An Android mobile GIS application for facilitating field work in electric utility

Xiaoguo Wan

October, 2014

Degree project thesis, Bachelor, 15 hp
Geomatics

Study programme in Geomatics

Supervisor: Fredrik Ekberg
Examiner: Nancy Joy Lim
Co-Examiner: Markku Pyykönen
Abstract

Geographic Information System (GIS) is a technology that can help collect, edit, store, manage, transform, analyze and represent data. This research aimed to show how mobile GIS application can be designed, developed and applied to electric field work. An Android mobile application, which had been integrated with GIS was developed in Java. The prototype was tested and evaluated by employees of China Southern Power Grid (CSG).

The resulting mobile application had three important features and functionalities: map view, location and electrical object querying, and attribute viewing and editing. According to the results of the user testing, the advantages of the application were its improved performance as brought about by the different scales used in the maps, its clear layout, and the speed of completing the query tasks. The disadvantages, on the other hand, were restrictions in the data types used and the query window.

The study showed that with wireless communication, GPS and Location-based (LBS) supplements, electric utilities can benefit from this mobile GIS application in the following aspects: 1) eliminate the obstacles of going out in the field; 2) increase the flexibility of field work by faster access to data; 3) increase the accuracy and efficiency of field workers by dealing with the querying and editing task, and field mapping on a portable devices in real time. According to these aspects, electric field work can easily be enhanced by saving time, resources, and costs.

**Keywords:** Mobile GIS, electric utilities, field work, Android, mobile application.
Acknowledgements

I would like to thank all the people who gave me help in this study.

First of all, I would like to extend my sincere gratitude to my supervisor, Fredrik Ekberg, who has given me constructive suggestions for this research, and gives me courage;

I feel really grateful to my examiner, Nancy Joy Lim, for helping me with the research proposal, and giving me very useful suggestions for the thesis writing;

In addition, I deeply appreciate the help of China Southern Power Grid Corporation Limited (CSG), for giving me the chance to do a field study, and showing me how a spatial map can work for electrical system management;

My sincere thanks are also given to Mr. Wang, an employee at CSG, for helping me test the application and debug the technical problems;

Besides, my gratitude also extends to Richard Hejdenberg, an employee at Gävle Energy, for giving me interviews;

Lastly, my thanks would go to my parents, my grandpa, and my friends, for their encouragement and support during this research.

October 2014
List of Figures

Figure 2.1: The different architectures of mobile GIS ............................................................. 11
Figure 3.1: Development environment for an Android application ........................................ 13
Figure 3.2: Overview of the test map ....................................................................................... 14
Figure 3.3: Maps in different scales: A. 1:40,000; B. 1: 15,000; C. 1: 6,000. ......................... 15
Figure 3.4: System Configuration of this application .............................................................. 16
Figure 3.5: Interface of *Attribute View/Edit* task ................................................................. 17
Figure 4.1: Icon and the main interface of the application..................................................... 19
Figure 4.2: Screen views in increasing map scales: a) layers of *urban, green, water* and *city* are visible; b) road and electrical networks consisting of *segment lines, abandoned segment* and *gadget points* are shown; and, c) at larger scale, the electrical codes and road names are displayed. .................................................................................................................................. 20
Figure 4.3: Items under the button *Location and Query Task.* ............................................. 20
Figure 4.4: Using the different query tasks for :a) *Electrical Line*; b) *Electrical Meter*; and, c) *Location* ............................................................................................................................................. 21
Figure 4.5: a) Example editing in the *Attribute/Edit* dialog; b) sample results after the attributes are edited. ............................................................................................................................................. 22
1. Introduction

1.1 Background

In this modern era, communication and information technology develops rapidly, and mobile devices are always upgrading. Smartphone has replaced most portable devices (e.g. Camera, MP3/MP4, and PDA) and combined them into one. At the same time, the speed and coverage of broadband internet have been improved. Therefore, using the internet on a smartphone for work or entertainment has become more convenient and efficient. According to Weng, Sun and Grigsby (2012), smartphone can support instant data manipulation, storage, transmission and direct uploading on a 3G/4G network or in an available Wi-Fi environment. Hence, taking advantage of such capabilities has been a focused area in the application development industry.

Geographical Information System (GIS) is an information technology system consisted of hardware, software, database, and spatial data. According to Jack Dangermond, (as cited in Goodchild, 1991, p. 194), “GIS is clearly not a new science, but rather a technology that requires a considerable scientific knowledge base for many of its data management functions.” Complete GIS can help manage various data formats that contain graphics, layers, symbols, and attributes. Moreover, with the growth of the Global Positioning System (GPS), Location-based Service (LBS), and wireless communication, GIS has developed from desktop software to mobile application.

Mobile GIS emerged to facilitate field work such as field surveying and utility maintenance in the middle of 1990s. With the supplement of widely covered wireless communications, mobile GIS has become a part of web GIS (Fu & Sun, 2010). At present, it can help access, use and store the latest data directly in the field work process (Poorazizi, Alesheikh, & Behzadi, 2008). In addition, with the integration of LBS and GPS, a map-based mobile GIS application can make it easier to find data locations, view objects nearby, and get optimal routes (Stoimenov, Stanimirović, Bogdanović, Davidović, & Krstić, 2010). Applying mobile GIS to data management has become more significant for utilities, especially in field work.

For electric utility, field work is needed day–by-day for ensuring that the electrical device is functioning properly. However, these devices are hidden under the ground or in protected areas, so they cannot be found easily. In the past decades, field workers used paper-based map and traditional data-recording devices in their inspections. Known disadvantages of these methods are: 1) they are time and labor consuming; 2) limited in instantaneous positioning; 3) difficult to query. The problems mentioned can lead to data loss, repeated work, lower efficiency and rising costs. Therefore, using electronic maps with GPS is significant and necessary. Mobile GIS equipped with the application programmed to compile map data, that
is portable to carry and with GPS functionality can assist field workers in managing their tasks efficiently.

1.2 Aim of the Research and Research Questions

The objective of this research was to study how mobile GIS application can be designed and developed to facilitate electric field work. This included defining the platform to be used and choosing the different GIS functionalities appropriate for the application. To achieve this objective, a test application was created using the data sample provided by China Southern Power Grid Corporation Limited (CSG). This application was also evaluated for its functionalities and to see whether it encounters actual problems in processing data from field work of electrical utility.

The following research questions were used as guides to do the research:

1. How can mobile-based GIS application be used for electric field work?

2. What problems have to be mainly considered in developing real-time mobile-based applications, especially for electric field work?

3. How can electric utilities benefit from real-time, mobile-based GIS applications?
2. Literature Review

In this part, previous researches concerning electric field work and mobile GIS are discussed to give an insight into the background of this study.

2.1 GIS-based Electric Management

Electric utilities control the supply of electric power. In China, they are responsible for managing power transmission, transformation, distribution, inspection, maintenance, and even customer service. In order to manage these processes, Supervisory Control and Data Acquisition (SCADA), Distribution Management System (DMS), and GIS are commonly used. SCADA and DMS are basic methods applied in real-time management of electric utilities, while GIS serves as a platform for data asset management. Shin and Feuerborn (2004) have pointed out that the importance of the electric systems is to stimulate the real distribution phenomenon with computer technology.

