3rd Feb 2015)

[12] http://docs.geoserver.org/2.6.x/en/user/index.html (last accessed 3rd Feb 2015)

data catalog. This means that it is possible to create processes based on data served in GeoServer, including the results of a process to be stored as a new layer. In this way, WPS acts as a full browser-based geospatial analysis tool, capable of reading and writing data from and to GeoServer.

2.12 Open Layers

Open Layers (OL)[13] allows multiple map servers or backend map servers that can deliver data to the map. These different map servers can be added to the map as layers using OL. Each layer on the map will add on top of each other just like the desktop GIS applications. Google or Yahoo map servers can be used as background maps and any kind of raster or vector data can be displayed on top of platforms. The advantage of using vector data is that it allows users to add GIS tools to the map on top of it.

2.13 Map Controls

In general a simple web map should have the following components[14]:

- Main Map: This is the major component of the map. On top of the map raster or vector data can be projected.
- Overview Map: This is also a main map but with navigation window. It acts as a reference map to provide overview location. The zoomed portion of the main map can be seen on the whole map with a rectangle box highlighting the zoomed area.
- Pan/Zoom Bar: It is a zoom slider to select the zooming level of the map.
- Layer Overview: This portion of the map contains the data present in layers. Layers can be selected or unselected here.
- Toolbar: This portion of the map contains GIS tools like panning button, zoom button etc. The tool box can be placed on the map or outside the map.
- Scale Bar: Scale bar shows the current scale of the map.

[13] http://openlayers.org/ (last accessed 3rd Feb 2015)

[14] https://www.e-education.psu.edu/ Spicing up your map with OpenLayers controls _ GEOG 585 Web Mapping.htm

2.14 Web page design

In order to build a web page some basic things are required as discussed below[15]:

Page construction (e.g. HTML)

Styling (e.g. CSS)

Page interaction (e.g. JQuery)

2.14.1 HTML

Hyper Text Markup Language (HTML) is the standard and universally accepted markup language used to construct the web page[16]. HTML defines the structure of the web page using standard tags which are enclosed in angle brackets (e.g.: <html>) and attributes. All the HTML tags are coming up with pair i.e., beginning tag and ending tag (<html>, </html>). The closing tag looks exactly the same as beginning tag with extra back slash "/" before the tag name. Although there are several hundreds of HTML tags available in the language, all these tags are not required to construct a web page. Two basic tags are necessary to construct a web page as demonstrated below.

<!DOCTYPE html> this tag defines the page as HTML

<html> starting the script with. All the script lies under this tag. (Beginning tag)

<body> content of the page resides under this tag. (Beginning tag)

<h1>My First Heading</h1> this tag is for heading on the web page (Both tags)

<p>My first paragraph. </p> this tag is for a paragraph on the web page. (Both tags)

</body> this tag in the script closes the beginning tag <body> (Close tag)

</html> this tag in the script closes the beginning tag <html> (Close tag)

[15] http://w3cschools.org

[16] http://w3cschools.org

2.14.2 CSS

Cascading Style Sheet (CSS). It defines the style of HTML objects on the web page[17].

Using CSS, it is possible to display or adjust the position of different objects like table, paragraph, heading, copyright etc. on the web page. A simple style for a paragraph in a web page is shown below:

```
<p style="color: sienna; margin-left: 20px">This is a paragraph.</p>
```

This defines color of the paragraph. It delineates that the paragraph should be at a distance of 20 pixels away from the left margin.

2.14.3 JQuery

JQuery is a client side JavaScript library that can be used to create dynamic web pages[18]. With JQuery it is possible to navigate to the web page and create animations on the page. The simple JQuery script is shown below.

```
$(document).ready(function() {

$("#p").click(function() {

$("h1").css("background-color","red");

});

});
```

This ensures that when a user clicks on the paragraph on the page, JQuery triggers background color to red for h1 heading on the web page.

[17] http://w3cschools.org (last accessed 3rd march 2015)

[18] http://w3cschools.org (last accessed 23rd Feb 2015)

2.15 PostGIS

PostGIS “spatially enables” the PostgreSQL open source relational database. The database can then be used to store and query spatial data (points, lines and polygons)19. It has the following capabilities:

- High performance, robust spatial database built on PostgreSQL[20]
- Simple Features for SQL (SFSQL) compliance
- Proven reliability and transactional integrity
- Provides spatial representations of geometry types (points, lines, polygons)
- Support for common and advanced spatial operations such as geometry creation and conversion, buffer, convex hull, generalization, union, and more.

[19] http://postgis.refractions.net/ (last accessed 23rd Jan 2015)

[20] http://www.postgresql.org/(last accessed 23rd April 2015)

CHAPTER 3: DESIGN AND METHODOLOGY

3.1 Model-View-Controller structure and Data Model

The web based GIS application in this study has been developed under a software pattern known as Model-View-Controller[21]. The following sections describe these tiers in detail. Figure 2 illustrates the interaction among tiers, outlining the main components of the system and their logical position. This is a common pattern for developing of software which divides the code according with the main parts to construct any computer program. These parts are generally based on data (information), interface and commands (functions) that include model, view and controller respectively. This structure represents an advantage to focus segments of software and makes modifications without affecting other parts. GIS web application makes comparison between MapServer and GeoServer which process WMS, WFS and WPS represents the "Controller" part, therefore "Model" and "View" could be the same for both developments. Model-View-Controller pattern makes the structure of the program to be divided by model or structure of data, view or interface and controller or commands to define functions.

MODEL: Data storage according with Entity-Relationship model which involves every edition of data into the same environment, the model is within PostgreSQL with its respective PostGIS extension. After analysis of data collection are defined variables which make up the system. Then the spatial database to store the information is designed. Additionally, creation of flow charts according to functions and basic structure "input/output" to identify queries of GIS web application is done.

VIEW: Interface and its design are done by a kind of programming language, which is separated from commands development or functions. The view segment is constructed through HTML and CSS.

CONTROLLER: This is the programming which responds to the client requests. The programming area is based on JavaScript for GeoServer.

[21] http://www.ibm.com/developerworks/Struts, an open-source MVC implementation.htm (last accessed 3rd May 2015)

Typical client-server architecture as well as the basic architecture is employed in the development of the system is shown in the figures 3.1 below. Also indicated are the FOSS software's used in each block.

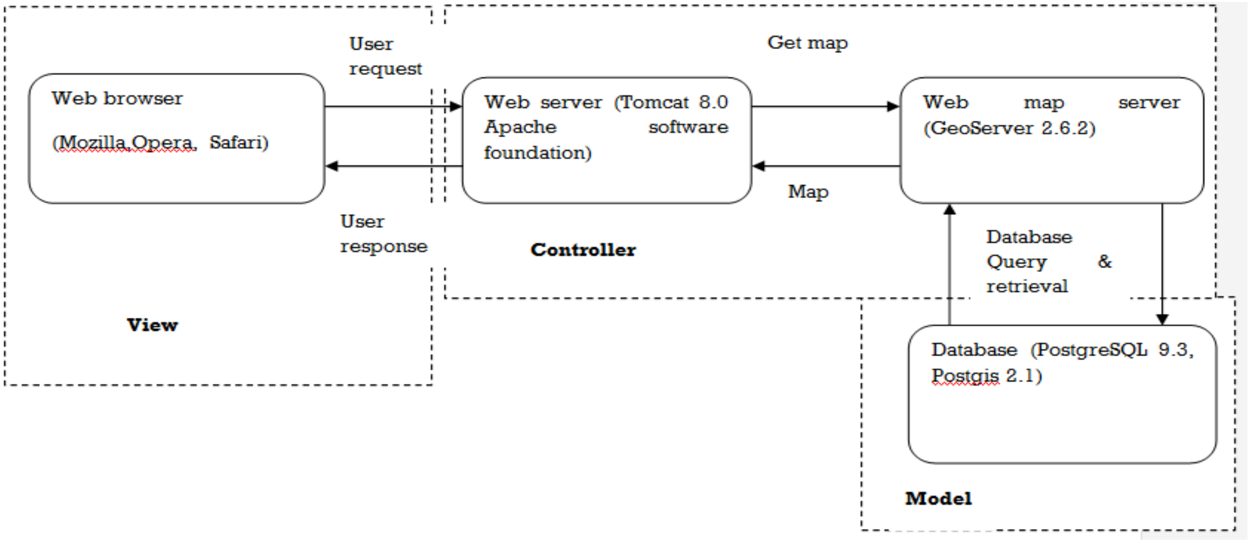
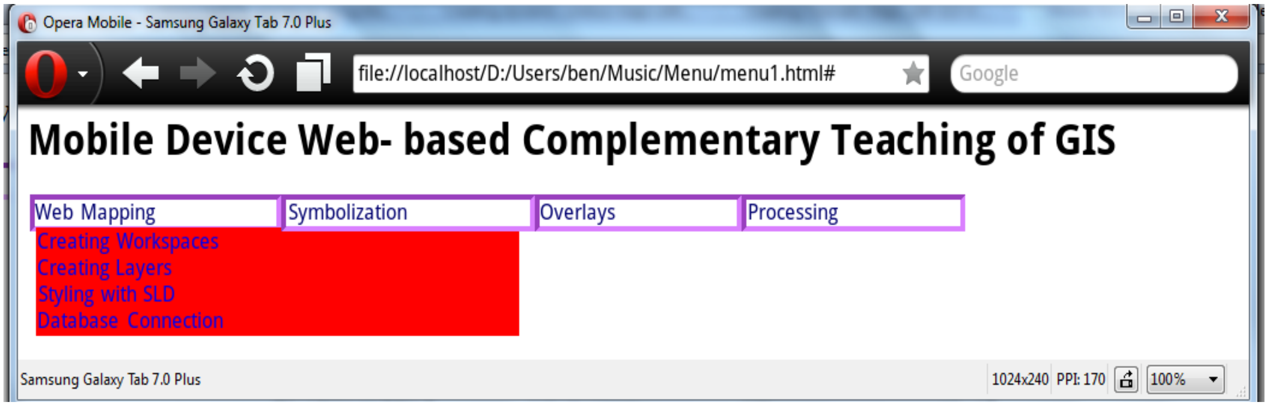

User request
Get map
Web browser
(Mozilla Opera, Safari)
Web server (Tomcat 8.0 Apache software foundation)
Web map server (GeoServer 2.6.2)
Map
User response
Controller
Database Query & retrieval
View
Database (PostgreSQL 9.3, Postgis 2.1)
Model

Figure 3.1: Typical client-server architecture

3.2 Applications user interface

Figure 3.2 is the user interface that the user uses to interact with in order to access the various system functionalities.


Opera Mobile - Samsung Galaxy Tab 7.0 Plus
file://localhost/D:/Users/ben/Music/Menu/menu1.html#
Google
Samsung Galaxy Tab 7.0 Plus
1024x240 PPI: 170
100%

Mobile Device Web- based Complementary Teaching of GIS

Web Mapping | Symbolization | Overlays | Processing

Creating Workspaces
Creating Layers
Styling with SLD
Database Connection

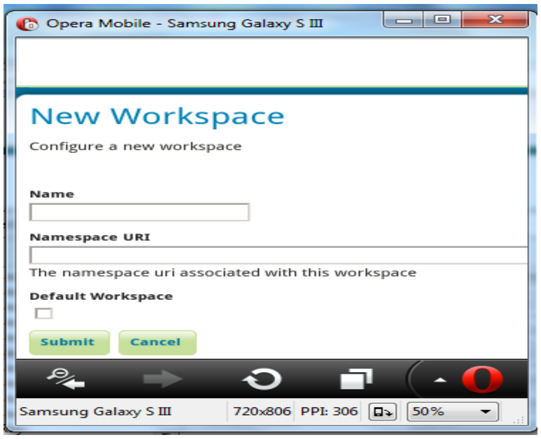
Figure 3.2: User interface on Samsung galaxy tab 7.0 plus

Web applications unlike desktop applications execute on a server far away (probably a continent away) from the user's device. Google's search engine has only one textbox and two buttons. Bandwidth speed is of the essence here. Processing takes place on a server which could hence design has to take care of the server response time. One of the aims of this application is to take advantage of the lecturer's 10 minutes before and after a lecture to show the progress of one's assignment for feedback purposes. Consequently, the speed of both request and response in the developed application is of the essence. The application is based on thin client architecture since all mobile devices have browsers. So the user needs not install any software to avoid compatibility issues with current and future mobile devices.

3.3 Description of key menu features as shown on the user interface diagram.

3.3.1 Workspace

A workspace (sometimes referred to as a namespace) is the name for a notional container for grouping similar data together. It is designed to be a separate, isolated space relating to a certain project (see Figure 3.3). Using workspaces, it is possible to use layers with identical names without creating conflicts. Workspaces are usually denoted by a prefix to a layer name or store name. For example, a layer called streets with a workspace prefix called Kibera would be referred to by Kibera:streets. This would not conflict with another layer called streets in another workspace called dc (dc: streets). Stores and layers must all have an associated workspace. In general, styles may optionally be associated with a workspace, but can also be global.



