Location-based Service Personal Navigation System for Nay Pyi Taw City

Myint Thu, Myint Myint Sein
University of Computer Studies, Yangon

myintthu@gmail.com, myintucsy@gmail.com

Abstract

With the development of mobile Internet, more and more people begin to get convenient service by mobile phones. Obtaining one's current location by GPS positioning or network positioning has become one of the important foundations in most applications of location based service. In this paper, we designed and implemented a personalized positioning and navigation system based on the Android platform. With the combination of GPS positioning and network positioning, and using OpenStreetMap API, this system provides the following functions: view the current location, define the current location by user and get the navigation route.

Keywords: *location; navigation; Android; OpenStreetMap (OSM).*

1. Introduction

In recent years, with the rapid development of mobile Internet, mobile phones are no longer just a tool for communication, more and more people hope to get more useful services by mobile phones. Location Based Service (LBS) is one of the popular applications in the field of mobile Internet. It obtains the location information (geographic coordinates or geodetic coordinates) of the mobile end-user through the wireless communication network (such as the GSM network, CDMA network) of communications carriers or external positioning (such as GPS or network), and then provides users with the appropriate service under the support of GIS (Geographic Information System) platform **[1]**. With the popularity of smart phones, obtaining one's current position by the GPS positioning or network positioning has become an important foundation of LBS applications. In this paper, we designed and implemented a personalized positioning and navigation systembased on the Android platform. Users can get convenient services from this system as described below:

A. Current Location

Users can get real-time positioning results that will be displayed on the map.

B. Navigation Route

Users can see the navigation route of the path between the start point and the end point on the map.

2. Android Platform and OSM

2.1. Android Platform

Android platform is of open system architecture, with versatile development and debugging environment, but also supports a variety of scalable user experience, which has optimized graphics systems, rich media support and a very powerful browser. It enables reuse and replacement of components and an efficient database support and support various wireless communication means. It uses a Dalvik virtual machine heavily optimized for mobile devices **[2]**. Android is a software stack for mobile devices that includes an operating system, middleware, and key applications. The Android SDK provides the tools and libraries necessary to begin developing applications that run on Android-powered devices **[3]**. As shown in Figure 1 **[4]**, Android architecture includes the following sections **[5]**:

1) Applications: A set of core applications are on the top level, including messages, web browser, contacts, etc. All apps are written using the Java programming language.

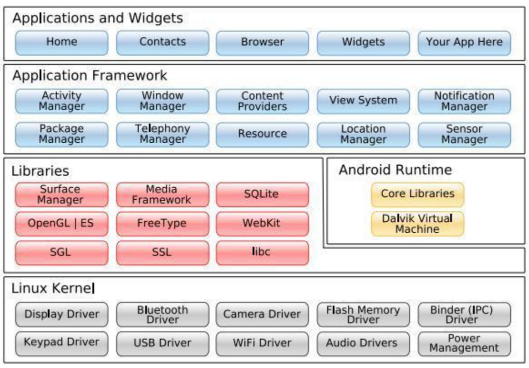

Figure 1. Android architecture

2) Application Framework: Developers have full access to the framework APIs used by the core applications. The application architecture is designed to simplify the reusing of all components. This

mechanism allows every component to be replaced by the user. Underlying all applications is a set of services and systems, including a rich and extensible set of Activities that can be used to build an application, TextView, ListView, Spinner, Button, MapView and so on. Content Provider enables applications to access data from other applications (such as Contacts), or to share their own data. Resource Manager makes non-code resources accessible from code. Notification Manager enables all applications to show custom alerts in the upper status bar. Activity Manager manages the life of each application and provides a useful navigation backtrack.

3) Libraries: Android includes a set of C/C++ libraries used by various components of the Android system. These capabilities are exposed to developers through the Android application framework. Some of the core libraries are listed in Figure 1.

4) Android Runtime: Android includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language. Every Android application runs in its own process given by the OS, and owns its own instance of the Dalvik virtual machine. The Dalvik VM executes files in the .dex (Dalvik Executable) format which was optimized for minimal CPU and memory usage. The Virtual Machine is register-based, and runs classes compiled by a Java language compiler that have been transformed at compile-time into the .dex format using the "dx" tool, that are shipped with the SDK.

5) Linux Kernel: Android relies on Linux (Kernel version 4.1) for core system services such as memory management, process management, network stack, security, and driver model. The core also acts as a hardware abstraction layer between the applications and all the hardware.

2.2. OpenStreetMap (OSM)

OSM is an open source map service which creates and provides free geographic data. OSM is an example of volunteered geographic information (VGI). People collect geographic data from several sources like a GPS devices and open source satellite imagery and later on upload that information to OSM's database; in this way the user can update, add or correct the map data in that area.