Compared with other electric systems, GIS have several extra functions in enhancing data performance. It can describe the spatial data on a map, such as the location and shape of geometric objects, and help analyze spatial relationships. It can also store attribute data of map features, and provide the capability to view the distribution network superimposed on different layers (Shin & Feuerborn, 2004). Lastly, GIS can support not only the analysis of electric system by using the graphical display, but also updating the data (Wei, Yun-Cheng, Tongyu, & Ying-li, 2010).

With those kinds of capabilities, GIS has attracted the attention of electric utility managers. Among electric management systems, there are two representative cases that have been widely studied. The first one is PowerView, which was developed by Burns & McDonnell Inc. Shin and Feuerborn (2004), who have conducted a research on it, claimed that more intuitive information of electric system could be obtained by analyzing and viewing the geographic layers on a map, together with the utilization of an electric database. The other case is GinisED, which was developed by PD Jugoistok Niš, Laboratory for Computer Graphics and Geo-Information systems, and the Faculty of Electronic Engineering in Niš, which is supported by the Ministry of Science in the Republic of Serbia (Stoimenov et al., 2010). This project is mainly used for collecting, editing, visualizing and analyzing spatial data based on GIS technologies. According to Stoimenov et al. (2010), Web GIS application can be helpful for electric utilities management.
2.2 Field Work

With the analysis capability and visualization features of GIS software, electric utilities can be managed efficiently and accurately. However, electric management does not only include desktop computer processing, but also include frequent outdoor maintenance and inspection or field work. Certain tasks that have to be completed outside the office are field surveys, power line maintenance, and emergency response, which are needed for regular inspections of the status of the electric equipment. When doing field work, two aspects should be considered. One is visualization (Pundt & Brinkkotter-Runde, 2000), and the other is data acquisition (Poorazizi et al., 2008).

Previously, field workers needed to hold the maps in their hands to accomplish asset inventory and inspection, to find the locations of the equipment, to get the overview of an incident area, and to drive in the optimal directions (Fu & Sun, 2010). However, for high-frequency tasks such as in electric management, it can be a waste of resource and time. Therefore, Field-based GIS has become the new solution for data visualization and acquisition.

From the visualization perspective, the actual power distribution can be well displayed on a GIS-based map. According to Awange and Kiema (2013a), GIS technology can handle geographical and attribute data. With the visualization servers, users can get a better understanding of real world conditions (Beard, Buttonfield, & Mackaness, 1994). Pundt and Brinkkotter-Runde (2000) also state that users can view the attributes of interests on the digital graphical display. Accordingly, GIS visualization supplement has contributed to enhancing the field work experience.

For data acquisition, accessing data can facilitate field work. Poorazizi et al. (2008) have listed different field processes that can benefit from this such as asset inventory recording, asset management, inspections maintenance, and incident reporting. They also point out that, with GIS technique, all of the data can be recorded with both location and attribute on a digital map.

2.3 Mobile GIS

Online data access for field work needs handheld devices such as laptops, smartphones, and Personal Digital Assistants (PDAs). Smartphone can be an optimal choice, because devices such as PDAs may have limited users. Additionally, although a laptop can function better, it would make tasks time-consuming and challenging due to its larger size. Hence, Mobile GIS using smartphones has been widely adopted by utilities and enterprises in field work.
2.3.1 Android Smartphone

Since the development of mobile industry in recent years, smartphones have been widely used. A modern mobile phone is not only used to make phone calls and send messages, but also contains the functions of a camera, MP3/MP4, PDA and GPS. According to the study of Weng, Sun and Grigsby (2012), a traditional Smartphone has about 8 to 32 gigabytes of memory, which is still increasing. In addition, it can connect to the internet through Wi-Fi or cellular networks, which allows users to send and receive various files of different types, such as texts, pictures, and videos via the communication network.

Like personal computers, various operating systems (OS), such as iOS, Android, Symbian, Windows and Blackberry can be chosen for Smartphone. Each OS has its own graphical user interface, thumb wheels, keyboard, and wireless communication capabilities. For application developers, the challenge of developing a mobile GIS application is choosing the optimal OS that will cater the needs of most its target users (Fu & Sun, 2010). Nowadays, the most popular mobile phone OS is Google-backed Android.

Android has been acquired by Google since 2005 and released to be an open source by Google and Open Handset Alliance (OHA) on November 12, 2007. It provides free Software Development Kit (SDK) for application developers, and the Android Market can also publish the applications (Kristian, Armanto & Frans, 2012). Considering the cost and convenience, majority of the Smartphone companies prefer Google-backed Android system as the operating system of their products. The growing market demand motivates more software developers to develop Android applications that run on Android devices.

2.3.2 Wireless communication, GPS & LBS Supplement

For data gathering and updates via mobile GIS, the availability of wireless communication is important as most of the applications are accessed online. Different wireless communication technologies have different speed, range and setup costs. Typical technologies are Bluetooth, Wi-Fi, and cellular networks, which Android devices can support. However, in order to work in the field, cellular network must be available. 3G is the most commonly used cellular network nowadays. But with 4G, it will have an approximated speed of 100 megabits per second (mbps) that can facilitate mobile GIS functionalities (Fu & Sun, 2010).

GPS function is of equal importance as wireless communication for mobile GIS application. It is the oldest and most widely used GNSS system, providing passive real-time 2D or 3D positioning, navigation and velocity data. Space, control and user segments are principal components that comprise a complete GPS (Awange & Kiema, 2013a). It can monitor the information on speed, bearing, track, trip distance, and time. Besides, GPS can work in different weather conditions, 24 hours a day without paying for the service (Gu, Lu, Guo &
However, GPS measurements have significant inevitable errors as mentioned by Awange and Kiema (2013a), such as: 1) ephemeris errors; 2) clock errors; 3) atmospheric errors; 4) multipath; 5) satellite constellation “geometry”. These usually lead to imprecise GPS tracking data on the map, but the accuracy mainly depends on the climatic and environmental conditions.

Mapping applications provide location-based service in mobile GIS. According to Virrantaus et al. (2001), LBS is an information service that can be used by mobile devices to acquire real-time positions. LBS can also be defined to integrate location, position and other information on map service. Major technologies used for LBS include hardware, software, programming tools, wireless communication, GPS and web services (Tsou, 2004). It benefits users in navigation, querying of points-of-interest (POIs) and locating people. For enterprises and utilities, LBS can provide more business opportunities of finding particular person or object, tracking resources, and proximity-based notification (Fu & Sun, 2010).

Location-based information has been common on the web, with nearly 20% of web-based search engines focusing on places or regions (Liu, Rau, & Gao, 2010). When users carry out activities in one place, the location-based information can be captured by LBS in an instant. Besides the immediacy, location information is also quite precise. In the study of Watters and Amoudi (2003), 80% of the location information on the website is accurate and accessible for query operations. People can search for nearby services to find the way to their destination. Shu, Du and Chen (2009) stated that the future of location-based service will be on the Android platform. For power grid, traffic or other management fields, LBS can act as an emergency support that can help speed up the response time, which relies on accurate positions both from the location-based database and GPS.