Opera Mobile - Samsung Galaxy S III

New Workspace

Configure a new workspace

Name

Namespace URI

The namespace uri associated with this workspace

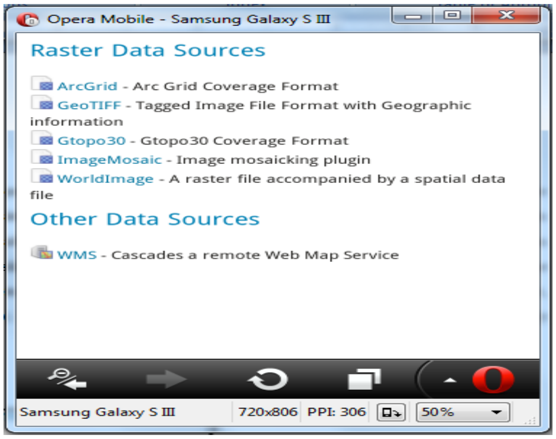
Default Workspace

□

Submit Cancel

Samsung Galaxy S III 720x806 PPI: 306 50%

Figure 3.3: Workspace window



Opera Mobile - Samsung Galaxy S III

Raster Data Sources

- ArcGrid - Arc Grid Coverage Format
- GeoTIFF - Tagged Image File Format with Geographic information
- Gtopo30 - Gtopo30 Coverage Format
- ImageMosaic - Image mosaicking plugin
- WorldImage - A raster file accompanied by a spatial data file

Other Data Sources

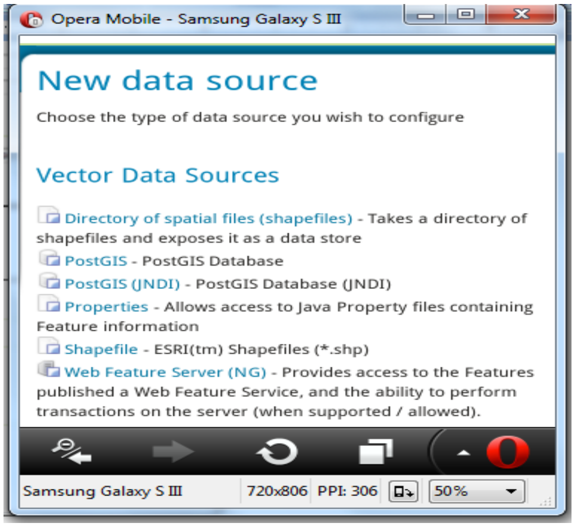
- WMS - Cascades a remote Web Map Service

Samsung Galaxy S III 720x806 PPI: 306 50%

Figure 3.4: Raster store window

3.3.2 Store

A store is the name for a container of geographic data. A store refers to a specific data source, be it a shapefile, database, or any other data source that GeoServer supports. A store can contain many layers, such as the case of a database that contains many tables as shown in Figures 3.5 and 3.6. A store can also have a single layer, such as in the case of a shapefile or GeoTIFF. A store must contain at least one layer. GeoServer saves the connection parameters to each store (the path to the shapefile, credentials to connect to the database). Each store must also be associated with one (and only one) workspace.



New data source

Choose the type of data source you wish to configure

Vector Data Sources

Directory of spatial files (shapefiles) - Takes a directory of shapefiles and exposes it as a data store

PostGIS - PostGIS Database

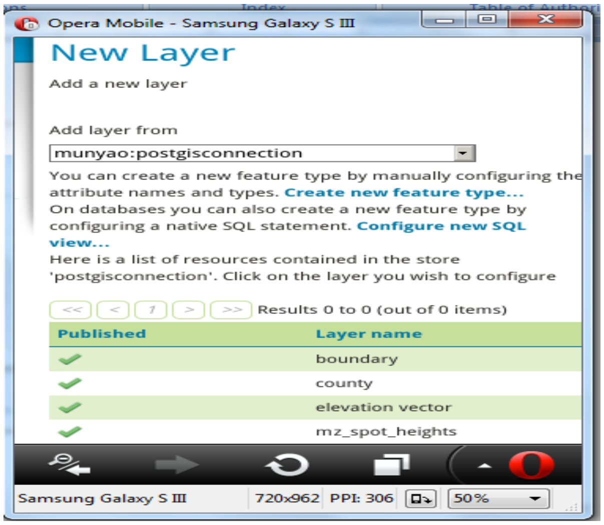
PostGIS (JNDI) - PostGIS Database (JNDI)

Properties - Allows access to Java Property files containing Feature information

Shapefile - ESRI(tm) Shapefiles (*.shp)

Web Feature Server (NG) - Provides access to the Features published a Web Feature Service, and the ability to perform transactions on the server (when supported / allowed).

Figure 3.5: Vector data store window



New Layer

Add a new layer

Add layer from

munyao:postgisconnection

You can create a new feature type by manually configuring the attribute names and types. Create new feature type...

On databases you can also create a new feature type by configuring a native SQL statement. Configure new SQL view...

Here is a list of resources contained in the store 'postgisconnection'. Click on the layer you wish to configure

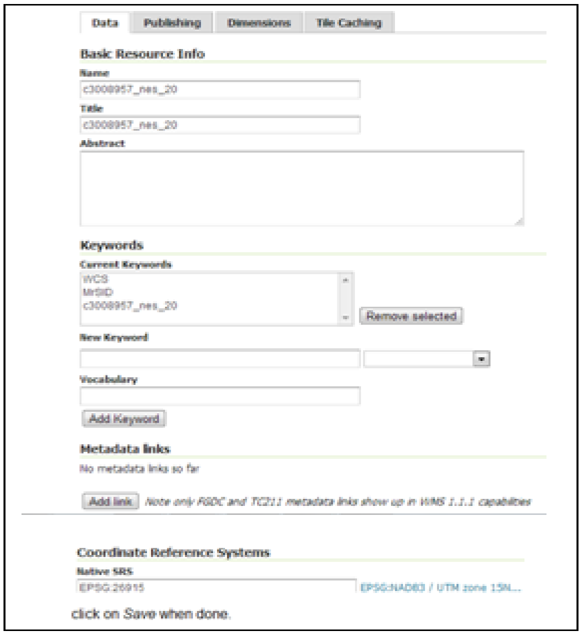
Results 0 to 0 (out of 0 items)

Published	Layer name
✓	boundary
✓	county
✓	elevation vector
✓	mz_spot_heights

Figure 3.6: New layer page

3.3.3 Layer

A layer (sometimes known as a feature type) is a collection of geospatial features or a coverage. Typically a layer contains one type of data (points, lines, polygons, raster) and has a single identifiable subject (streets, houses, country boundaries, etc.) as illustrated in Figure 3.7 and 3.8. A layer corresponds to a table or view from a database, or an individual file. GeoServer stores information associated with a layer, such as projection information, bounding box, and associated styles. Each layer must be associated with one (and only one) workspace.



Basic Resource Info

Name: c3008957_nes_20

Title: c3008957_nes_20

Abstract

Keywords

Current Keywords: WCS, MrSID, c3008957_nes_20

New Keyword

Vocabulary

Metadata links

No metadata links so far

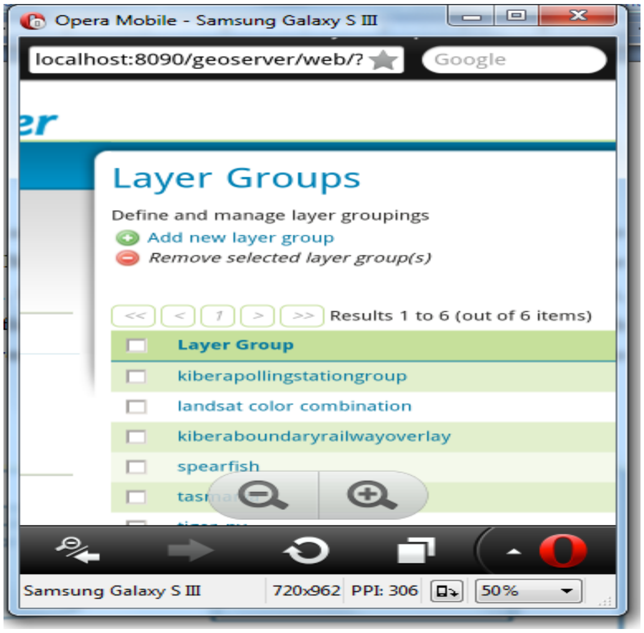
Note only FGDC and TC211 metadata links show up in WMS 1.1.1 capabilities

Coordinate Reference Systems

Native SRS: EPSG:26915 EPSG:NAD83 / UTM zone 15N...

click on *Save* when done.

Figure 3.7: Edit layer page



localhost:8090/geoserver/web/?

Layer Groups

Define and manage layer groupings

Add new layer group

Remove selected layer group(s)

Results 1 to 6 (out of 6 items)

	Layer Group
□	kiberapollingstationgroup
□	landsat color combination
□	kiberaboundaryrailwayoverlay
□	spearfish
□	tasma

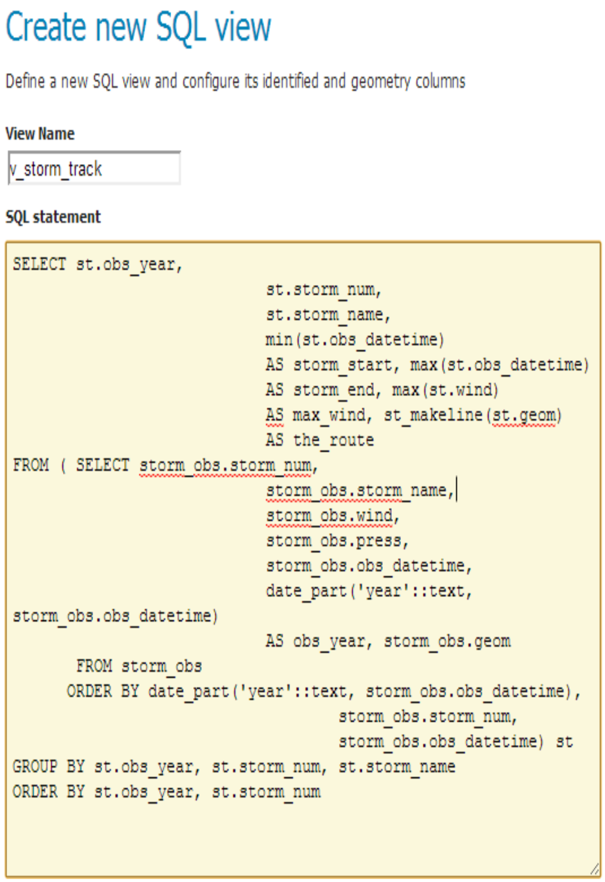
Figure 3.8: Layer group page

3.3.4 Layer group

A layer group, as its name suggests, is a collection of layers. A layer group makes it possible to request multiple layers with a single WMS request. A layer group contains information about the layers that comprise the layer group, the order in which they are rendered, the projection, associated styles, and more (see Figure 3.7). This information can be different from the defaults for each individual layer. Layer groups do not respect the concept of workspace, and are relevant only to WMS requests as shown in Figure 3.8.

3.3.5 Adding a SQL Parametric View based Layer

The traditional way to use database backed data is to configure either a table or a database view as a new layer in GeoServer Selecting the link Configure new SQL view will open a new page where the SQL statement can be specified as displayed in Figure 3.9.



Create new SQL view

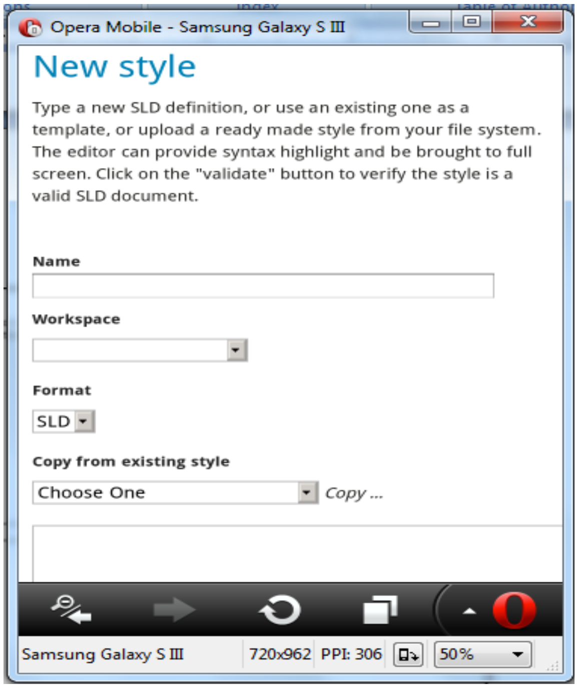
Define a new SQL view and configure its identified and geometry columns

View Name

v_storm_track

SQL statement

```
SELECT st.obs_year,
                            st.storm_num,
                            st.storm_name,
                            min(st.obs_datetime)
                            AS storm_start, max(st.obs_datetime)
                            AS storm_end, max(st.wind)
                            AS max_wind, st_makeline(st.geom)
                            AS the_route
FROM ( SELECT storm_obs.storm_num,
                            storm_obs.storm_name,
                            storm_obs.wind,
                            storm_obs.press,
                            storm_obs.obs_datetime,
                            date_part('year'::text,
storm_obs.obs_datetime)
                            AS obs_year, storm_obs.geom
       FROM storm_obs
      ORDER BY date_part('year'::text, storm_obs.obs_datetime),
                                    storm_obs.storm_num,
                                    storm_obs.obs_datetime) st
GROUP BY st.obs_year, st.storm_num, st.storm_name
ORDER BY st.obs_year, st.storm_num
```



Opera Mobile - Samsung Galaxy S III

New style

Type a new SLD definition, or use an existing one as a template, or upload a ready made style from your file system. The editor can provide syntax highlight and be brought to full screen. Click on the "validate" button to verify the style is a valid SLD document.

Name

Workspace

Format

SLD

Copy from existing style

Choose One Copy ...

Samsung Galaxy S III 720x962 PPI: 306 50%

Figure 3.9: Sql view page

Figure 3.10: Style layer page on Samsung Galaxy s III

3.3.6 Adding a Style

Perhaps the most important function of a web map server is the ability to style and render data. This section covers the task of adding a new style to Geo Server and configuring the default style for a particular layer[22].

A style is a visualization directive for rendering geographic data. A style can contain rules for color, shape, and size, along with logic for styling certain features or points in certain ways based on attributes or scale level. Every layer must be associated with at least one style as shown in Figure 3.10. GeoServer recognizes styles in Styled Layer Descriptor (SLD) format.