Open source is a free sharing system that can be contributed by everyone who wants to work on it and can benefit everyone who wants to use it. VGI is based on wiki technology as well and is a user generated geographic information content, and for which each volunteer is a sensor and is collecting geographic data, uploading the data to the server, editing the data uploaded or edited by other users and downloading the data for personal use, without considering that the user is a GIS expert or amateur. As for Wikipedia some issues are arising concerning the VGI such as data uncertainty or accuracy, human privacy, etc. **[6]**.

At the heart of OSM's technical infrastructure lies the central database holding the live data, which is implemented in MySQL. The database schema is designed to support wiki behaviors, such as versioning and rollbacks, and keeps copies of modified or deleted features indefinitely. All geographical entities are recorded as points (nodes), which contain the latitude and longitude coordinates along with user name and timestamp information. Linear and area features are defined by reference to a list of ordered nodes, called ways.

Access to the core OSM database is provided by a dedicated RESTful API, which is implemented in Ruby on Rails and supports authentication, enabling users to add, update, and delete geographical features. The API accepts and outputs data in OSM XML, a dedicated data transport format developed for the project that replicates the databases' specific entity model. All editing tools use this API for accessing and updating the main database. As a result, editing and presentation tools can be developed independently from the database, with the lightweight communication protocol acting as a glue between the elements of OSM's GeoStack **[7]**.

Before using the OSM service in Android applications, applying for an OSM Android API Key is pre-requisite to the development of the corresponding map function. With class MapActivity, MapView and OSM API, we can control the map easily on Android. Class LocationManager and LocationProvider are important in Android location services. LocationManager provides methods of getting the system location, therefore, positioning, tracking and approaching prompts are easy to accomplish with LocationManager. LocationProvider defines a method of providing location service (for example, provided by GPS or provided through a network, etc.). In order to get a suitable LocationProvider, we can use class Criteria to set some conditions we preferred. In addition, class Geocoder can complete the conversion between the address information and the latitude and longitude coordinates **[8]**.

Most of the smart mobile phones provide location services. Currently, Android platform supports GPS positioning and network positioning.

GPS positioning is completed by GPS satellite, it has a high accuracy and wide coverage, and it often fails when blocked by large buildings and always consumes more power, so it is suitable for outdoor positioning. Mobile network positioning is completed through the mobile operator's cellular mobile base station or WiFi access point, the location accuracy is lower but is suitable for indoor positioning.

2.2.1. Offline Use of OpenStreetMap

In this system, we designed and implemented an application for both online and offline use for navigation. Users can get offline service from this system when GPS and network connection is unavailable. Users can navigate by the distance location from Nay Pyi Taw City. Users can define current location on the map, search destination and see the navigation route on the map.

To get the required map data, we downloaded Nay Pyi Taw region as OSM format from OpenStreetMaps. Then we extracted the road network and generated the routing nodes to process path finding algorithm. We finally compressed using gzip algorithm for all the necessary data into one file and put the file inside the application. The application can un-compress when open and use the necessary road data to display, and load routing nodes for path finding.

3. System Design and Implementation

3.1. System Design

GIS applications, as well as several other applications, consist of three fundamental components: data, logic and interface. Where the data component represents the database, the logic represents the processing component and finally interface component refers to User Interface (UI). Using UI of the GIS software users perform actions for example queries or overlays which logic component is processing whereas logic component generally is relying on data in the database to generate the results **[9]**. Proposed system is stand-alone architecture where all three components: presentation, logic and data, are residing in the Smartphone.

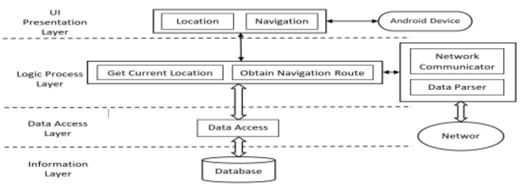

Figure 2. System architecture

As is shown in Figure 2, the system is divided into four layers as described below:

1) UI Presentation Layer: Be responsible for implementing friendly Android user interface that including location and navigation with different kinds of Android view component and Activity component.

2) Logic Process Layer: Receive instructions and data from UI presentation layer, then process the logic business including obtaining current location, obtaining navigation route, network communication, data parser and so on. Finally, return the results to UI presentation layer.

3) Data Access Layer: Be used by logic process layer to access data in the database.

4) Information Layer: Database. In order to save resources of Android client, user's data will be stored in the database of server.