2.4 Key Characteristics of Application Development

With the availability of wireless communications, GPS, and LBS, mobile GIS applications for field work can enjoy faster speed of information transmission at lower cost. As a result, the development of an Android-based GIS application has become more attractive. There are several key characteristics in application development. These are: Programming language, system architectures, SDK, Application Programming Interface (API), Integrated Development Environment (IDE), and database.

In Android, the programming language used is Java, which has been tested, extended, and refined for a longer period of time by developers (Java, n. d.). Android is based on Linux platform, the host middleware, which is programmed with Java to run the applications. In the research of Blasing, Batyuk, Schmidt, Camtepe and Albayrak (2010), they pointed out that Java can help analyze Android application applicable as cloud service. Martinez-Llario and Gonzalez-Alcaide (2011) have compared Java with C project, and conclude that Java applications have more advantages in spatial extensions for GIS Relational Database Management Systems (RDBMSs).
Similar to web GIS, which can provide GIS functionalities on the web through Hypertext Transfer Protocol (HTTP) and Hypertext Markup Language (HTML), mobile GIS has system architectures (Shanzhen, Lizhu, Chunxiao, Qilun, & Yong, 2001), such as: Stand-alone; Client-server; Distributed; Services; and, Peer-to-peer (Hassin, 2003). In the study of Poorazizi et al. (2008), the authors briefly described these five architectures, as seen in Figure 2.1. The simplest is the Stand-alone client, in which the Smartphone application and database are both entirely residing in the device. Client-Server (C/S) architecture is the most adopted one, wherein the client-side in mobile hardware displays the maps and the demands of users analyzed by using GIS, while the server-side components offer the geospatial data and follow the instructions from the client-side (Tsou, 2004). As an extension to C/S, distributed client-server architecture has the ability for data resource management to overcome the internet connectivity problems of C/S. The Services architecture further extends the back-end functionality of mobile GIS, making mobile applications able to connect to the GIS server to provide web services. It supports communication between many mobile phones and web services. Peer-to-peer architecture requires several mobile devices, which can be on the client-side or server-side. If a mobile device stores the database, it can work as the Stand-alone Client.

GIS application requires specific SDK and API. ESRI supplies ArcGIS Runtime SDK for Android and ArcGIS for Server (ESRI, 2013a), which can be a server site for web or mobile application, that allows access to GIS functionalities on any client-side devices, such as editing web information, using network analysis, and modeling. These server sites consist of
three main access points: ArcGIS Server Manager, the Services Directory and the ArcGIS Server Administrator Directory. ArcGIS Server Manager is a web browser-based application, which allows users to manage the web services site properties (Law, 2013). It is integrated with the desktop applications, the web and the mobile devices by giving access to web mapping APIs and mobile runtime SDKs.

IDE is used for assisting programmers to develop software. With the help of IDE, programmers are able to program and test their applications. For Android application development, it is possible to use an ordinary IDE such as Eclipse. Based on most of researches, Eclipse is preferred for Android application development (Vidas, Zhang & Christin, 2011).

The last characteristic is database, which is used for storing data. GIS database is where data, such as statistical data, maps, as well as remotely sensed images are stored (Poorazizi et al., 2008). Awange and Kiema (2013b) regard Database Management System (DBMS) as the most modern data operation approach.
3. Materials & Method

This part illustrates how the application has been designed and developed. Data processing of the different map layers and how the user testing is conducted are also explained in this chapter.

3.1 Development Environment

Hall, Gordon, Newall, and James (2004) have mentioned that different operating systems have different IDEs. An open platform can support more application developers to design new works. For example, Google provides an open source to develop Android applications, and ESRI provides majority of free software development kits for Android application development. To date, the most popular method to develop mobile GIS application is to combine these two.

The application was programmed in Eclipse using Java as the programming language. ArcGIS for Server was used on the server side, to allow access to the GIS map services through the web. It was used to create and edit the map and spatial data. Oracle 11g was the database where the data that needed publishing on the web service by the server were stored. ArcSDE was the database engine that transformed the spatial data from ArcGIS to Oracle. Figure 3.1 shows the development environment for this work.

![Development Environment Diagram](image)

**Figure 3.1: Development environment for an Android application**

3.2 Map and Data Processing

The test data used for this project was created following the model of the Guiyang map, which is provided by China Southern Power Grid Corporation Limited (CSG). The spatial data were stored in tables following the relational model. These tables were imported to ArcCatalog and saved in a geodatabase. Each electric feature was given an ID code with nine numbers that
were used as primary keys. Recoding each element, which was used for the application, was necessary to keep the privacy of customer information.

The test map model was produced by CSG, and it contained several feature layers. Urban and forest were chosen to be the background layers. Roads, gadget point, segment lines and abandoned segments were used as operational layers that served as the APIs for querying, editing, and updating the application. The overview of the test map is shown in Figure 3.2. Users can apparently identify the symbols of City and gadget point according to respective colors and names on the enlarged map, which might be less identifiable in the preview map.

![Figure 3.2: Overview of the test map](image)

The different APIs are shown in Table 3.1. The three layers, Gadget Point, Segment Lines, and Abandoned Segment, represented the electric equipment, electric lines and abandoned/broken electric lines respectively. In addition, the layer Road represented the road of Guizhou Province. These layers fulfilled the requirements of field workers who inspect these elements.

Table 3.1: API of each layer

<table>
<thead>
<tr>
<th>Attribute</th>
<th>API</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gadget point</td>
<td>GADGET_ID</td>
<td>Numeric</td>
</tr>
<tr>
<td>Segment lines</td>
<td>SEG_ID</td>
<td>Numeric</td>
</tr>
<tr>
<td>Abandoned segment</td>
<td>SEG_ID</td>
<td>Numeric</td>
</tr>
<tr>
<td>Road</td>
<td>NAME</td>
<td>Text</td>
</tr>
</tbody>
</table>

One significant step in map creation was to choose the symbol and color to represent each layer on the map. The choice of symbol and design aimed to make each feature easy to view.
and distinguish on the small size screen of the Smartphone. Therefore, high-contrast colors and basic symbols were used.

The coordinate system of the test map was WGS 1984. The map was set to be displayed in different scales. This can help to improve mobile performance, particularly when working outdoors or in the field. To increase the internet browsing speed and improve the display capacity, users can choose the most suitable scale of each feature layer.

Choosing the whole range scale gives an overview of the entire test area, where the outline of urban, water, forest and cities are displayed. When the scale is set between 1:30,000 and 1:10,000, the test map shows features of road, segment lines, abandoned segment and gadget point. When the scale of test map is 1:10,000, the name of each road and the ID codes of the segment and gadgets are displayed. The maps at different scales are shown in Figure 3.3.

Figure 3.3: Maps in different scales: A. 1:40,000; B. 1: 15,000; C. 1: 6,000.

The World Topographic Map by ESRI (2013b) was also used in this application. It served as a background map for testing the GPS functionality.

### 3.3 Map Publishing

After the software mentioned earlier were installed, a PC acted as a map service provider, where external mobile devices can connect to the database. ArcSDE, ArcGIS for Server and Oracle all required the users to set the username and password. Moreover, some specific data
could only be accessed in the intranet of the company, which made it able to protect the privacy of data.