[22] http://geoserver.org/ (last accessed 3rd May 2015)

3.4 Sample style code Inside the Style Editor

```
<?xml version="1.0" encoding="UTF-8"?>
<sld:StyledLayerDescriptor xmlns="http://www.opengis.net/sld"
        xmlns:sld="http://www.opengis.net/sld"
        xmlns:ogc="http://www.opengis.net/ogc"
        xmlns:gml="http://www.opengis.net/gml"
        version="1.0.0">
  <sld:UserLayer>
        <sld:LayerFeatureConstraints>
          <sld:FeatureTypeConstraint/>
        </sld:LayerFeatureConstraints>
        <sld:UserStyle>
          <sld:Name>BoulderCityLimits</sld:Name>
          <sld:Title/>
          <sld:IsDefault>1</sld:IsDefault>
          <sld:FeatureTypeStyle>
                <sld:Name>group 0</sld:Name>
                <sld:FeatureTypeName>Feature</sld:FeatureTypeName>
                <sld:SemanticTypeIdentifier>generic:geometry</sld:SemanticTypeIdentifier>
                <sld:SemanticTypeIdentifier>simple</sld:SemanticTypeIdentifier>
                <sld:Rule>
                  <sld:Name>Filled</sld:Name>
                  <sld:MinScaleDenominator>75000</sld:MinScaleDenominator>
                  <sld:PolygonSymbolizer>
                        <sld:Fill>
                          <sld:CssParameter name="fill">#7F7F7F</sld:CssParameter>
                          <sld:CssParameter name="fill-opacity">0.5</sld:CssParameter>
                        </sld:Fill>
                        <sld:Stroke>
                          <sld:CssParameter name="stroke">#7F7F7F</sld:CssParameter>
                          <sld:CssParameter name="stroke-opacity">0.5</sld:CssParameter>
                          <sld:CssParameter name="stroke-width">2.0</sld:CssParameter>
                        </sld:Stroke>
                  </sld:PolygonSymbolizer>
                  <sld:TextSymbolizer>
                        <sld:Label>
                          <ogc:Literal>Boulder</ogc:Literal>
                        </sld:Label>
                        <sld:Font>
                          <sld:CssParameter name="font-family">Arial</sld:CssParameter>
                          <sld:CssParameter name="font-size">14.0</sld:CssParameter>
                          <sld:CssParameter name="font-style">normal</sld:CssParameter>
                          <sld:CssParameter name="font-weight">normal</sld:CssParameter>
                        </sld:Font>
                        <sld:LabelPlacement>
                          <sld:PointPlacement>
                <sld:AnchorPoint>
                 <sld:AnchorPointX>
                    <ogc:Literal>0.0</ogc:Literal>
                 </sld:AnchorPointX>
                 <sld:AnchorPointY>
                    <ogc:Literal>0.5</ogc:Literal>
                 </sld:AnchorPointY>
                </sld:AnchorPoint>
                <sld:Rotation>
                 <ogc:Literal>0</ogc:Literal>
                </sld:Rotation>
             </sld:PointPlacement>
            </sld:LabelPlacement>
            <sld:Fill>
             <sld:CssParameter name="fill">#000000</sld:CssParameter>
            </sld:Fill>
            <sld:VendorOption name="maxDisplacement">200</sld:VendorOption>
            <sld:VendorOption name="Group">true</sld:VendorOption>
         </sld:TextSymbolizer>
        </sld:Rule>
     </sld:FeatureTypeStyle>
    </sld:UserStyle>
 </sld:UserLayer>
</sld:StyledLayerDescriptor>
```

Explanation of the style SLD code

The most important sections are[23]:

The <Rule> tag combines a number of symbolizers (one has also the possibility to define the OGC filter) to define the portrayal of a feature.

The <PolygonSymbolizer> styles polygons and contain styling information about their border (stroke) and their fill.

The <TextSymbolizer > specifies text labels and their style:

<Label> specifies the content of the text label

<Font> specifies the font information for the labels.

[23] http://docs.geoserver.org/trunk/en/user/styling/sld (last accessed 3rd April 2015)

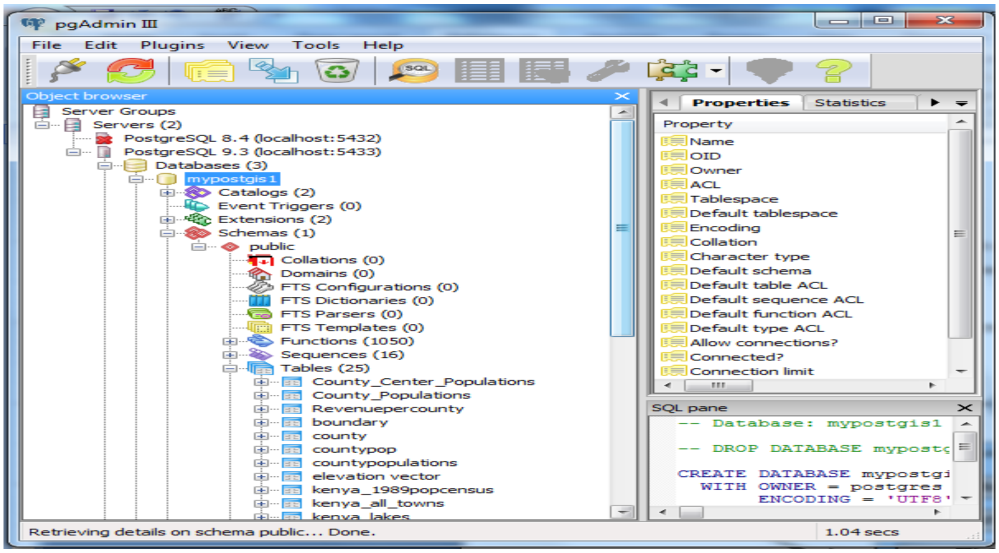
<LabelPlacement> sets the position of the label relative its associate feature.
<Fill> determines the fill color of the text label.

VendorOption maxDisplacement: This controls the displacement of the label along a line. Normally GeoServer would label a polygon in its centroid, provided the location is not busy with another label and that the label is not too big compare to the polygon, or not label it at all otherwise. When the maxDisplacement is set, the labeller will search for another location within maxDisplacement pixels from the pre-computed label point.

VendorOption Group: Sometimes one will have a set of related features that one only want a single label for. The grouping option group all features with the same label text, then finds a representative geometry for the group.

3.5 PostgreSQL - PostGIS database layout interface

PostgreSQL database has graphical administration tool known as pgAdmin that makes it easy to create and configure a database. Figure 3.11 is a screen shot of pgAdmin exposing part of the database tables have used for this project. PostgreSQL is installed first and the postGIS is installed as a spatial extension to PostgreSQL. Mypostgis1 database for this application has one schema and twenty-five tables. PgAdmin provides an interface to write select, delete, insert, update and edit queries.



```
-- Database: mypostgis1

-- DROP DATABASE mypostg

CREATE DATABASE mypostgi
  WITH OWNER = postgres
       ENCODING = 'UTF8'
```

Retrieving details on schema public... Done.   1.04 secs

Figure 3.11: Sample Postgres-postgis database layout interface using pgAdmin

Table 3.1 shows the FOSS tools selected for use in this study. Most FOSS software are only compatible with Linux operating system but all the software's covered here are compatible with windows 7 operating system.

Table 3.1: List of selected FOSS tools for developing the education system

Name	**Description**
PostgreSQL (and PostGIS)	A Database and its spatial extension
Quantum GIS	A complete Geographic Information System used to compare results from web application
Geoserver	An open-source server written in Java that allows users to share process and edit geospatial data.
OpenLayers	An open source AJAX library for accessing geographic data layers of all kinds
GeoTools	An open source GIS toolkit written in Java using Open Geospatial Consortium specifications.
R	A free software environment for statistical computing and graphics.
Opera Mobile Emulator	Used to generate browsers for a wide range of mobile devices
Tomcat	Open source web server by Apache software foundation

3.6 Data and their Sources

The datasets used in the study, their sources and characteristics are shown in Table 3.2.

Table 3.2: Data Sources

Data	**Format**	**Sources**
Kenya county boundaries dataset	Shape file	Survey of Kenya (downloaded)
County revenue allocation 2014/2015	Comma-separated values (CSV)	Kenya National Bureau of Statistics (downloaded)

CHAPTER 4: RESULTS AND ANALYSIS

This Chapter shows the results of the implementation on mobile devices. Screenshots of the user interfaces and the resulting layer visualizations are then described. The best way to demonstrate the success of the developed web based system would be redoing assignments covered during the course of our study especially Digital Cartography. Geospatial data has no intrinsic visual component. In order to see data, it must be styled. This means to specify color, thickness, and other visual attributes. All the code in this chapter is similar to the contents of section 3.4 which are well explained in section 3.4.1. To generate the map in figure 4.1 using figure 4.3 shape file:

- Click on the workspace menu item in figure 3.2 and provide an appropriate name then click submit
- Click on the database connection menu item in figure 3.2 to link to the county dataset
- Click on the layer menu item in figure 3.2 to create a layer for county dataset
- Click on the style menu item in figure 3.2 to open the style page
- Insert the code listing in section 4.1 in the editor section of the webpage and click the submit button
- Click on the layer menu item in figure 3.2 and link it to this style from the publish section of the webpage.
- Click on the preview layer section figure 3.2 to view this layer in a browser. A map like the one shown in figure 4.1 and 4.2 will be displayed.

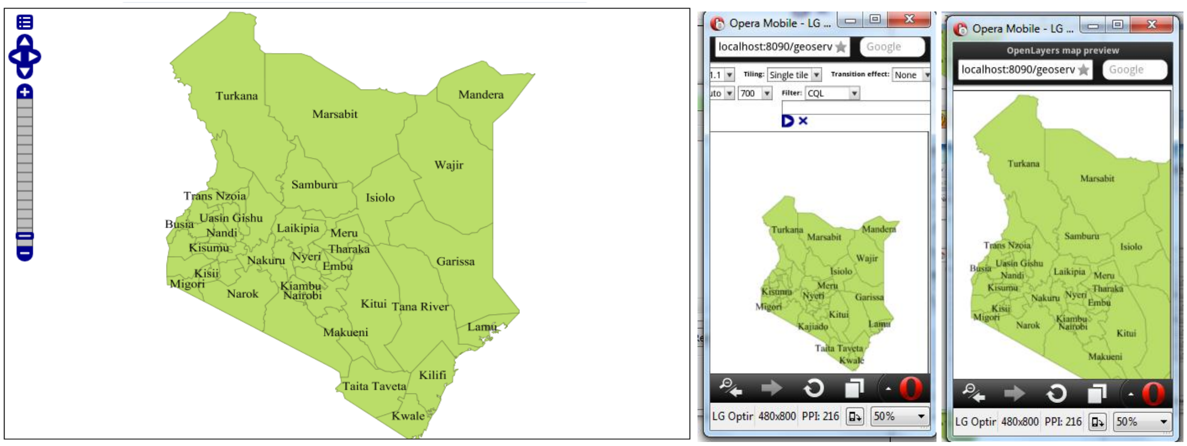
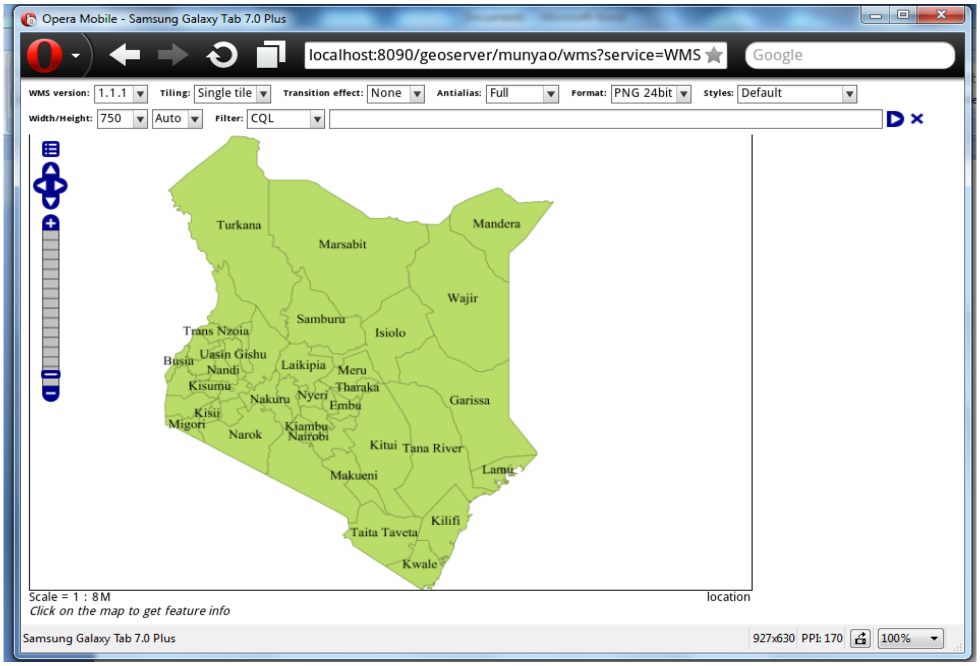


Figure 4.1: Kenya County boundary map on Samsung Galaxy SIII and LG Optimus 3D smartphones



Opera Mobile - Samsung Galaxy Tab 7.0 Plus

localhost:8090/geoserver/munyao/wms?service=WMS

WMS version: 1.1.1 Tiling: Single tile Transition effect: None Antialias: Full Format: PNG 24bit Styles: Default