3.2. Implementation

The implementation of each module is described in the following:

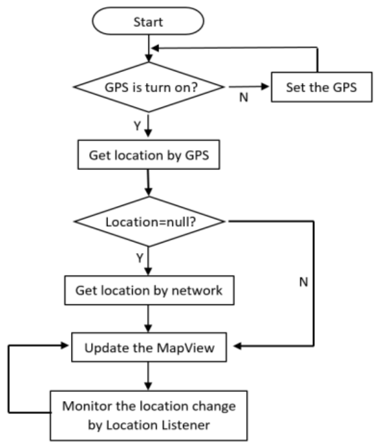

Figure 3. Flow of current location

1) Current Location Module: The flow of current location is shown in Figure 3. First, check whether the GPS of Android is turned on, if not, a dialog box will be popped up to alert user to turn on the GPS. After that, system tries to get user's current location by GPS positioning with function getLastKnownLocatin(), if the value of location is null, use network positioning instead. At the same time, add a LocationListener to monitor the change of location. When location is successfully obtained, get the latitude, longitude and detailed street address information by class Geocoder, then display location information on the map with MapView and Overlay. Finally, user can view the location details.

2) Navigation Route Module: The flow of navigation is shown in Figure 4. First of all, gather start point, end point and navigation mode that user input, then send a request for JSON format response packet with these parameters to the OSM by communication module. Afterwards, OSM will respond a corresponding JSON format packet. Parse the response packet with JSON parser and get the navigation route information that including key points between the starting point and the end point. Finally draw the navigation route on the MapView with custom Overlay.

3) Address Query Module: Convert the address information that users input into geographic coordinate information by class Geocoder, then mark the location on the MapView with custom Overlay.

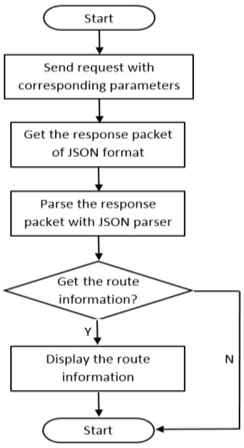

Figure 4. Flow of navigation

3. System Test Result

We test the function of this system on One Plus 3T Smartphone of Android 8.0.0. The screenshots shown below are part of the test results.

Figure 5 shows the result of current location on the map.

Figure 5. Current location

Figure 6 shows the search list to select the destination which is listed on the screen.

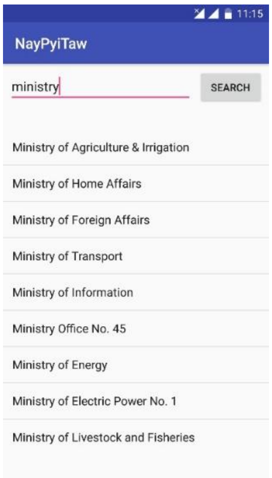

Figure 6. Search list to select destination

Figure 7 shows the navigation route which is drew on the map.

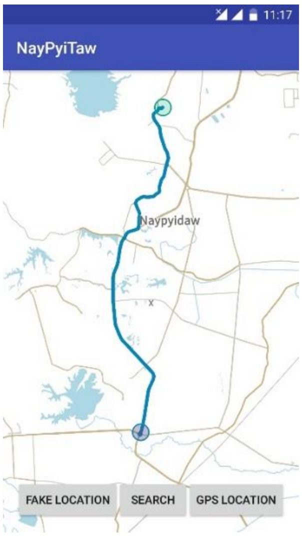

Figure 7. Navigation route

4. Conclusion

This location based personal navigation system consists of the following functions: viewing the current location, defining the current location by user and getting the navigation route. The user interface is friendly and each module works well, various LBS applications can be implemented based on this system. We hope to extend the functionality of the system by adding some other features such as getting address query and viewing historical location records in the future.

References

[1]Q Ren, M. H. Dunham, "Using Semantic Caching to Manage Location Dependent Data in Mobile Computing", In the 6th Annual International Conference on Mobile Computing and Networking, Boston: ACM Press, 2000, pp.210-222.

[2]Android - An Open Handset Alliance Project, http://code.google.com-/intl/zh-CN/android/.

[3]Android, https://developers.google.com/android/.

[4]Android Developers, http://developer.android.com/.

[5]Marko Gargenta and Masumi Nakamura, Learning Android, 2nd Edition, O'Reilly Media, Inc., CA 95472. USA, 2014.

[6]Jia T., Exploring Massive Volunteered Geographic Information for Geographic Knowledge Discovery, Royal Institute of Technology (KTH) Thesis, Sweden, 2010.

[7]Mordechai Haklay and Patrick Weber, OpenStreetMap: User-Generated Street Maps, IEEE Pervasive Computing, Volume: 7, Issue: 4, Oct.-Dec. 2008.

[8]Huai Yu Fan, Android development essentials. Machinery Industry Press, BeiJing, 2012.

[9]Peng Z. and Tsou M., Internet GIS: Distributed Geographic Information Services for the Internet and Wireless Networks, John Wiley & Sons, 2003.