In order to publish a map service, first of all, it was necessary to connect Oracle and ArcGIS for Server. Next, the feature layers created with ArcMap should be transformed to Oracle by ArcSDE. After that, the map project could be published to ArcGIS for Server. ArcGIS for Server has several types of capabilities, like mapping, Web Coverage Service (WCS), Web Map Service (WMS), Feature Access, Schematics, mobile data Access, Keyhole Markup Language (KML), Web Feature Service (WFS), and Network Analysis. Here in this application, different feature layers were separately published. The Feature Access capability was needed because layers need to be accessed for query and edit functions. Otherwise, the Mapping and WMS capabilities were enough.

To make the map services in ArcGIS for Server operational, the link to each feature layer had to be created. The local host in the link needs to be replaced by the Internet Protocol version 4 (IPv4) if this link has to be accessed by other devices. After the links were created, three map services were published in the PC that allows querying and attribute editing. The first map service was for the background maps, as well as for the city and abandoned segments features. The second service only contained the electric features such as segment lines and gadget point. The third map service provided the road layer.

### 3.4 Application Design and Development

Eclipse was used to develop the mobile application. The entire programming codes are presented in the Appendix. Figure 3.4 displays the detailed Client-Service structure and interprets the procedures of data management functionalities provided.

The primary functions of this application are to query, locate, and update data. Therefore, the interfaces were designed according to these functions. As a GIS-based application, the query needs be able to visualize the results on the map. With the help of the GPS, workers should be able to locate their current positions, as well as the electrical distributions at their positions. Data update function should also be available for attribute editing in the field.
**Location and Query Task**

By clicking on the button *Location and Query Task*, a dialog which contains four items to access the location and query functions (i.e. *Electrical Line Query*, *Location Query*, *My Location* and *Electrical Meter Query*) is called out.

The *Electrical Line Query* function is to find out the specific electric wire by using the ID number (SEG_ID) in the attribute table of the *segment lines* layer. The interface of the dialog contains a text field, where users can type the code of an electric line, and two other buttons (*Submit* and *Cancel*). After typing the code and clicking the *Submit* button, the location of the electric line is highlighted, and at the same time the map is zoomed in to the selected feature.

*Location Query* and *Electrical Meter Query* has similar interface and result overview to the *Electrical Line Query*. The former is used for finding out the specific road on the basis of the *Name* indicated in the *road* layer, while the latter query button is for locating specific electrical meter/equipment based on the *Gadget ID* number in the *gadget point* layer.

*My Location* is for identifying the location of the user, with the help of GPS service on the mobile phone. After clicking this item, the location result, which is shown with the *World Topography Map* provided by ESRI (2013b) as the background map, is marked with a red point in the mobile device.

**Attribute View/Edit**

With a click on the *Attribute View/Edit* button in this application, all the names of the roads on the map can be viewed and edited. Once saved, the names in the database are also updated at the same time. The interface of this task is presented in Figure 3.5.

![Figure 3.5: Interface of Attribute View/Edit task](image)

There were two reasons for choosing roads as the edit feature class. Firstly, electric failure usually happens in a specific area, thus, workers can mark the road instead of the electric element if they do not know accurately which of the latter breaks down. Secondly, the database of roads is smaller than electric elements. Therefore, loading the roads for editing will improve the performance of the application.
3.5 User Testing

Since this application needed to be tested by potential users, H. Wang, a development engineer working for CSG, and three other workers (Workers A, B and C) in his group accepted the request to test the application in field by using the database of CSG on March 28, 2014. The user testing aimed to examine potential bugs or problems in this application, and to help evaluate the functionalities, advantages and disadvantages of the application.

The test devices used were Samsung Galaxy GT-N8000 and Samsung GT-9300, with the same China mobile SIM card that could provide the 3G cellular network. H. Wang published the maps of electric points, electric lines, and road of the real CSG database via the ArcGIS for Server to the service side of CSG. This step aimed to generate the new links to map services and APIs instead of the original.

Worker A was responsible for testing the functions of Electric Point Query and My location under the Location and Query Task. Firstly, the function was tested in both Samsung Galaxy GT-N8000 and Samsung GT-9300 to find the location of electric points with its numeric codes. Next, My Location function was examined in the same two devices, to get the accuracy of GPS when locating. Various electric points were separately tested in different places.

The test processes of Worker A were adopted by Worker B in evaluating the Electric Line Query function of the application with individual electric lines. Worker C tested the function Location Query with the roads. H. Wang made an assessment on Attribute editing in both devices by updating attributes of roads separately. He tested the functionality using both the test database created for the application in this study, and using the CSG database.

Users were also asked to comment on the application after the tests to get their overall impression of the application and to find out if there are issues to be addressed in its design and functionality.
4. Results

4.1 Functions of the Application

4.1.1 Interface Overview

Figure 4.1 shows the icon and main interface of the application. The test map is displayed in whole range scale with the feature layers of urban, forest, water and city points named from one to sixteen. The buttons Location and Query Task and Attribute View/Edit are at the top of the screen.

![Figure 4.1: Icon and the main interface of the application](image)

4.1.2 Map View/Zoom

With the help of ArcGIS for Server that provided the map view function, users can zoom in and out the map on an Android phone by finger operation, and the map can display and illustrate more detailed information. Figure 4.2 shows the map views at different scales. The left image shows the layers of urban, green, water and city. While in the middle, roads and electric network that consist of segment lines, abandoned segment and gadget point, are displayed. The right image shows the electric codes and the names.
Figure 4.2: Screen views in increasing map scales: a) layers of urban, green, water and city are visible; b) road and electrical networks consisting of segment lines, abandoned segment and gadget points are shown; and, c) at larger scale, the electrical codes and road names are displayed.

4.1.3 Location and Query Tasks

Figure 4.3 shows the four items under the Location and Query Task button: Electrical Line Query, Location Query, My Location and Electrical Meter Query. Users can choose the element they like to query by clicking on the corresponding item.

Figure 4.3: Items under the button Location and Query Task.
When one clicks on the different query tasks (i.e. *Electrical Line Query*, *Electrical Meter Query*, or *Location query*), a dialog box appears along with the two buttons *Submit* and *Cancel* (Figure 4.4). The text box allows users to type in each code of the electric element or the road name for querying. After entering the code, the selected lines and points are highlighted in cyan and shown on the screen.

![Figure 4.4: Using the different query tasks for: a) Electrical Line; b) Electrical Meter; and, c) Location](image)

Figure 4.4: Using the different query tasks for: a) *Electrical Line*; b) *Electrical Meter*; and, c) *Location*
When using My Location, the map shows the position using GPS. A red point showing the current location of the user is displayed on the screen with the map. The position is updated in real-time as one moves while carrying the mobile device. If the GPS does not work or is closed, a warning dialog will appear in the middle of the screen that will recommend to open the GPS, otherwise the function will not be available.