Width/Height: 750 Auto Filter: CQL

Scale = 1 : 8M

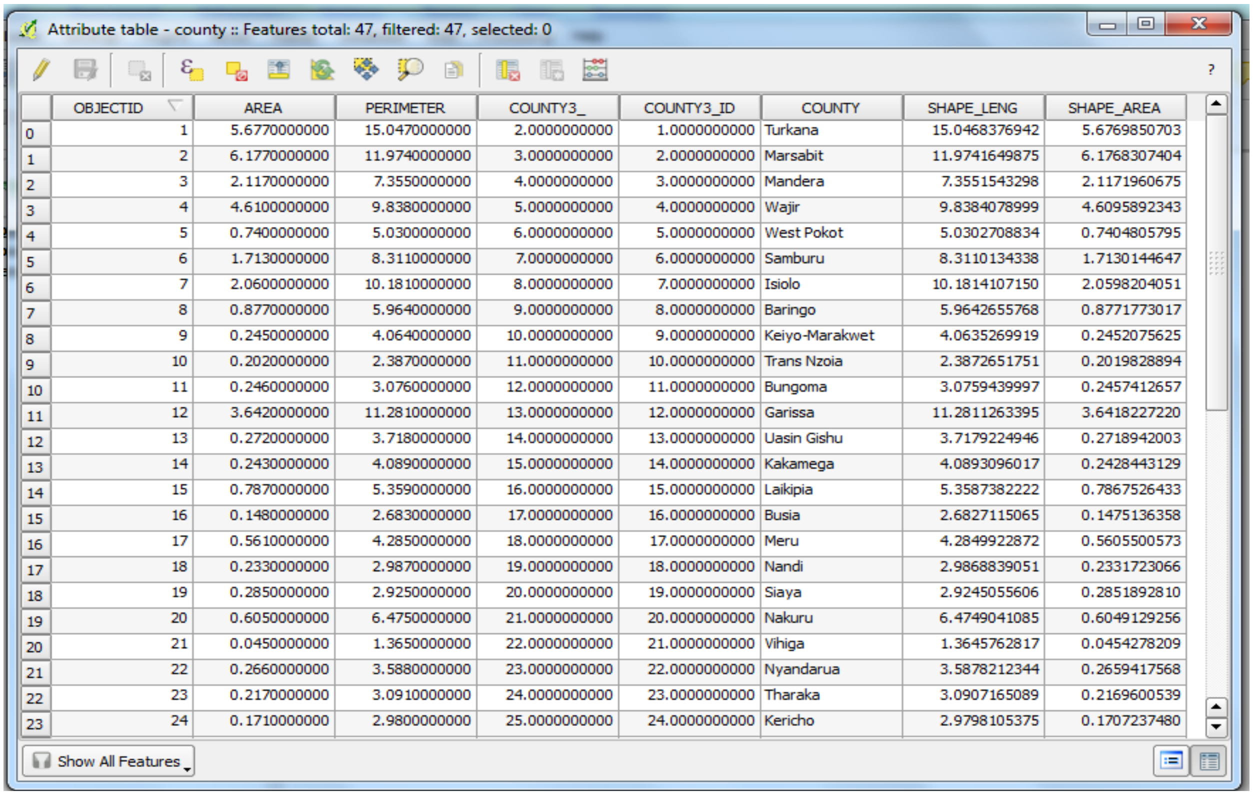
Click on the map to get feature info

location

Samsung Galaxy Tab 7.0 Plus

927x630 PPI: 170 100%

Figure 4.2: Kenya County boundary map on Samsung Galaxy Tab 7.0 Plus



Attribute table - county :: Features total: 47, filtered: 47, selected: 0

	OBJECTID	AREA	PERIMETER	COUNTY3_	COUNTY3_ID	COUNTY	SHAPE_LENG	SHAPE_AREA
0	1	5.6770000000	15.0470000000	2.0000000000	1.0000000000	Turkana	15.0468376942	5.6769850703
1	2	6.1770000000	11.9740000000	3.0000000000	2.0000000000	Marsabit	11.9741649875	6.1768307404
2	3	2.1170000000	7.3550000000	4.0000000000	3.0000000000	Mandera	7.3551543298	2.1171960675
3	4	4.6100000000	9.8380000000	5.0000000000	4.0000000000	Wajir	9.8384078999	4.6095892343
4	5	0.7400000000	5.0300000000	6.0000000000	5.0000000000	West Pokot	5.0302708834	0.7404805795
5	6	1.7130000000	8.3110000000	7.0000000000	6.0000000000	Samburu	8.3110134338	1.7130144647
6	7	2.0600000000	10.1810000000	8.0000000000	7.0000000000	Isiolo	10.1814107150	2.0598204051
7	8	0.8770000000	5.9640000000	9.0000000000	8.0000000000	Baringo	5.9642655768	0.8771773017
8	9	0.2450000000	4.0640000000	10.0000000000	9.0000000000	Keiyo-Marakwet	4.0635269919	0.2452075625
9	10	0.2020000000	2.3870000000	11.0000000000	10.0000000000	Trans Nzoia	2.3872651751	0.2019828894
10	11	0.2460000000	3.0760000000	12.0000000000	11.0000000000	Bungoma	3.0759439997	0.2457412657
11	12	3.6420000000	11.2810000000	13.0000000000	12.0000000000	Garissa	11.2811263395	3.6418227220
12	13	0.2720000000	3.7180000000	14.0000000000	13.0000000000	Uasin Gishu	3.7179224946	0.2718942003
13	14	0.2430000000	4.0890000000	15.0000000000	14.0000000000	Kakamega	4.0893096017	0.2428443129
14	15	0.7870000000	5.3590000000	16.0000000000	15.0000000000	Laikipia	5.3587382222	0.7867526433
15	16	0.1480000000	2.6830000000	17.0000000000	16.0000000000	Busia	2.6827115065	0.1475136358
16	17	0.5610000000	4.2850000000	18.0000000000	17.0000000000	Meru	4.2849922872	0.5605500573
17	18	0.2330000000	2.9870000000	19.0000000000	18.0000000000	Nandi	2.9868839051	0.2331723066
18	19	0.2850000000	2.9250000000	20.0000000000	19.0000000000	Siaya	2.9245055606	0.2851892810
19	20	0.6050000000	6.4750000000	21.0000000000	20.0000000000	Nakuru	6.4749041085	0.6049129256
20	21	0.0450000000	1.3650000000	22.0000000000	21.0000000000	Vihiga	1.3645762817	0.0454278209
21	22	0.2660000000	3.5880000000	23.0000000000	22.0000000000	Nyandarua	3.5878212344	0.2659417568
22	23	0.2170000000	3.0910000000	24.0000000000	23.0000000000	Tharaka	3.0907165089	0.2169600539
23	24	0.1710000000	2.9800000000	25.0000000000	24.0000000000	Kericho	2.9798105375	0.1707237480

Figure 4.3: Sample data for Kenya counties shape file from postgreSQL-PostGIS database

4.1 Style layer description code for labeled Kenya 47 counties

```
<?xml version="1.0" encoding="UTF-8"?>
<StyledLayerDescriptor xmlns="http://www.opengis.net/sld" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.1.0"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/sld
http://schemas.opengis.net/sld/1.1.0/StyledLayerDescriptor.xsd" xmlns:se="http://www.opengis.net/se">
  <NamedLayer>
    <se:Name>county</se:Name>
    <UserStyle>
      <se:Name>county</se:Name>
      <se:FeatureTypeStyle>
        <se:Rule>
          <se:Name>Single symbol</se:Name>
          <se:PolygonSymbolizer>
            <se:Fill>
              <se:SvgParameter name="fill">#badd69</se:SvgParameter>
            </se:Fill>
            <se:Stroke>
              <se:SvgParameter name="stroke">#809848</se:SvgParameter>
              <se:SvgParameter name="stroke-width">0.26</se:SvgParameter>
              <se:SvgParameter name="stroke-linejoin">bevel</se:SvgParameter>
            </se:Stroke>
          </se:PolygonSymbolizer>
        </se:Rule>
        <Rule>
        <TextSymbolizer>
            <Label>
              <ogc:PropertyName>county</ogc:PropertyName>
            </Label>
            <Font>
              <CssParameter name="font-family">Times New Roman</CssParameter>
              <CssParameter name="font-style">Normal</CssParameter>
              <CssParameter name="font-size">14</CssParameter>
            </Font>
            <LabelPlacement>
              <PointPlacement>
                <AnchorPoint>
                  <AnchorPointX>0.5</AnchorPointX>
                  <AnchorPointY>0.5</AnchorPointY>
                </AnchorPoint>
              </PointPlacement>
            </LabelPlacement>
          </TextSymbolizer>
        </Rule>
      </se:FeatureTypeStyle>
    </UserStyle>
  </NamedLayer>
</StyledLayerDescriptor>
```

4.2 Thematic maps

Thematic maps classify continuous data into different colors, to display numeric information visually. GeoServer already has the ability to dynamically generate SLD documents from data sources, with support for equal intervals and quartiles, and creating color ramps using a number of categories supplied by the user. Styling interfaces for thematic mapping require the classification of data, and the assignment of styles to each classification. Unique Value thematic maps use different colors or symbols to shade or represent objects with different values for an attribute. Objects with the same unique value of this attribute are assigned the same color or symbol. Section 4.3 provides the SLD code for thematic maps shown in figure 4.4, 4.5 and 4.6. This code template can be modified to generate a different type of thematic map rather than having to start from scratch. A join query that combines data in figure 4.7 written in SQL view section of figure 3.9 has been used. The code is inserted in style section of GEO Server as shown in figure 3.10.

- Within the same workspace created above create a view that joins revenue allocation CSV dataset to the county shape file dataset.
- Create a new style and insert the code listing in section 4.3
- Create a new layer based on this view and link it with the new style in the publish section
- Open print preview menu item and click on this layer to view it on the browser using open layers.

Note: deleting the section within the label placement of the SLD styles a map with no labels.