### 4.1.4 Attribute View/Edit

When clicking on the button Attribute View/Edit, a list containing all street/road names from the attribute of the road layer (Figure 4.5a) is shown. Users can edit and mark the attribute in the text field if needed to update the information or add some remarks. In the test examples, the first four attributes in the list view were edited: Beijing Road was changed to Beijing Road/Receiced by XG; Nanjing Road to Nanjing Road/Emergency; Ruidian Road to Ruidian Road/accident; and, Zhongguo Road to Zhongguo Road/Unknown Problem. Once the OK button beside the text field is pressed, the edited information for the particular road is updated (Figure 4.5b). If the Cancel button is pressed, the user will be returned to the map.

![Figure 4.5](image)

**Figure 4.5:** a) Example editing in the Attribute/Edit dialog; b) sample results after the attributes are edited.
4.2 Result of User Testing

Different functions of the application were tested in user testing. The results showed that this application was able to work sufficiently. The test results of the functionalities of the Location and Query Task in both Samsung Galaxy GT-N8000 and Samsung GT-9300 by Workers A, B, and C resulted to an acceptable response time. However the results of the location query varied in terms of the accuracy of the position shown.

Testing of the Attribute View/Editing task by H. Wang, which included the alteration of road attributes in both devices, resulted to one system crash (out of the five tests carried out) in one of the devices (i.e. Samsung Galaxy GT-N8000), when using the CSG database. On the other hand, with the evaluation using the test database created for this study, both devices performed well without crashing.

The users were also positive of the application particularly on the scale properties of the dynamic map, which they said to have improved the performance of this application; the clear layout; and the speed of completing the query tasks. However, they also commented that the allowed data type was restricted to some extent, and the query window should be simplified, as well.
5. Discussions

The results of this research showed how mobile GIS helped facilitate electric field work tasks. From the visualization perspective, it assisted workers in viewing the electrical distribution information on a digital map. In terms of data access, querying and editing data on the mobile device became possible. Nevertheless, this research has also some limitations.

Fu and Sun (2011) have stated that, there are some challenges that have to be addressed when developing a mobile GIS application, which are enumerated as follows: 1) the limitation of the system resource such as CPU speed, memory size, and battery power; 2) the limitation of bandwidth and network connections during the outdoor survey; and, 3) the small size of screen and keyboard, as well as the outdoor environment.

In this research, after the development and test of the application, several solutions can be suggested to help meet these challenges. First, in order to save the system resource, the online database was used instead of the mobile device memory. The functions were also simplified to enhance the CPU speed. To enhance online file transfer, the data types for this application only consisted of texts and vector graphics, which were smaller in size. Lastly, the map and interface were designed to make the contents visible and easy to recognize. Furthermore, more buttons were used rather than menus or toolbars.

As compared with other GIS applications such as PowerView (Shin & Feuerborn, 2004) and GINISED (Stoimenov et al., 2010) that pay more attention to data acquisition and provide more functionalities on data analysis, the application developed for this study focuses on the query function. Thus, it has more simplified function and better performance. Though applications such as GINISED and PowerView are able to require more electrical data types, more powerful devices and more professional operation skills, the permission for accessing electric data of the utility is difficult at present. Besides, since the application in the study is less confusing, it can be adopted by more field workers.

5.1 Applicability of Mobile-based GIS Application to Electric Field Work

In this study, it was illustrated how mobile GIS can be integrated into the practice of electric field work. First, GIS functions, which are good in visualizing and accessing data, can be adopted in the application. Second, since Google provides such an open free Integrated Development Environment, developers can create more features to make it more personal, dynamic, and flexible. A portable device, as well as an integration of multiple functions, is able to raise the efficiency of field work. Instead of going out in the field, workers can do preliminary check and search indoors by using the digital map.
Since digital geospatial data is not bounded by the desktop platform, web GIS and web mapping can help field work through mobile internet (Reichenbacher, 2001). Along with LBS and GPS, a mobile-based GIS application can provide location and querying functions by web mapping. In addition, with wireless supplement, workers can have faster access to data. If all of the procedures above are implemented, workers can be able to grasp a general condition of each electric site and come up with a maintenance scheme.

## 5.2 Problems in Developing Real-time Mobile-based Applications

In order to build the IDE, the hardware components such as computer and test mobile device are necessary. Each software and kits need to be installed without conflict. The database of the electric system should be quite large, which means that it is important to have a powerful database in the computer to activate the server side. Due to its real-time property, the web service must be operational all the time, and the wireless supplements of the Smartphone must stay in connection. Otherwise, this application would not work normally. In case that the data are regarded as private and sensitive, the access to the intranet should be restricted. To protect the data and ensure safety in accessing them, ArcGIS for Server, ArcSDE, and Oracle require username and password. In addition, ArcGIS for Server can also stop any map service if needed.

Due to the nature of a mobile application, it is crucial to select a matching OS to complete the whole development process. To maximize the rate of acceptance of this application by different target groups (i.e. authorities in electrical management), a widely-used system and mobile device should be the choice. Fu and Sun (2010) also stated that, it is a challenge for developing a mobile GIS application to choose the optimal OS that will cater the needs of most its users. Therefore, Android was valuable to be the main platform in this study. It was not only because of its dominant position in the mobile market, but also for its free resources, which is open for customers and developers. From this perspective, Android can save costs of application development.

The map design and user interface also played an important role in this research. To distinguish the electric features clearly on the small screen of the Smartphone, the symbols and colors should be considered and tested time after time. The reference coordinate system should also be taken into account if it is an application that will be based on location and GPS service. In this work, the positioning systems relied on both cellular phone signals and GPS signals, thus, they needed to be in the same coordinate system if users want to generate satisfactory results (Tsou, 2004). The test map used was part of Guizhou province, which was in WGS 84 reference system, which is commonly used around the world. The coordinates and geometric distribution laid the foundation for location and query functions.

During the user testing, there were also several concerns with the developed application that came out. First was the redundant code that can hardly be detected by the application. This can be the reason that caused the system crash during the test. Second, there were restrictions in the allowed data type. Third was the need to improve and simplify the query window.
Fourth, the accuracy of the location query varied during the testing, which could be affected by errors in the GPS as brought about by the unstable 3G network in the mountainous and cloudy conditions of Guizhou Province. But this error can possibly be reduced if the workers use this function with Wi-Fi network, or in a different weather condition, or use a different device to run the application. Nevertheless, the application operated smoothly.

5.3 Benefits of Electric Utilities from Real-time, Mobile-based GIS Applications

Users who did the evaluation were positive after they compared using the application with their traditional field work method, which was querying on paper map, recording updated data by writing and finally inputting it to the database of CSG through the computer once they returned. The results of the developed Android application for this study also presented the possibility of accessing each location of an individual wire and meter, visualizing the detailed geographic location of each single electrical unit by querying operation on a real-time web map service, and identifying user location by GPS. These possibilities are pointed out by Shin and Feuerborn (2004) as the essentials of an electric management system. In addition, Wei et al. (2010) indicate that GIS should be able to support real-time editing of data by operating directly on the graphical display interface.