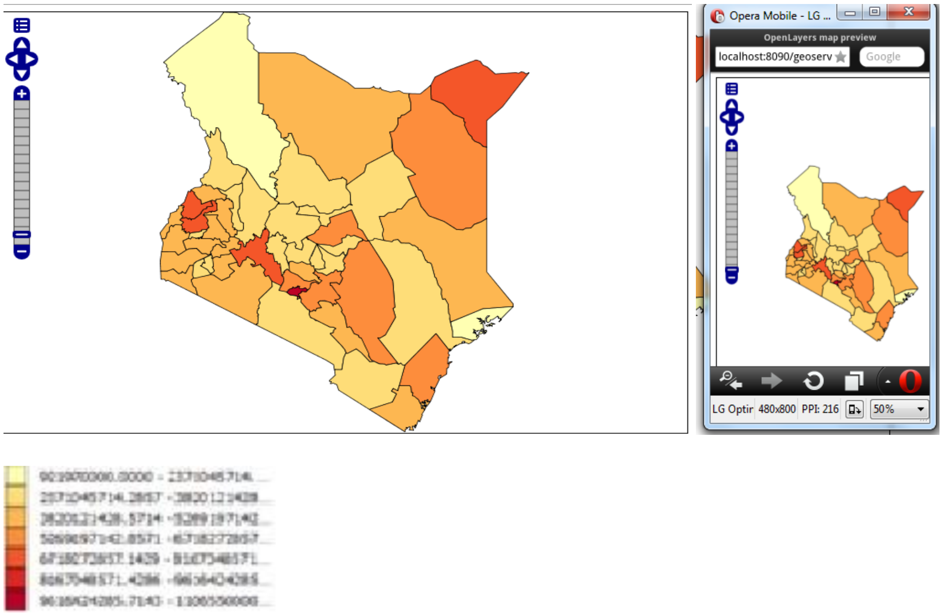


Figure 4.4: Unlabeled Kenya County revenue allocations 2014/2015 thematic map

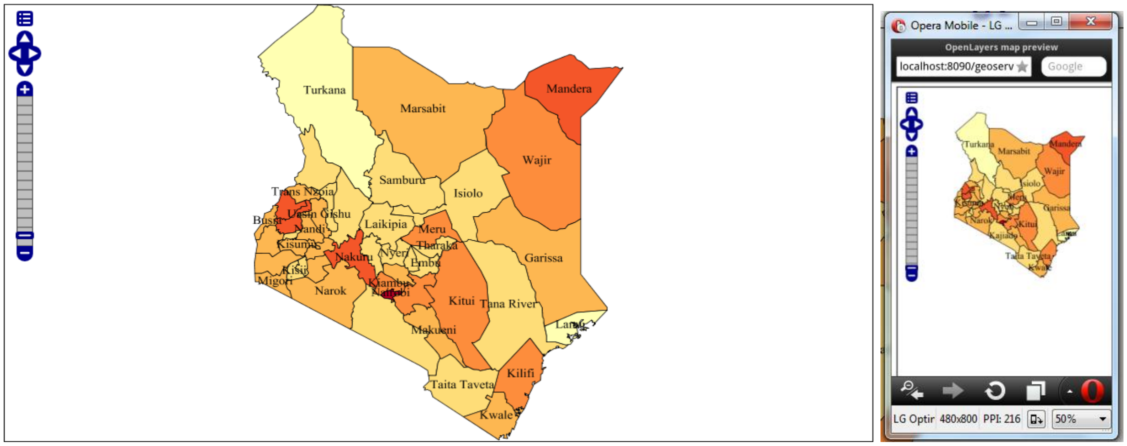


Figure 4.5: Label styled Kenya County revenue allocations 2014/2015 thematic map

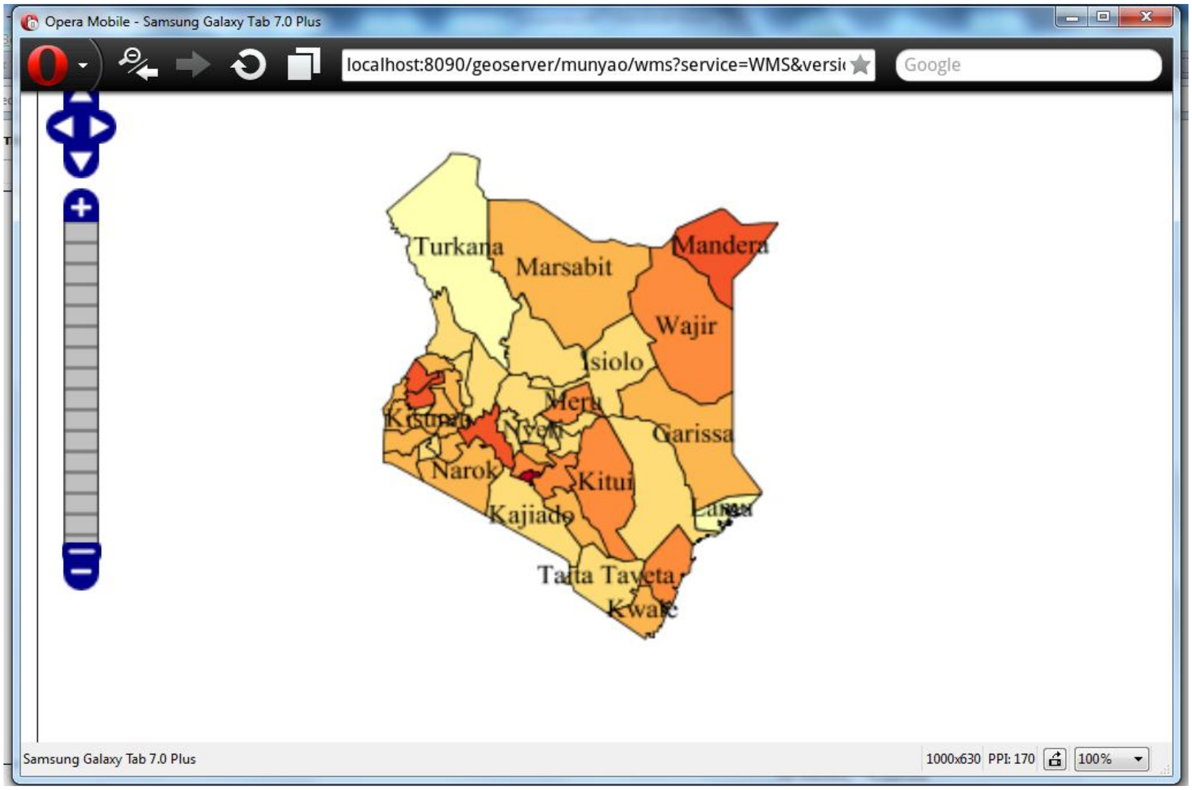
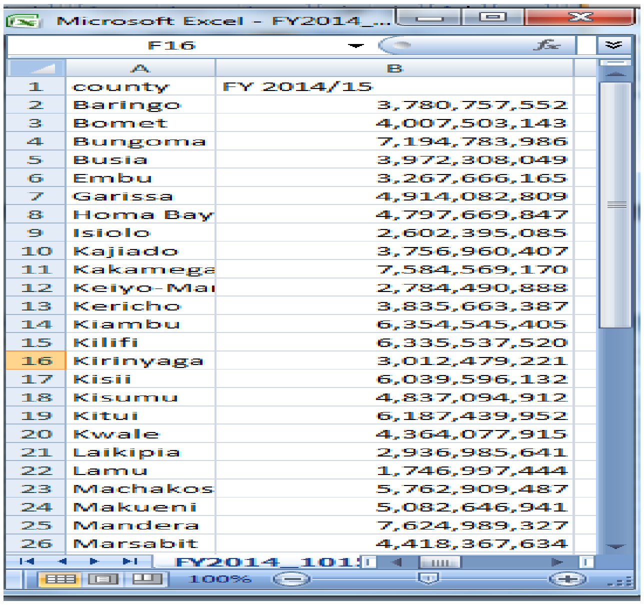

Opera Mobile - Samsung Galaxy Tab 7.0 Plus
localhost:8090/geoserver/munyao/wms?service=WMS&versi
Google
Turkana
Marsabit
Mandera
Wajir
Isiolo
Meru
Kisumu
Nyeri
Garissa
Narok
Kitui
Lamu
Kajiado
Taita Taveta
Kwale
Samsung Galaxy Tab 7.0 Plus
1000x630 PPI: 170
100%

Figure 4.6: Labeled Kenya County revenue allocations 2014/2015 thematic map



	A	B
1	county	FY 2014/15
2	Baringo	3,780,757,552
3	Bomet	4,007,503,143
4	Bungoma	7,194,783,986
5	Busia	3,972,308,049
6	Embu	3,267,666,165
7	Garissa	4,914,082,809
8	Homa Bay	4,797,669,847
9	Isiolo	2,602,395,085
10	Kajiado	3,756,960,407
11	Kakamega	7,584,569,170
12	Keiyo-Mai	2,784,490,888
13	Kericho	3,835,663,387
14	Kiambu	6,354,545,405
15	Kilifi	6,335,537,520
16	Kirinyaga	3,012,479,221
17	Kisii	6,039,596,132
18	Kisumu	4,837,094,912
19	Kitui	6,187,439,952
20	Kwale	4,364,077,915
21	Laikipia	2,936,985,641
22	Lamu	1,746,997,444
23	Machakos	5,762,909,487
24	Makueni	5,082,646,941
25	Mandera	7,624,989,327
26	Marsabit	4,418,367,634

Figure 4.7: Sample data 2014/2015 for county revenue allocations (Source: Kenya National Bureau of Statistics)

4.3 Style layer description code for county revenue allocation thematic map

```
<?xml version="1.0" encoding="UTF-8"?>
<StyledLayerDescriptor xmlns="http://www.opengis.net/sld" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.1.0"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/sld
http://schemas.opengis.net/sld/1.1.0/StyledLayerDescriptor.xsd" xmlns:se="http://www.opengis.net/se">
 <NamedLayer>
  <se:Name>county</se:Name>
  <UserStyle>
   <se:Name>county</se:Name>
   <se:FeatureTypeStyle>
    <se:Rule>
     <se:Name> 921870000.0000 - 2371045714.2857 </se:Name>
     <se:Description>
      <se:Title> 921870000.0000 - 2371045714.2857 </se:Title>
     </se:Description>
     <ogc:Filter xmlns:ogc="http://www.opengis.net/ogc">
      <ogc:And>
       <ogc:PropertyIsGreaterThan>
        <ogc:PropertyName>revenue</ogc:PropertyName>
        <ogc:Literal>9.2187e+08</ogc:Literal>
       </ogc:PropertyIsGreaterThan>
       <ogc:PropertyIsLessThanOrEqualTo>
        <ogc:PropertyName>revenue</ogc:PropertyName>
        <ogc:Literal>2.37105e+09</ogc:Literal>
       </ogc:PropertyIsLessThanOrEqualTo>
    </ogc:And>
  </ogc:Filter>
  <se:PolygonSymbolizer>
    <se:Fill>
      <se:SvgParameter name="fill">#ffffb2</se:SvgParameter>
    </se:Fill>
    <se:Stroke>
      <se:SvgParameter name="stroke">#000000</se:SvgParameter>
      <se:SvgParameter name="stroke-width">0.26</se:SvgParameter>
      <se:SvgParameter name="stroke-linejoin">bevel</se:SvgParameter>
    </se:Stroke>
  </se:PolygonSymbolizer>
</se:Rule>
<se:Rule>
  <se:Name> 2371045714.2857 - 3820121428.5714 </se:Name>
  <se:Description>
    <se:Title> 2371045714.2857 - 3820121428.5714 </se:Title>
  </se:Description>
  <ogc:Filter xmlns:ogc="http://www.opengis.net/ogc">
    <ogc:And>
      <ogc:PropertyIsGreaterThan>
        <ogc:PropertyName>revenue</ogc:PropertyName>
        <ogc:Literal>2.37105e+09</ogc:Literal>
      </ogc:PropertyIsGreaterThan>
      <ogc:PropertyIsLessThanOrEqualTo>
        <ogc:PropertyName>revenue</ogc:PropertyName>
        <ogc:Literal>3.82012e+09</ogc:Literal>
      </ogc:PropertyIsLessThanOrEqualTo>
    </ogc:And>
  </ogc:Filter>
  <se:PolygonSymbolizer>
    <se:Fill>
      <se:SvgParameter name="fill">#fedd78</se:SvgParameter>
    </se:Fill>
    <se:Stroke>
      <se:SvgParameter name="stroke">#000000</se:SvgParameter>
      <se:SvgParameter name="stroke-width">0.26</se:SvgParameter>
      <se:SvgParameter name="stroke-linejoin">bevel</se:SvgParameter>
    </se:Stroke>
  </se:PolygonSymbolizer>
</se:Rule>
<se:Rule>
  <se:Name> 3820121428.5714 - 5269197142.8571 </se:Name>
  <se:Description>
    <se:Title> 3820121428.5714 - 5269197142.8571 </se:Title>
  </se:Description>
  <ogc:Filter xmlns:ogc="http://www.opengis.net/ogc">
    <ogc:And>
      <ogc:PropertyIsGreaterThan>
```

```
        <ogc:PropertyName>revenue</ogc:PropertyName>
        <ogc:Literal>3.82012e+09</ogc:Literal>
      </ogc:PropertyIsGreaterThan>
      <ogc:PropertyIsLessThanOrEqualTo>
        <ogc:PropertyName>revenue</ogc:PropertyName>
        <ogc:Literal>5.2692e+09</ogc:Literal>
      </ogc:PropertyIsLessThanOrEqualTo>
    </ogc:And>
  </ogc:Filter>
  <se:PolygonSymbolizer>
    <se:Fill>
      <se:SvgParameter name="fill">#fdb751</se:SvgParameter>
    </se:Fill>
    <se:Stroke>
      <se:SvgParameter name="stroke">#000000</se:SvgParameter>
      <se:SvgParameter name="stroke-width">0.26</se:SvgParameter>
      <se:SvgParameter name="stroke-linejoin">bevel</se:SvgParameter>
    </se:Stroke>
  </se:PolygonSymbolizer>
</se:Rule>
<se:Rule>
  <se:Name> 5269197142.8571 - 6718272857.1429 </se:Name>
  <se:Description>
    <se:Title> 5269197142.8571 - 6718272857.1429 </se:Title>
  </se:Description>
  <ogc:Filter xmlns:ogc="http://www.opengis.net/ogc">
    <ogc:And>
      <ogc:PropertyIsGreaterThan>
        <ogc:PropertyName>revenue</ogc:PropertyName>
        <ogc:Literal>5.2692e+09</ogc:Literal>
      </ogc:PropertyIsGreaterThan>
      <ogc:PropertyIsLessThanOrEqualTo>
        <ogc:PropertyName>revenue</ogc:PropertyName>
        <ogc:Literal>6.71827e+09</ogc:Literal>
      </ogc:PropertyIsLessThanOrEqualTo>
    </ogc:And>
  </ogc:Filter>
  <se:PolygonSymbolizer>
    <se:Fill>
      <se:SvgParameter name="fill">#fd8d3c</se:SvgParameter>
    </se:Fill>
    <se:Stroke>
      <se:SvgParameter name="stroke">#000000</se:SvgParameter>
      <se:SvgParameter name="stroke-width">0.26</se:SvgParameter>
      <se:SvgParameter name="stroke-linejoin">bevel</se:SvgParameter>
    </se:Stroke>
  </se:PolygonSymbolizer>
</se:Rule>
<se:Rule>
```

```
 <se:Name> 6718272857.1429 - 8167348571.4286 </se:Name>
 <se:Description>
  <se:Title> 6718272857.1429 - 8167348571.4286 </se:Title>
 </se:Description>
 <ogc:Filter xmlns:ogc="http://www.opengis.net/ogc">
  <ogc:And>
   <ogc:PropertyIsGreaterThan>
    <ogc:PropertyName>revenue</ogc:PropertyName>
    <ogc:Literal>6.71827e+09</ogc:Literal>
   </ogc:PropertyIsGreaterThan>
   <ogc:PropertyIsLessThanOrEqualTo>
    <ogc:PropertyName>revenue</ogc:PropertyName>
    <ogc:Literal>8.16735e+09</ogc:Literal>
   </ogc:PropertyIsLessThanOrEqualTo>
  </ogc:And>
 </ogc:Filter>
 <se:PolygonSymbolizer>
  <se:Fill>
   <se:SvgParameter name="fill">#f45629</se:SvgParameter>
  </se:Fill>
  <se:Stroke>
   <se:SvgParameter name="stroke">#000000</se:SvgParameter>
   <se:SvgParameter name="stroke-width">0.26</se:SvgParameter>
   <se:SvgParameter name="stroke-linejoin">bevel</se:SvgParameter>
  </se:Stroke>
 </se:PolygonSymbolizer>
</se:Rule>
<se:Rule>
 <se:Name> 8167348571.4286 - 9616424285.7143 </se:Name>
 <se:Description>
  <se:Title> 8167348571.4286 - 9616424285.7143 </se:Title>
 </se:Description>
 <ogc:Filter xmlns:ogc="http://www.opengis.net/ogc">
  <ogc:And>
   <ogc:PropertyIsGreaterThan>
    <ogc:PropertyName>revenue</ogc:PropertyName>
    <ogc:Literal>8.16735e+09</ogc:Literal>
   </ogc:PropertyIsGreaterThan>
   <ogc:PropertyIsLessThanOrEqualTo>
    <ogc:PropertyName>revenue</ogc:PropertyName>
    <ogc:Literal>9.61642e+09</ogc:Literal>
   </ogc:PropertyIsLessThanOrEqualTo>
  </ogc:And>
 </ogc:Filter>
 <se:PolygonSymbolizer>
  <se:Fill>
   <se:SvgParameter name="fill">#df2722</se:SvgParameter>
  </se:Fill>
  <se:Stroke>
```

```
   <se:SvgParameter name="stroke">#000000</se:SvgParameter>
   <se:SvgParameter name="stroke-width">0.26</se:SvgParameter>
   <se:SvgParameter name="stroke-linejoin">bevel</se:SvgParameter>
  </se:Stroke>
 </se:PolygonSymbolizer>
</se:Rule>
<se:Rule>
 <se:Name> 9616424285.7143 - 11065500000.0000 </se:Name>
 <se:Description>
  <se:Title> 9616424285.7143 - 11065500000.0000 </se:Title>
 </se:Description>
 <ogc:Filter xmlns:ogc="http://www.opengis.net/ogc">
  <ogc:And>
   <ogc:PropertyIsGreaterThan>
    <ogc:PropertyName>revenue</ogc:PropertyName>
    <ogc:Literal>9.61642e+09</ogc:Literal>
   </ogc:PropertyIsGreaterThan>
   <ogc:PropertyIsLessThanOrEqualTo>
    <ogc:PropertyName>revenue</ogc:PropertyName>
    <ogc:Literal>1.10655e+10</ogc:Literal>
   </ogc:PropertyIsLessThanOrEqualTo>
  </ogc:And>
 </ogc:Filter>
 <se:PolygonSymbolizer>
  <se:Fill>
   <se:SvgParameter name="fill">#bd0026</se:SvgParameter>
  </se:Fill>
  <se:Stroke>
   <se:SvgParameter name="stroke">#000000</se:SvgParameter>
   <se:SvgParameter name="stroke-width">0.26</se:SvgParameter>
   <se:SvgParameter name="stroke-linejoin">bevel</se:SvgParameter>
  </se:Stroke>
 </se:PolygonSymbolizer>
</se:Rule>
<Rule>
<TextSymbolizer>
  <Label>
   <ogc:PropertyName>county</ogc:PropertyName>
  </Label>
  <Font>
   <CssParameter name="font-family">Times New Roman</CssParameter>
   <CssParameter name="font-style">Normal</CssParameter>
   <CssParameter name="font-size">14</CssParameter>
  </Font>
  <LabelPlacement>
   <PointPlacement>
    <AnchorPoint>
     <AnchorPointX>0.5</AnchorPointX>
     <AnchorPointY>0.5</AnchorPointY>
```

```
              </AnchorPoint>
            </PointPlacement>
          </LabelPlacement>
        </TextSymbolizer>
      </Rule>
    </se:FeatureTypeStyle>
  </UserStyle>
 </NamedLayer>
</StyledLayerDescriptor>
```