Mobile-based GIS application can also help electrical utilities to deal with the querying and editing task, and field mapping on portable devices Smartphones with LBS supplement, instead of massive paper for manual drawing. As Virrantaus et al. (2001) mentioned, mobile devices can take advantage of real-time position provided by LBS. Furthermore, when there is a warning of an electric emergency, users can quickly acquire the necessary information (e.g. location, distribution) to deal with the problem. For example, when users know where the source of emergency is and the type of problem, they can organize workers to head on at once. Moreover, they can also be informed of the road network situation, thereby finding the fastest way to get to their destination. Hence, for electrical utilities, the application can be: time-saving for field inspection and decision making; resource-saving rather than drawing on massive paper maps; and, cost-saving in terms of using human resources and paying fees for on-site inspections. These benefits were also recognized in the development and implementation of PowerView (Shin and Feuerborn, 2004) and GINISED (Stoimenov et al., 2010), which were seen as helpful for electric utilities management.

5.4 Future Perspectives

Many researchers have stated that there are more GIS functionalities that will help facilitate electrical system management, such as the electric flow analysis, and users management system. Kale and Lad (2006) point out that the integration of SCADA, DMS and GIS can manifest their advantages to benefit the electric power system. It is hoped that this GIS
application can be a useful tool for enhancing different management capabilities for an electrical utility in terms of data, maintenance, and decision-making.

Researchers can make use of this study to analyze how the electric data can be managed and integrated on a dynamic map, how an Android application can be used with GIS technology, and how to build an IDE using the ArcGIS for Server and Android development tools. Besides, this study has also given a description of how to develop an Android application with GIS functionalities, and how it can be used in electric field work.

However, this application still has some limitations and needs to be improved. For example, it will be more user-friendly if the feature can be listed beside the mark on the map and if users can edit the attribute of each feature themselves.

To solve the existing problems mentioned in the previous section, further development and user testing have to be conducted. The application can also be improved to be able to be applied not only to electrical management, but to other kinds of GIS-based field works.
6. Conclusions

This study focused on the design and development of a mobile GIS application for electrical utilities, which allows users to view, query, and visualize the spatial data instantly on a web map service. The results showed that the mobile GIS application developed can help facilitate the electric field work through its available functionalities. For example, the digital map, which contains the geometric spatial features of the electric distribution network can be stored and called out from the database. Besides, the map service can also provide the APIs for mobile GIS application operations to query, locate, and edit the attributes online at the client side, thus offering convenience to field workers.

In general, this study had presented the applicability of mobile based GIS application to electric field work with GIS functions and geospatial data. People can directly access the location and data on a portable device. They can also know what need to be concern during the mobile-based applications development. As a consequence, mobile based GIS can be a useful tool to be applied to electric management.
7. Reference


Appendix

The programming codes of the application functions

This part shows and explains the main programming codes of each functionality in the application. For more detailed information, https://developers.arcgis.com/en/android/, which is provided by ESRI group, and the Help plug-in in Eclipse 4.2.1 can provide additional source codes and tutorials.

1. Make the application activity GuiZhouMobileGIS extend the Activity declaration of Android application, and define the elements which can be used:

```java
Public class GuiZhouMobileGISActivity extends Activity {
    Polyline poly = new Polyline();
    GraphicsLayer gLayer;
    ProgressDialog progress;
    ProgressDialog progress2;
    Point pointClicked;
    LayoutInflater inflator;
    Envelope initextent;
    ListView listView;
    View listLayout;
    Polygon polygon;
    Polygon polygonElec = null;
    Graphic[] grs;
    int selectedIndex = -1;
    List<Map<Geometry, String>> list = new ArrayList<Map<Geometry, String>>();
```

2. Loading the MapView service to display the map services from ArcGIS for Server, and put the background (BGmap), road (road1) and electrical (elect) layers to the map area of the main.xml layout in this application:

```java
Public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    MapView mMapView = null;
    mMapView.addLayer(BGmap);
    mMapView.addLayer(road1);
    mMapView.addLayer(elect);
    mMapView = (MapView)findViewById(R.id.map);
```

3. Define the url link of the background map (BGmap) for displaying:

```java
String BGmap = "http://192.168.1.100:6080/arcgis/rest/services/BackgroundMapService/MapServer";
BGmap = new ArcGISFeatureLayer("BGmap", MODE.SELECTION);
```
4. Define the url link of (road1) service, which contains the Road layer for querying. Give this layer the ArcGISFeatureLayer capability for feature accessing:

```java
String  road1 =
"http://192.168.1.100:6080/arcgis/rest/services/QueryRoad/MapServer ";
ArcGISFeatureLayer road1;
road1 = new ArcGISFeatureLayer("road1",MODE.SELECTION );
```

5. Define the url link of electrical layer service elect which contains SegmentLine and GadgetsPoint for querying. Also, give this layer the ArcGISFeatureLayer capability for feature accessing:

```java
String  elect =
"http://192.168.1.100:6080/arcgis/rest/services/ElectricPointLine/MapServer ";
ArcGISFeatureLayer elect;
elect = new ArcGISFeatureLayer("elect",MODE.SELECTION );
```

6. Add GraphicsLayer gLayer to MapView for the Highlight functionality in the application:

```java
GraphicsLayer gLayer;
gLayer = new GraphicsLayer();
mMapView.addLayer(gLayer);
```

7. Display these dynamic map service BGmap, elect, and road1:

```java
ArcGISDynamicMapServiceLayer BGmap = new ArcGISDynamicMapServiceLayer(BGmap);
ArcGISDynamicMapServiceLayer elect = new ArcGISDynamicMapServiceLayer(elect);
ArcGISDynamicMapServiceLayer road1 = new ArcGISDynamicMapServiceLayer(road1);
```

8. Display the background map "World Topography Map" and define the url link for this map service:

```java
ArcGISDynamicMapServiceLayer test = new ArcGISDynamicMapServiceLayer(
"http://services.arcgisonline.com/ArcGIS/rest/services/World_Topo_Map/MapServer");
```

9. Load the Location Manager Service (Location_Service) for controlling location updates:

```java
private LocationManager lm;
lm=(LocationManager)getSystemService(Context.LOCATION_SERVICE);
```

10. Set the display text in the two buttons Location and Query Task and Attribute View/Edit on the main.xml layout, findViewById is to make the feature on the layout can be called out by its ID codes:

```java
Button localButton;
Button Attribute;

localButton= (Button)findViewById(R.id.LocalButtom);
localButton.setText("Location And QueryTask");
```
Attribute = (Button)findViewById(R.id.attributeedit);
Attribute.setText("Attribute View/Edit");

11. Set the dialog listview.xml to show the names of each road when click on the Attribute View/Edit button:

    Attribute.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
       Dialog dialog = new Dialog(
GuiZhouMobileGISActivity.this);
       View view = LayoutInflater.from( GuiZhouMobileGISActivity.this).inflate( R.layout.listview, null);
       dialog.setContentView(view);
       dialog.setTitle("Street Info");
       ListView listview = (ListView) view.findViewById(R.id.listview);
       streetAdapter adapter = new streetAdapter(
GuiZhouMobileGISActivity.this, list,road1, dialog);
       dialog.show();
    }
    
12. Build new project file Street Adapter to get the functionality of Attribute View/Edit:

a) Define the StreetAdapter to extends the base Adapter from Android application:

    public class streetAdapter extends BaseAdapter { 
    private List<Map<Geometry,String>> list = new ArrayList<Map<Geometry,String>>();
    private LayoutInflater inflater;
    private Context context;
    private Map<Geometry,String> map = new HashMap<Geometry ,String >();
    private ArcGISFeatureLayer layer;
    private streetCom com;
    private Geometry geo = null;
    private String str = null;
    private Dialog dialog;

    public streetAdapter(Context context,List<Map<Geometry,String>> list ,ArcGISFeatureLayer layer,Dialog dialog)
    {
        this.context = context;
        this.list = list;
        this.inflater = LayoutInflater.from(context);
        this.layer = layer;
        this.dialog = dialog;
    
    c) Create the listener for Result update:

        CallbackListener<FeatureEditResult[][][]>
        createEditCallbackListener(final boolean updateLayer) {
            Return new CallbackListener<FeatureEditResult[][][]>()
{
Public void onCallback(FeatureEditResult[][] result) {
    if (updateLayer) {
        layer.refresh();
    }
}

d) Define the ListView of view.xml layout, One row has to be created, and the rest can be copied by the first row. Each row in the listview will display the different attributes:

    public View getView(int position, View convertView, ViewGroup parent) {
        com = new streetCom();
        map = list.get(position);
        for (Map.Entry<Geometry, String> mapEntry : map.entrySet()) {
            geo = mapEntry.getKey();
            str = mapEntry.getValue();
        }
        if (convertView == null) {
            convertView = inflater.inflate(R.layout.view, null);
            com.tv = (TextView) convertView.findViewById(R.id.widget35);
            com.et = (EditText) convertView.findViewById(R.id.widget36);
            com.OK = (Button) convertView.findViewById(R.id.widget37);
            com.cancle = (Button) convertView.findViewById(R.id.widget38);
            com.et.setText(str);
            com.OK.setOnClickListener(new OnClickListener() {
                @Override
                public void onClick(View v) {
                    Map<String, Object> attr = new HashMap<String, Object>();
                    String value = com.et.getText().toString();
                    attr.put("NAME", value);
                    Graphic gra = new Graphic(geo, null, attr, null);
                    layer.applyEdits(null, null, new Graphic[]{gra},
                                     createEditCallbackListener(true));
                    dialog.dismiss();
                }
            });
        } else {
            convertView.setTag(com);
        }
    }

e) View the attribute NAME of road1 map service in the text field (com.et), and apply the edits for the NAME attribute:

    public void onClick(View v) {
        Map<String, Object> attr = new HashMap<String, Object>();
        String value = com.et.getText().toString();
        attr.put("NAME", value);
        Graphic gra = new Graphic(geo, null, attr, null);
        layer.applyEdits(null, null, new Graphic[]{gra},
                         createEditCallbackListener(true));
        dialog.dismiss();
    }

f) Return to the map view when the Cancel button is clicked:

    com.cancle.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {
            dialog.dismiss();
        }
    });
    convertView.setTag(com);
}

g) Return to the map view when the text field is filled and OK button is clicked:

    Else {
        com.OK = (streetCom) convertView.getTag();
        com.et.setText(str);
    }
13. Set the dialog when the Location and Query Task button is clicked:

a) Show the Electrical Line Query dialog box:

```java
case 0:
final EditText powerInfoEditText = new EditText(GuiZhouMobileGISActivity.this);
new AlertDialog.Builder(GuiZhouMobileGISActivity.this).setTitle("Enter the Line code (Number)"
break;
```

b) Show the Location Query dialog box:

```java
case 1:
final EditText spaceInfoEditText = new EditText(GuiZhouMobileGISActivity.this);
new AlertDialog.Builder(GuiZhouMobileGISActivity.this).setTitle("Enter the Location Name")
.setIcon(android.R.drawable.ic_dialog_info).setView(spaceInfoEditText)
.setPositiveButton("Submit", returnSpaceInfoListener( spaceInfoEditText )).setNegativeButton("Cancel", null).show();
break;
```

c) Show My Location dialog box:

```java
case 2:
LocationGPS();
break;
```

d) Call the highlight function for GPS location:

```java
protected void LocationGPS() {
LocationService gpsLS=mMapView.getLocationService();
try{
    gpsLS.start();
gpsLS.setAutoPan(false);
gpsLS.setAccuracyCircleOn(true);
gpsLS.setAllowNetworkLocation(true);
gpsLS.setSymbol(new SimpleMarkerSymbol(Color.RED,20,SimpleMarkerSymbol.STYLE.CIRCLE));
mMapView.zoomToScale(gpsLS.getPoint(), 4000);
gpsLS.stop();
gpsLS.start();
}
```

e) Show the Electrical Meter Query dialog box:

```java
case 3:
final EditText PointInfoEditText = new EditText(GuiZhouMobileGISActivity.this);
```
break; }
});}.create();
default:
break;
}
returnnull;
}

14. Set the Query condition codes:

Query query=new Query();
query.setOutFields(new String[] {"Electrical Line Query", "Location Query", "My Location", "Eletrical Meter Query"});
query.setSpatialRelationship(SpatialRelationship.INTERSECTS);
query.setInSpatialReference(mMapView.getSpatialReference());
query.setReturnGeometry(true);
query.setWhere("name is not null");

15. Set functionality codes for Location Query. Define the search type, attribute and the url link of the layers for querying. This query task is an AsyncQueryTask, which means that other operations can be performed, and can run in the background at the same time:

```java
public DialogInterface.OnClickListener returnSpaceInfoListener(final EditText editText) {
return new DialogInterface.OnClickListener(){
public void onClick(DialogInterface dialog, int whichButton) {
if (editText.getText().toString()!=null)
{
StringBuilder strBuilder=new StringBuilder("NAME");
strBuilder.append(editText.getText().toString());
strBuilder.append("%'");
String queryStr=strBuilder.toString();
String[] queryParams = {
"http://192.168.1.100:6080/arcgis/rest/services/QueryRoad/MapServer/0", queryStr }
AsyncQueryTask ayncQuery = new AsyncQueryTask();
asyncQuery.searchType="road";
asyncQuery.execute(queryParams);
}
else {
Toast.makeText(GuiZhouMobileGISActivity.this, "Empty",Toast.LENGTH_LONG).show();}}
``` ed

16. Set functionality codes for Electrical Line Query. Define the search type, attribute and the url link of the layer for querying:

```java
public DialogInterface.OnClickListener returnPowerInfoListener(final
```
EditText editText ) {
    return new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int whichButton) {
            if (editText.getText() != null) {
                Toast.makeText(GuiZhouMobileGISActivity.this, editText.getText().toString(), Toast.LENGTH_LONG).show();
                StringBuilder strBuilder = new StringBuilder("SEG_ID = ");
                strBuilder.append(editText.getText().toString());
                String queryStr = strBuilder.toString();
                String[] queryParams = {
                        "http://192.168.1.100:6080/arcgis/rest/services/ElectricPointLine/MapServer/1", queryStr
                    };
                AsyncQueryTask asyncQuery = new AsyncQueryTask();
                asyncQuery.searchType = "line";
                asyncQuery.execute(queryParams); };
        }
    }
}