4.4 Thematic map with graduated dot map overlaid on it

Figure 4.8 and 4.9 contains a thematic and graduated dot map overlaid on it. Point symbols with that have different sizes in proportion to the revenue have been used to create the graduated dot map. The same procedure as used in section 4.2 above has been followed to generate the maps. The aim is to provide different data visualization templates for the student.

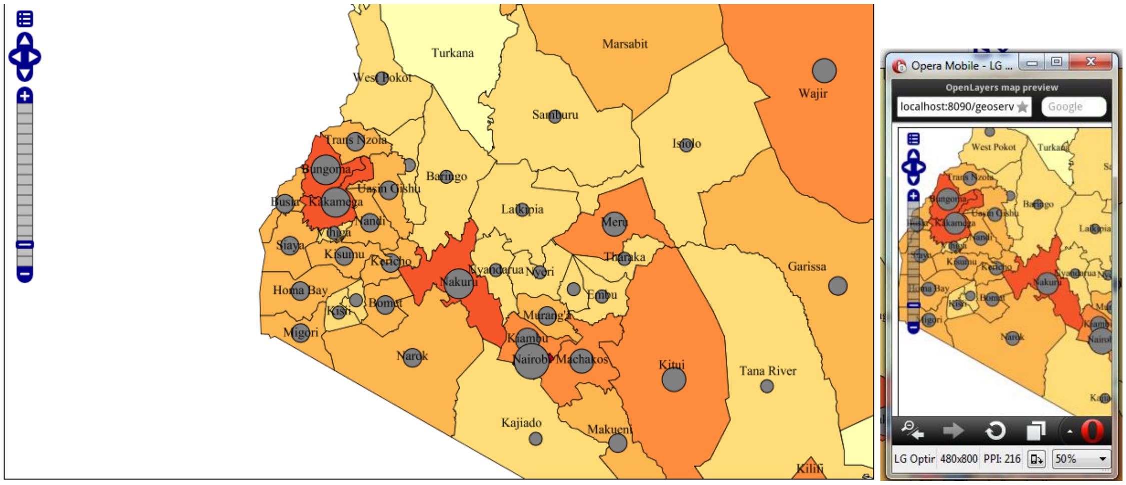


Figure 4.8: Zoomed in Kenya County revenue allocations 2014/2015 thematic map with point symbols

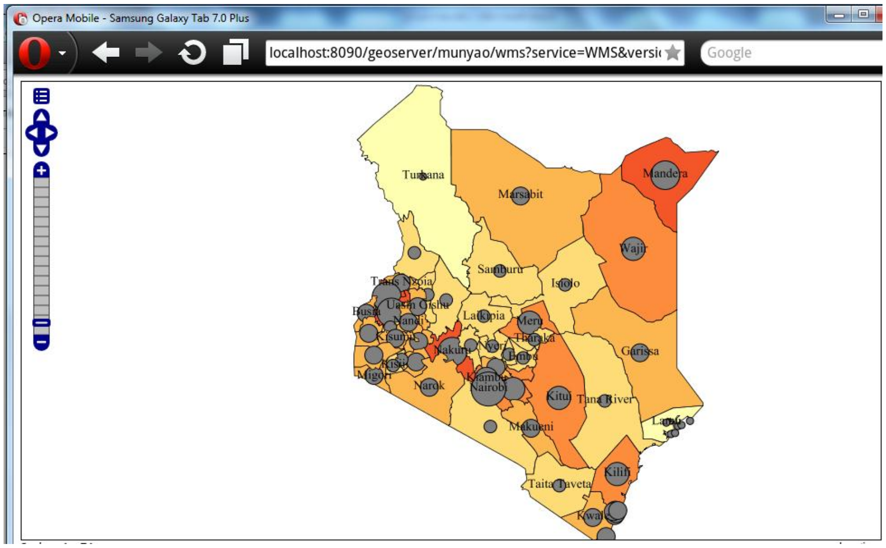


Figure 4.9: Kenya County revenue allocations 2014/2015 thematic map with point symbols

4.5 Style layer description code for county revenue allocation thematic map together with point symbolization

```
<?xml version="1.0" encoding="UTF-8"?>

<StyledLayerDescriptor xmlns="http://www.opengis.net/sld" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.1.0"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/sld
http://schemas.opengis.net/sld/1.1.0/StyledLayerDescriptor.xsd" xmlns:se="http://www.opengis.net/se">

 <NamedLayer>
  <se:Name>county</se:Name>
  <UserStyle>
   <se:Name>county</se:Name>
   <se:FeatureTypeStyle>
    <se:Rule>
     <se:Name> 921870000.0000 - 2371045714.2857 </se:Name>
     <se:Description>
      <se:Title> 921870000.0000 - 2371045714.2857 </se:Title>
     </se:Description>
     <ogc:Filter xmlns:ogc="http://www.opengis.net/ogc">
      <ogc:And>
       <ogc:PropertyIsGreaterThan>
        <ogc:PropertyName>revenue</ogc:PropertyName>
        <ogc:Literal>9.2187e+08</ogc:Literal>
       </ogc:PropertyIsGreaterThan>
       <ogc:PropertyIsLessThanOrEqualTo>
        <ogc:PropertyName>revenue</ogc:PropertyName>
        <ogc:Literal>2.37105e+09</ogc:Literal>
       </ogc:PropertyIsLessThanOrEqualTo>
      </ogc:And>
     </ogc:Filter>
   <se:PolygonSymbolizer>
     <se:Fill>
       <se:SvgParameter name="fill">#ffffb2</se:SvgParameter>
     </se:Fill>
     <se:Stroke>
       <se:SvgParameter name="stroke">#000000</se:SvgParameter>
       <se:SvgParameter name="stroke-width">0.26</se:SvgParameter>
       <se:SvgParameter name="stroke-linejoin">bevel</se:SvgParameter>
     </se:Stroke>
   </se:PolygonSymbolizer>
   <PointSymbolizer>
  <Graphic>
    <Mark>
      <WellKnownName>dot</WellKnownName>
      <Fill>
        <CssParameter name="fill">#0033CC</CssParameter>
      </Fill>
    </Mark>
    <Size>8</Size>
  </Graphic>
</PointSymbolizer>
 </se:Rule>
 <se:Rule>
   <se:Name> 2371045714.2857 - 3820121428.5714 </se:Name>
   <se:Description>
     <se:Title> 2371045714.2857 - 3820121428.5714 </se:Title>
   </se:Description>
   <ogc:Filter xmlns:ogc="http://www.opengis.net/ogc">
     <ogc:And>
       <ogc:PropertyIsGreaterThan>
         <ogc:PropertyName>revenue</ogc:PropertyName>
         <ogc:Literal>2.37105e+09</ogc:Literal>
       </ogc:PropertyIsGreaterThan>
       <ogc:PropertyIsLessThanOrEqualTo>
         <ogc:PropertyName>revenue</ogc:PropertyName>
         <ogc:Literal>3.82012e+09</ogc:Literal>
       </ogc:PropertyIsLessThanOrEqualTo>
     </ogc:And>
   </ogc:Filter>
   <se:PolygonSymbolizer>
     <se:Fill>
       <se:SvgParameter name="fill">#fedd78</se:SvgParameter>
     </se:Fill>
     <se:Stroke>
       <se:SvgParameter name="stroke">#000000</se:SvgParameter>
       <se:SvgParameter name="stroke-width">0.26</se:SvgParameter>
       <se:SvgParameter name="stroke-linejoin">bevel</se:SvgParameter>
     </se:Stroke>
   </se:PolygonSymbolizer>
   <PointSymbolizer>
  <Graphic>
    <Mark>
      <WellKnownName>dot</WellKnownName>
      <Fill>
        <CssParameter name="fill">#0033CC</CssParameter>
      </Fill>
    </Mark>
    <Size>14</Size>
  </Graphic>
</PointSymbolizer>
 </se:Rule>
 <se:Rule>
   <se:Name> 3820121428.5714 - 5269197142.8571 </se:Name>
   <se:Description>
     <se:Title> 3820121428.5714 - 5269197142.8571 </se:Title>
   </se:Description>
   <ogc:Filter xmlns:ogc="http://www.opengis.net/ogc">
     <ogc:And>
       <ogc:PropertyIsGreaterThan>
         <ogc:PropertyName>revenue</ogc:PropertyName>
         <ogc:Literal>3.82012e+09</ogc:Literal>
       </ogc:PropertyIsGreaterThan>
       <ogc:PropertyIsLessThanOrEqualTo>
         <ogc:PropertyName>revenue</ogc:PropertyName>
         <ogc:Literal>5.2692e+09</ogc:Literal>
       </ogc:PropertyIsLessThanOrEqualTo>
     </ogc:And>
   </ogc:Filter>
   <se:PolygonSymbolizer>
     <se:Fill>
       <se:SvgParameter name="fill">#fdb751</se:SvgParameter>
     </se:Fill>
     <se:Stroke>
       <se:SvgParameter name="stroke">#000000</se:SvgParameter>
       <se:SvgParameter name="stroke-width">0.26</se:SvgParameter>
       <se:SvgParameter name="stroke-linejoin">bevel</se:SvgParameter>
     </se:Stroke>
   </se:PolygonSymbolizer>
   <PointSymbolizer>
  <Graphic>
    <Mark>
      <WellKnownName>dot</WellKnownName>
      <Fill>
        <CssParameter name="fill">#0033CC</CssParameter>
      </Fill>
    </Mark>
    <Size>20</Size>
  </Graphic>
</PointSymbolizer>
 </se:Rule>
 <se:Rule>
   <se:Name> 5269197142.8571 - 6718272857.1429 </se:Name>
   <se:Description>
     <se:Title> 5269197142.8571 - 6718272857.1429 </se:Title>
   </se:Description>
   <ogc:Filter xmlns:ogc="http://www.opengis.net/ogc">
     <ogc:And>
       <ogc:PropertyIsGreaterThan>
         <ogc:PropertyName>revenue</ogc:PropertyName>
         <ogc:Literal>5.2692e+09</ogc:Literal>
       </ogc:PropertyIsGreaterThan>
       <ogc:PropertyIsLessThanOrEqualTo>
```

```
         <ogc:PropertyName>revenue</ogc:PropertyName>
         <ogc:Literal>6.71827e+09</ogc:Literal>
       </ogc:PropertyIsLessThanOrEqualTo>
     </ogc:And>
   </ogc:Filter>
   <se:PolygonSymbolizer>
     <se:Fill>
       <se:SvgParameter name="fill">#fd8d3c</se:SvgParameter>
     </se:Fill>
     <se:Stroke>
       <se:SvgParameter name="stroke">#000000</se:SvgParameter>
       <se:SvgParameter name="stroke-width">0.26</se:SvgParameter>
       <se:SvgParameter name="stroke-linejoin">bevel</se:SvgParameter>
     </se:Stroke>
   </se:PolygonSymbolizer>
   <PointSymbolizer>
  <Graphic>
    <Mark>
      <WellKnownName>dot</WellKnownName>
      <Fill>
        <CssParameter name="fill">#0033CC</CssParameter>
      </Fill>
    </Mark>
    <Size>26</Size>
  </Graphic>
</PointSymbolizer>
 </se:Rule>
 <se:Rule>
   <se:Name> 6718272857.1429 - 8167348571.4286 </se:Name>
   <se:Description>
     <se:Title> 6718272857.1429 - 8167348571.4286 </se:Title>
   </se:Description>
   <ogc:Filter xmlns:ogc="http://www.opengis.net/ogc">
     <ogc:And>
       <ogc:PropertyIsGreaterThan>
         <ogc:PropertyName>revenue</ogc:PropertyName>
         <ogc:Literal>6.71827e+09</ogc:Literal>
       </ogc:PropertyIsGreaterThan>
       <ogc:PropertyIsLessThanOrEqualTo>
         <ogc:PropertyName>revenue</ogc:PropertyName>
         <ogc:Literal>8.16735e+09</ogc:Literal>
       </ogc:PropertyIsLessThanOrEqualTo>
     </ogc:And>
   </ogc:Filter>
   <se:PolygonSymbolizer>
     <se:Fill>
       <se:SvgParameter name="fill">#f45629</se:SvgParameter>
     </se:Fill>
     <se:Stroke>
       <se:SvgParameter name="stroke">#000000</se:SvgParameter>
       <se:SvgParameter name="stroke-width">0.26</se:SvgParameter>
       <se:SvgParameter name="stroke-linejoin">bevel</se:SvgParameter>
     </se:Stroke>
   </se:PolygonSymbolizer>
   <PointSymbolizer>
  <Graphic>
```

```
    <Mark>
      <WellKnownName>dot</WellKnownName>
      <Fill>
        <CssParameter name="fill">#0033CC</CssParameter>
      </Fill>
    </Mark>
    <Size>32</Size>
  </Graphic>
</PointSymbolizer>
 </se:Rule>
 <se:Rule>
   <se:Name> 8167348571.4286 - 9616424285.7143 </se:Name>
   <se:Description>
     <se:Title> 8167348571.4286 - 9616424285.7143 </se:Title>
   </se:Description>
   <ogc:Filter xmlns:ogc="http://www.opengis.net/ogc">
     <ogc:And>
       <ogc:PropertyIsGreaterThan>
         <ogc:PropertyName>revenue</ogc:PropertyName>
         <ogc:Literal>8.16735e+09</ogc:Literal>
       </ogc:PropertyIsGreaterThan>
       <ogc:PropertyIsLessThanOrEqualTo>
         <ogc:PropertyName>revenue</ogc:PropertyName>
         <ogc:Literal>9.61642e+09</ogc:Literal>
       </ogc:PropertyIsLessThanOrEqualTo>
     </ogc:And>
   </ogc:Filter>
   <se:PolygonSymbolizer>
     <se:Fill>
       <se:SvgParameter name="fill">#df2722</se:SvgParameter>
     </se:Fill>
     <se:Stroke>
       <se:SvgParameter name="stroke">#000000</se:SvgParameter>
       <se:SvgParameter name="stroke-width">0.26</se:SvgParameter>
       <se:SvgParameter name="stroke-linejoin">bevel</se:SvgParameter>
     </se:Stroke>
   </se:PolygonSymbolizer>
 </se:Rule>
 <se:Rule>
   <se:Name> 9616424285.7143 - 11065500000.0000 </se:Name>
   <se:Description>
     <se:Title> 9616424285.7143 - 11065500000.0000 </se:Title>
   </se:Description>
   <ogc:Filter xmlns:ogc="http://www.opengis.net/ogc">
     <ogc:And>
       <ogc:PropertyIsGreaterThan>
         <ogc:PropertyName>revenue</ogc:PropertyName>
         <ogc:Literal>9.61642e+09</ogc:Literal>
       </ogc:PropertyIsGreaterThan>
       <ogc:PropertyIsLessThanOrEqualTo>
         <ogc:PropertyName>revenue</ogc:PropertyName>
         <ogc:Literal>1.10655e+10</ogc:Literal>
       </ogc:PropertyIsLessThanOrEqualTo>
     </ogc:And>
   </ogc:Filter>
   <se:PolygonSymbolizer>
```

```
            <se:Fill>
              <se:SvgParameter name="fill">#bd0026</se:SvgParameter>
            </se:Fill>
            <se:Stroke>
              <se:SvgParameter name="stroke">#000000</se:SvgParameter>
              <se:SvgParameter name="stroke-width">0.26</se:SvgParameter>
              <se:SvgParameter name="stroke-linejoin">bevel</se:SvgParameter>
            </se:Stroke>
          </se:PolygonSymbolizer>
          <PointSymbolizer>
         <Graphic>
           <Mark>
             <WellKnownName>dot</WellKnownName>
             <Fill>
               <CssParameter name="fill">#0033CC</CssParameter>
             </Fill>
           </Mark>
           <Size>38</Size>
         </Graphic>
       </PointSymbolizer>
        </se:Rule>
        <Rule>
        <TextSymbolizer>
            <Label>
              <ogc:PropertyName>county</ogc:PropertyName>
            </Label>
            <Font>
              <CssParameter name="font-family">Times New Roman</CssParameter>
              <CssParameter name="font-style">Normal</CssParameter>
              <CssParameter name="font-size">14</CssParameter>
            </Font>
            <LabelPlacement>
              <PointPlacement>
                <AnchorPoint>
                  <AnchorPointX>0.5</AnchorPointX>
                  <AnchorPointY>0.5</AnchorPointY>
                </AnchorPoint>
              </PointPlacement>
            </LabelPlacement>
          </TextSymbolizer>
        </Rule>
      </se:FeatureTypeStyle>
    </UserStyle>
  </NamedLayer>
</StyledLayerDescriptor>
```

4.6 Using charts for data visualization

This application is online so JavaScript programming has been used to create the charts in figures 4.10, 4.11, 4.12, 4.13 and 4.14 below. All that is required is to copy the code, paste on a text editor like notepad then save the file with html extension to make it a web page. Open the file using your mobile browser.

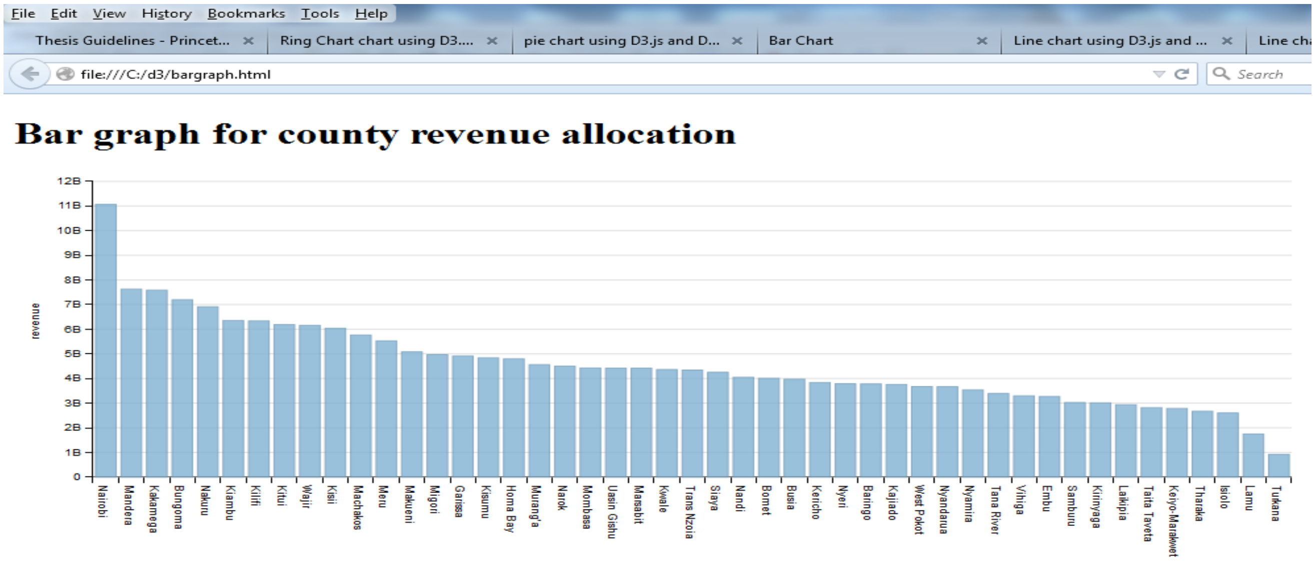
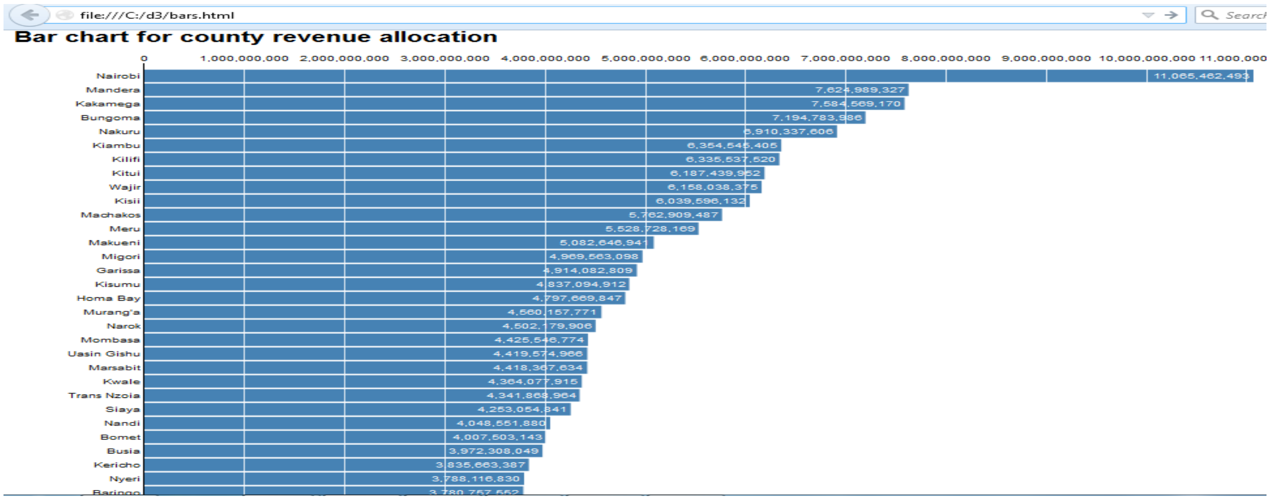


Figure 4.10: Bar chart for county revenue allocations 2014/1015

```
<!DOCTYPE html>
  <html>
    <head lang="en">
      <meta charset="UTF-8">
        <title>Bar chart using D3.js and Dimple.js</title>
      </head>
      <body>
                               <h1> Bar graph for county revenue allocation</h1>
        <div id="chartContainer"></div>
      </body>
      <script src="d3.v3.min.js"></script>
      <script src="dimple.v2.1.2.min.js"></script>

  <div id="chartContainer">
  <script type="text/javascript">
  var svg = dimple.newSvg("#chartContainer", 990, 400);
  d3.csv("countyrevenue.csv", function (data) {
    var myChart = new dimple.chart(svg, data);
    myChart.setBounds(60, 10, 910, 305)
    var x = myChart.addCategoryAxis("x", "county");
    x.addOrderRule("Date");
    myChart.addMeasureAxis("y", "revenue");
   var s = myChart.addSeries(null, dimple.plot.bar);
          s.barGap = 0.2;
    myChart.draw();
  });
 </script>
</div>
  </script>
   </html>
```


file:///C:/d3/bars.html

Bar chart for county revenue allocation

Figure 4.11: Bar chart for county revenue allocations 2014/2015

```
<!DOCTYPE html>
<html>
  <head>
    <title>Bar Chart</title>
    <script src="d3.v3.min.js"></script>
        <style type="text/css">
body {
  font: 10px sans-serif;
}
.bar rect {
  fill: steelblue;
}
.bar text.value {
  fill: white;
}
.axis {
  shape-rendering: crispEdges;
}
.axis path {
  fill: none;
}
.x.axis line {
  stroke: #fff;
  stroke-opacity: .8;
}
.y.axis path {
  stroke: black;
}
    </style>
  </head>
  <body>
  <h1> pie chart for county revenue allocation</h1>
    <script type="text/javascript">
var m = [10, 10, 10, 100],
    w = 960 - m[1] - m[3],
    h = 930 - m[0] - m[2];
```

```
var format = d3.format(",.0f");
var x = d3.scale.linear().range([0, w]),
    y = d3.scale.ordinal().rangeRoundBands([0, h], .1);
var xAxis = d3.svg.axis().scale(x).orient("top").tickSize(-h),
    yAxis = d3.svg.axis().scale(y).orient("left").tickSize(0);
var svg = d3.select("body").append("svg")
    .attr("width", w + m[1] + m[3])
    .attr("height", h + m[0] + m[2])
  .append("g")
    .attr("transform", "translate(" + m[3] + "," + m[0] + ")");
d3.csv("sample-data.csv", function(data) {
  // Parse numbers, and sort by value.
  data.forEach(function(d) { d.value = +d.value; });
  data.sort(function(a, b) { return b.value - a.value; });
  // Set the scale domain.
  x.domain([0, d3.max(data, function(d) { return d.value; })]);
  y.domain(data.map(function(d) { return d.name; }));
  var bar = svg.selectAll("g.bar")
      .data(data)
    .enter().append("g")
      .attr("class", "bar")
      .attr("transform", function(d) { return "translate(0," + y(d.name) + ")"; });
  bar.append("rect")
      .attr("width", function(d) { return x(d.value); })
      .attr("height", y.rangeBand());
  bar.append("text")
      .attr("class", "value")
      .attr("x", function(d) { return x(d.value); })
      .attr("y", y.rangeBand() / 2)
      .attr("dx", -3)
      .attr("dy", ".35em")
      .attr("text-anchor", "end")
      .text(function(d) { return format(d.value); });
  svg.append("g")
      .attr("class", "x axis")
      .call(xAxis);
  svg.append("g")
      .attr("class", "y axis")
      .call(yAxis);
});
    </script>
  </body>
</html>
```