17. Set functionality codes for Electrical Meter Query:

public DialogInterface.OnClickListener returnPointInfoListener(final EditText editText) {
    return new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int whichButton) {
            if (editText.getText() != null) {
                Toast.makeText(GuiZhouMobileGISActivity.this, editText.getText().toString(), Toast.LENGTH_LONG).show();
                StringBuilder strBuilder = new StringBuilder("GADGET_ID = ");
                strBuilder.append(editText.getText().toString());
                String queryStr = strBuilder.toString();
                String[] queryParams = {
                        "http://192.168.1.100:6080/arcgis/rest/services/ElectricPointLine/MapServer/0", queryStr
                    };
                AsyncQueryTask asyncQuery = new AsyncQueryTask();
                asyncQuery.searchType = "point";
                asyncQuery.execute(queryParams); }
            }
    }
}

18. Assign codes for My Location to locate position on the map by GPS function:

a) Define the LocationManager:

    private void locationManager() {

b) Show a warning dialog with white background and red text in the center of the screen, prompting to either click the Setting button to set the GPS function of the mobile phone, or the Cancel button to return to the application:

    if (!lm.isProviderEnabled(LocationManager.GPS_PROVIDER)) {
        TextView textView = new TextView(GuiZhouMobileGISActivity.this);
        String info = "can't get the Coordinate Information.

        If you want
the Coordinate Information, 
Please Open GPS in System Settings 

```
textView.setText(info);
textView.setBackgroundColor(Color.WHITE);
textView.setTextColor(Color.RED);
textView.setGravity(Gravity.CENTER);
textView.setTextSize(20);
new AlertDialog.Builder(GuiZhouMobileGISActivity.this).setTitle("My Location")
    .setIcon(android.R.drawable.ic_dialog_info).setView(textView)
    .setPositiveButton("Setting", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int whichButton) {
            Intent intent = new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
            startActivityForResult(intent, 0);
        }
    }).setNegativeButton("Cancel", null).show();
```

```
c) Set the capabilities and criteria for different cases of GPS locating operation of the application:

    String bestProvider = lm.getBestProvider(getCriteria(), true);
    Location loc = lm.getLastKnownLocation(bestProvider);
    updateView(loc);
    lm.requestLocationUpdates(LocationManager.GPS_PROVIDER, 1000, 1,
    locationListener);
}
```

```jav
private LocationListener locationListener = new LocationListener() {
    public void onLocationChanged(Location location) {
        updateView(location);
    }
    public void onStatusChanged(String provider, int status, Bundle extras) {
        switch (status) {
            case LocationProvider.AVAILABLE:
                break;
            case LocationProvider.OUT_OF_SERVICE:
                break;
            case LocationProvider.TEMPORARILY_UNAVAILABLE:
                break;
        }
    }
    public void onProviderEnabled(String provider) {
        Location location = lm.getLastKnownLocation(provider);
        updateView(location);
    }
    public void onProviderDisabled(String provider) {
        updateView(null);
    }
};
```

```jav
GpsStatus.Listener listener = new GpsStatus.Listener() {
    public void onGpsStatusChanged(int event) {
        switch (event) {
            case GpsStatus.GPS_EVENT_FIRST_FIX:
                break;
            case GpsStatus.GPS_EVENT_SATELLITE_STATUS:
                GpsStatus gpsStatus = lm.getGpsStatus(null);
                break;
        }
    }
};
```

39
int maxSatellites = gpsStatus.getMaxSatellites();
Iterator<GpsSatellite> iters = gpsStatus.getSatellites().iterator();
int count = 0;
while (iters.hasNext() && count <= maxSatellites) {
    iters.next();
    count++;
} break;
    case GpsStatus.GPS_EVENT_STARTED:
        break;
    case GpsStatus.GPS_EVENT_STOPPED:
        break;
}}};

private void updateView(Location location){
    if(location!=null){
        String.valueOf(location.getLongitude());
        String.valueOf(location.getLatitude());
    }
    else{
    }
}

private Criteria getCriteria(){
    Criteria criteria=new Criteria();
    criteria.setAccuracy(Criteria.ACCURACY_FINE);
    criteria.setSpeedRequired(false);
    criteria.setCostAllowed(false);
    criteria.setBearingRequired(false);
    criteria.setAltitudeRequired(false);
    criteria.setPowerRequirement(Criteria.POWER_LOW);
    return criteria; }

19. Highlight the results of the querying task:

a) Show an Error warning if the result does not exist:

protected void onPostExecute(FeatureSet result) {
    String message = "Error";
}

b) Add new graphics to highlight the results, show dialog box indicating the number of results found:

if (result != null) {
    grs = result.getGraphics();
    if (grs.length > 0) {
        message = (grs.length == 1 ? "1 result has " : Integer.toString(grs.length) + " results have ") + "come back";
        ArrayList<String> strs=new ArrayList<String>();
        gLayer.removeAll();
        poly.setEmpty();
        for(Graphic gra:grs) {
            String str = null;
            strs.add(str);
        }
    Envelope env = new Envelope();
poly.queryEnvelope(env);
mMapView.setExtent(env);

    int size=strs.size();
strs.toArray(new String[size]);

progress.dismiss();
}
}
progress.dismiss();
Toast toast = Toast.makeText(GuiZhouMobileGISActivity.this, message, Toast.LENGTH_LONG);
toast.show();

c) Highlight the result of querying roads using a cyan polyline:

if(this.searchType.equals("road"))
str=(String) gra.getAttributeValue("NAME");
Graphic graphic = null;
graphic = new Graphic(grs[0].getGeometry(), new SimpleLineSymbol(Color.CYAN, 8));
poly.add((Polyline)graphic.getGeometry(),true);
gLayer.addGraphic(graphic);
mMapView.setExtent(gra.getGeometry());
}

d) Highlight the result of querying electrical meter using a cyan cross-hair:

else if(this.searchType.equals("point"))
{
Graphic graphic = new Graphic(grs[0].getGeometry(), new SimpleMarkerSymbol(Color.CYAN, 20, STYLE.CROSS));
gLayer.addGraphic(graphic);
mMapView.setExtent(gra.getGeometry());
}

e) Highlight the result of querying electrical line using a cyan polyline:

else if(this.searchType.equals("line")){
str= gra.getAttributeValue("SEG_ID").toString();
gLayer.removeAll();
Graphic graphic = null;
graphic = new Graphic(grs[0].getGeometry(), new SimpleLineSymbol(Color.RED, 8));
poly.add((Polyline)graphic.getGeometry(),true);
gLayer.addGraphic(graphic);
mMapView.setExtent(gra.getGeometry());
}

f) Show the result of GPS location using a red circle:

    protected void LocationGPS() {
        LocationService gpsLS=mMapView.getLocationService();
        try{
            gpsLS.start();
gpsLS.setAutoPan(false);
gpsLS.setAccuracyCircleOn(true);
gpsLS.setAllowNetworkLocation(true);
gpsLS.setSymbol(new

    Polyline)
SimpleMarkerSymbol(Color.RED, 20, SimpleMarkerSymbol.STYLE.CIRCLE));
mMapView.zoomToScale(gpsLS.getPoint(), 4000);
gpsLS.stop();
gpsLS.start();
}