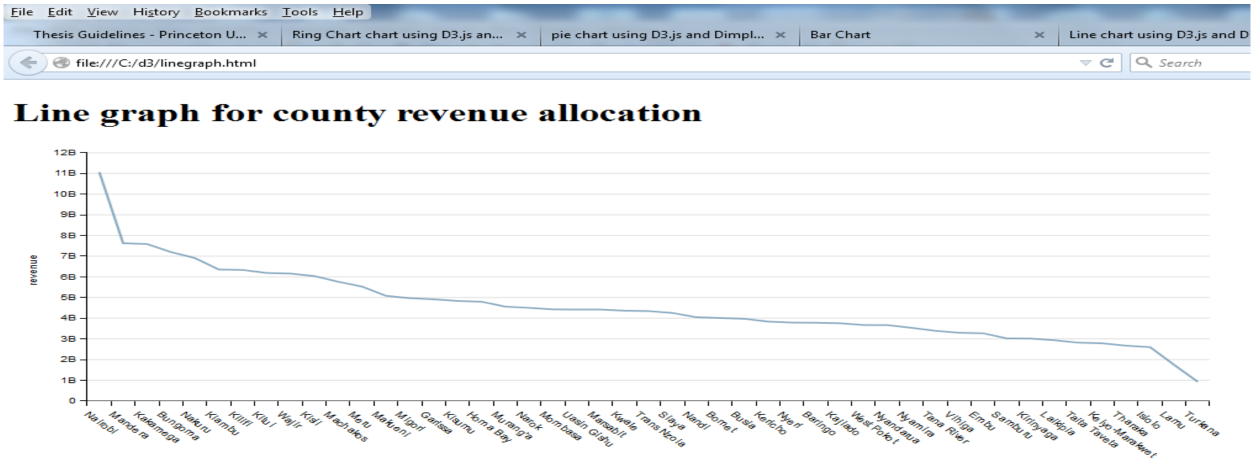


Figure 4.12: Line graph for county revenue allocations 2014/2015

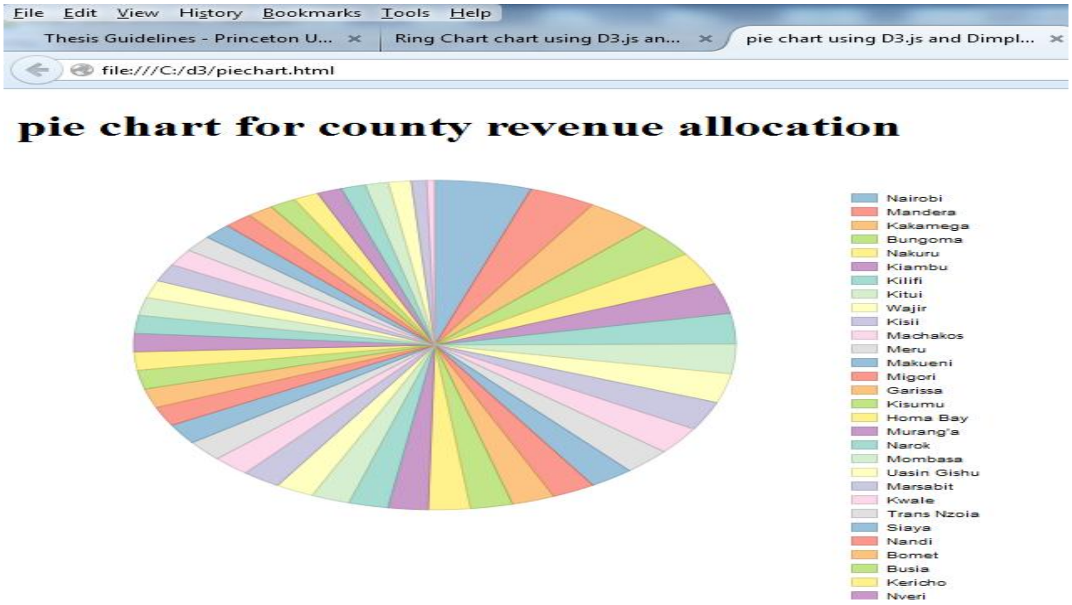


Figure 4.13: Pie chart for county revenue allocations 2014/2015

```
<!DOCTYPE html>
   <html>
      <head lang="en">
         <meta charset="UTF-8">
            <title>pie chart using D3.js and Dimple.js</title>
         </head>
         <body>
                              <h1> pie chart for county revenue allocation</h1>
            <div id="chartContainer"></div>
         </body>
         <script src="d3.v3.min.js"></script>
         <script src="dimple.v2.1.2.min.js"></script>
```

```
 <div id="chartContainer">
  <script type="text/javascript">
  var svg = dimple.newSvg("#chartContainer", 590, 800);
  d3.csv("countyrevenue.csv", function (data) {
    var myChart = new dimple.chart(svg, data);
    myChart.setBounds(20, 20, 460, 360)
    myChart.addMeasureAxis("p", "revenue");
    myChart.addSeries("county", dimple.plot.pie);
    myChart.addLegend(500, 20, 90, 800, "left");
    myChart.draw();
  });
 </script>
</div>
  </script>
   </html>
```

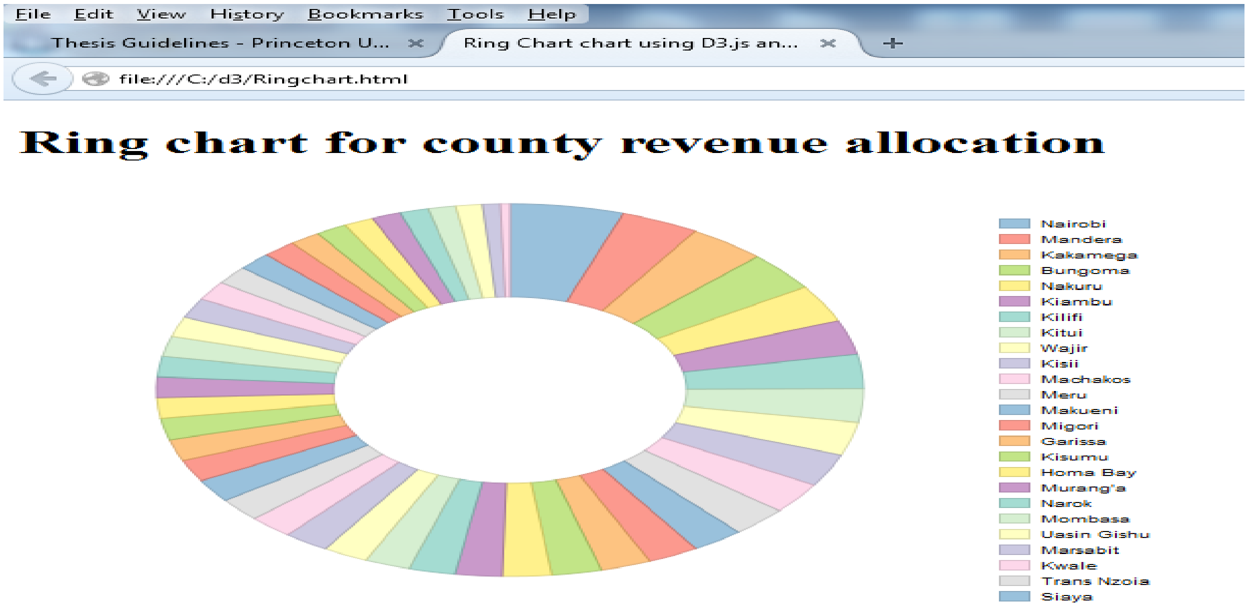


Figure 4.14: Ring chart for county revenue allocations 2014/2015

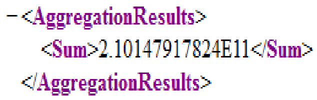
4.7 Web processing service xml file for data aggregation

GeoServer has the capability to data aggregation and analysis. All that is required is to open the WPS menu item and specify the data layer and the process. An XML file with the results is generated.

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
```

```
<ows:Identifier>gs:Aggregate</ows:Identifier>
<wps:DataInputs>
<wps:Input>
        <ows:Identifier>features</ows:Identifier>
        <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
                  <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
                  xmlns:munyao="www.munyao.co.ke">
                  <wfs:Query typeName="munyao:county"/>
                  </wfs:GetFeature>
        </wps:Body>
        </wps:Reference>
</wps:Input>
<wps:Input>
        <ows:Identifier>aggregationAttribute</ows:Identifier>
        <wps:Data>
        <wps:LiteralData>revenue</wps:LiteralData>
        </wps:Data>
</wps:Input>
<wps:Input>
        <ows:Identifier>function</ows:Identifier>
        <wps:Data>
        <wps:LiteralData>Sum</wps:LiteralData>
        </wps:Data>
</wps:Input>
<wps:Input>
        <ows:Identifier>singlePass</ows:Identifier>
        <wps:Data>
        <wps:LiteralData>true</wps:LiteralData>
        </wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput mimeType="text/xml">
        <ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```

4.8 Results of Aggregation



```
- <AggregationResults>
    <Sum>2.10147917824E11</Sum>
  </AggregationResults>
```

210 billion is the sum total that counties received from the central government. Other web processing service aggregation functions include Average, standard deviation, maximum, minimum and medium.

4.9 Discussions

The aim is to show county revenue allocation in Kenya using thematic maps and charts. The revenue data used is for financial year 2014/2015. A thematic map using a red color hue has been generated as shown in figures 4.4 and 4.5. A dot map overlaid on a thematic map has also been generated as shown in figure 4.8 and 4.9. Charts have also been generated using JavaScript programming language. Figure 4.10 and 4.11 show bar graphs with landscape and portrait orientations simultaneously. Pie and ring charts have also been generated as shown in figure 4.13 and 4.14. A picture speaks a thousand words, from the figures one can be able to tell at a glance the county that received the largest and the smallest allocations together with those that received almost the same amounts. The results shown here are the same as those gotten using Arcgis and excel. However in this case the student is using a mobile device. The software templates provided are enough for this task. All the user needs to do is modification of the code provided and not reinventing the wheel. The system is web-based hence the task can be done anywhere. The student can discuss the task with their lecturer along the stairs or a car and a change in say color scheme done on fly with instant feedback being provided. The system is hosted from a central server hence the lecturer can track assignment progress and provide the necessary feedback. With Arcgis and excel the assignment would only exist on students laptop and therefore appointments will be required and with large class numbers you can only get a few of these if any.

CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS

In this chapter, a summary status of the constructed system and possible future work on the other components of the system is presented.

5.1 Conclusions

The main objective of this study is to provide a Web GIS-based education software solution that can be used across mobile devices. It consists of a software stack with PostGIS[24] as a data store, GeoServer [25] as a data accessor and a client implemented in JavaScript with HTML5/CSS3[26]. The client utilizes the OpenLayers [27] JavaScript library, as well as other JavaScript utility libraries. The application is compliant with current standards for storing and presenting and communicating geographic data, as well as current standards in web development. The most central geospatial standards employed are WMS[28], WFS[29]and WPS[30].

Once installed in the server users are able to access it online through the internet. No software installation is required and data visualization assignments can be carried from mobile devices. Discussions on assignment progress are not restricted to the computer lab and prevent assignments from looking like time constrained exams. This meets the first specific objective. The system provides a perceptive way of discussing the progress of your assignment with the instructor as opposed to verbal discussion encouraging visible learning. This meets my second specific objective. FOSS tools have been used to develop the system which lowered the financial barriers to entry. It would have been very expensive to use proprietary software for development. Thus it has been demonstrate that open source tools have come of age and can be used in developing Web GIS based teaching software.

[24] http://opengeo.org/technology/postgis/ (last accessed 23rd April 2015)

[25] http://docs.geoserver.org/2.6.x/en/user/index.html (last accessed 3rd May 2015)

[26] http://w3cschools.org (last accessed 3rd May 2015)

[27] http://openlayers.org/ (last accessed 3rd May 2015)

[28] http://docs.geoserver.org/2.6.x/en/user/index.html (last accessed 3rd May 2015)

[29] http://docs.geoserver.org/2.6.x/en/user/index.html (last accessed 3rd May 2015)

[30] http://docs.geoserver.org/2.6.x/en/user/index.html (last accessed 3rd May 2015)

5.2 Recommendations

GIS programming course should be taught in the first semester. Classification of remote sensing data takes long processing time even days. Browsers have been designed to time out if they keep making requests for 15 minutes without response. Therefore this functionality could not be implemented. A possible way forward would be providing the user with a web page that specifies the dataset, type of classification to be performed and their email. Processing is carried out off line and the results emailed back to the user. We are also witnessing advances in both hardware and software capable of performing massive parallel processing. This in future will ensure faster processing of remote sensing data.

5.3 Future work

Having implemented the challenging part i.e. carrying out GIS laboratory work online, developing the remaining components for a GIS online learning portal for all the other GIS theory units. This will make it possible to have a GIS online institution. Most institutions have embraced PowerPoint presentation slides as a model of content delivery. Web-based seminar (Webinar) provide an online opportunity of doing the same albeit more efficiently. Webinars provide an opportunity not only to present slides online but also to interact with students and respond to students questions i.e. both audio and visual. More important, attendance can be monitored based on students who login for class. The opportunities here are immense. First, the instructor can conduct their class from any part of the world. Second, students can attend the class from any part of the world. Thus traffic jams, cost of transport and location are no longer barriers to learning. Institutions have the opportunity to market GIS courses to students all over the world. Email software will also be important for students to receive lecture notes, important announcements and send back written assignments. This should be complemented with instant messaging software like Facebook. Lastly services have to be paid for and a software component is required to achieve this task. Because of security requirements involved in financial transactions the best model would be link up with online organizations that have been known to do this safely and securely like PayPal. Students are able to pay their school fees online with credit cards while PayPal ensures secure transactions. In this age of eCommerce, ePayments, eBanking this system should make the dream of Open eDistance learning come true. Apart from "electronic this electronic that" we are now in the age of "mobile this mobile that" i.e. mPesa, mBanking this system will fill in the mDistance learning gap when fully complete.

REFERENCES

Andone, D. D. and Pemberton, (2009). Developing a desirable learning environment for digital students. Technology, Instruction, Cognition and Learning, 6(4), 253-271.

Baker, T.R. (2005). Internet-Based GIS Mapping in Support of K-12 Education.

Caverly, D. C., Ward, A. R. & Caverly, M. J. (2009). Techtalk: Mobile learning and access. Journal of Developmental Education, 33(1), 38-39.

Commack (1999). Cartography and Geographic Information Systems.

Kerski, J. J. (2003). The implementation and effectiveness of GIS in secondary schools.

Mahmoud, S. S. (2008). A proposed model for distributing e-courses content through mobile technology architectures

Sui, D.Z. (1995). A Pedagogic Framework to Link GIS to the Intellectual Core of Geography.