

# Design and Implementation of a Group Management Application System

Kyubark Shim<sup>1</sup>, Jaegeol Yim<sup>2,\*</sup> and Sangheon Kim<sup>1</sup>

<sup>1</sup>*Dept. of Applied Statistics*

<sup>2</sup>*Dept. of Computer Engineering*

*Dongguk University at Gyeongju, Gyeongbuk, Korea*

*{shim, yim, tkdgs}@dongguk.ac.kr*

## Abstract

*This paper introduces a group management application system. One of the most important objectives of this system is letting the teacher know the current locations of students in real time. There are many cases where a teacher should know where students are but knowing the current locations of students is very difficult. For example, consider a field-trip of visiting a huge 5-story museum. Most of students disappear from the teacher's sight within a few minutes after entering the museum. If the teacher knows the current location of a student, the teacher may advise the student to visit some educational exhibits near to the student. After the field trip, the teacher may analyze the traces of students to find out the exhibit that is the most attractive to the students. This paper reviews group management related researches and summarizes the desired features that a group management system should offer. Our implementation of a naïve group management prototype is discussed.*

**Keywords:** *Fleet Management, Group Management, Android, Mobile Application, Global Positioning System*

## 1. Introduction

A group management system (GMS) is defined to be an application that helps group members have a field trip more efficiently and safely. A GMS provides many types of valuable services. For example, a teacher can be aware of current locations of all the students in real time during a field trip. A GMS can be seen as a kind of an indoor version of the fleet management service. In the early stage of location based service, fleet management was one of the hottest research topics. This paper reviews group management related research results.

After reviewing related researches, we discuss important services that a group management system should provide. Letting a teacher know the current locations of students is one of the basic services of a GMS. Whenever a student needs a help, he/she should be able to easily ask help to his/her teacher. It would be nice if parents of a student are able to know the current location of their child. There are many other services GMSs should provide.

Positioning, handling maps, handling a database and mobile programming are indispensable technologies for the development of GMSs. Making use of these technologies, we have developed a GMS prototype. The efficiency of the indoor positioning we have implemented is discussed.

---

Received (March 15, 2017), Review Result (June 21, 2017), Accepted (July 4, 2017)

\* Corresponding Author

## 2. Traditional Fleet Management

Traditional fleet management systems were location-based systems that managed a fleet of taxis, trucks, and vessels. Therefore, majority of them relied on the GPS (global positioning system) in order to obtain locations of moving objects. However, the civilian use GPS error was very big until several years ago. The differential global positioning system (DGPS) enhances the accuracy of GPS by using local reference base stations. A reference station knows the exact coordinates of the position where it is located. A reference station receives the GPS signal and calculates the difference between the exact coordinates and the received GPS coordinates. Then, it broadcasts the difference to the GPS users. These users adjust their GPS coordinates with the difference and achieve an accuracy of about 5 meters. Perez *et al.*, [1] designed a base station that broadcasts the difference through VHF technology.

An Intelligent Transportation System (ITS) monitors vehicles running in the nation and it can be considered to be a kind of fleet management system. In ITS, an autonomous vehicle location (AVL) attached on the vehicle is responsible to estimate the location of the vehicle. The AVL mainly relies on the GPS in determining its location and improves the accuracy with the dead reckoning technique. The AVL also uses the pre-defined points where it can always correctly determine its location [2].

## 3. Recent Fleet Management Systems

Backman *et al.*, [3] proposed an idea of enhancing existing fleet management systems with the Internet of things (IoT) technology. A set of functionalities that the IoT-based framework for fleet management system should support are discussed.

A fleet management system that improves road safety, energy efficiency and environment friendliness was proposed in [4]. In smart cities, fleet assets such as fire-engines, ambulances, trucks, drones, helicopters and so on should be efficiently managed in order to save lives and property in emergencies such as fires, traffic accidents, flooding, earth quakes, landslides, and so on. Polo *et al.*, [5] proposed a location-based decision support system that efficiently manages fleet assets in order to maximize the utility of assets, minimize the dispatch time, and minimize the cost of operating assets.

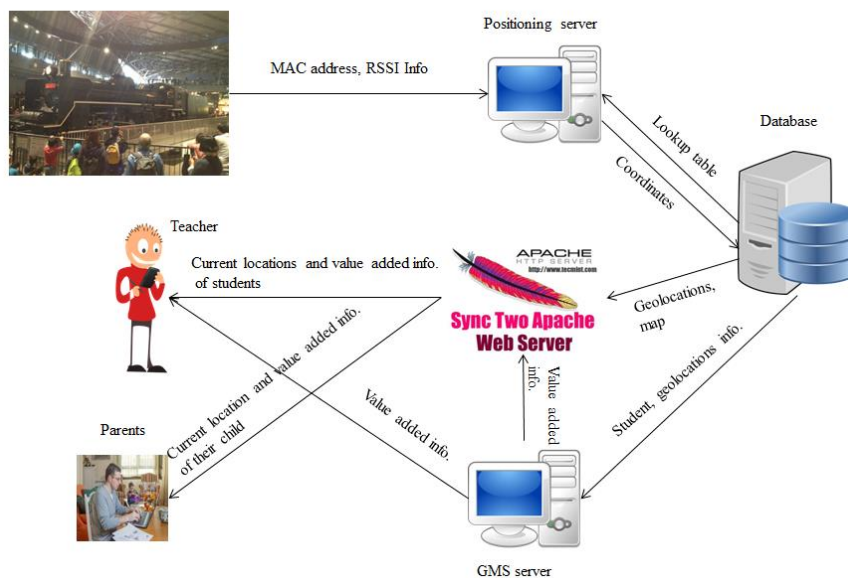
Hajibabai and Ouyang [6] introduced a snow plow fleet management system that minimizes the cost for truck deadheading and repositioning while maximizes the benefits of plowing. Lee *et al.*, [7] proposed a wireless sensor network based group management application with which a guardian can monitor group tourists while they are visiting a large museum or an exhibition hall. Using the ubiquitous medical sensor network, Tolentino and Park [8] developed a system that monitors patients all the time.

Won *et al.*, [9] proposed a management system for a fleet of dangerous goods vehicles. The server system of this management system consists of the path, journey, vehicle, accident management systems and is connected with the national intelligent transportation system to exchange information of road construction, accidents, and construction. Kang *et al.*, [10] proposed an Android application that can be used to prevent the spread of animal disease. The application system monitors vehicles and provides the information of "which vehicle visited or passed by which farms". Lee [11] discussed software technologies with which we can implement a group management application that provides the following functions: chatting service, providing contact information, photo gallery, schedule management, sharing location information.

## 4. Design of the Group Management System

Reviewing the related works suggests that a GMS should provide the following functions:

- Estimating the current location of each member of the group both indoor and outdoor
- Collecting the coordinates of the current locations of all group members.
- For each member of the group, it should record a sequence of locations where he/she has been located in a database.
- Exporting the database to other components of the GMS including the learning management system.
- Displaying a map or a floor map of the field trip site on the user interface of the GMS.
- Visually marking the current locations of the members of the group on the map.
- Allowing the parents of the group member to know the current location of their children through the Internet.
- Providing the navigation service to the members of the group.
- Providing the search and guide service to the members of the group. For example, if a student wants to use a restroom, then the system should show the route from the current location to the nearest (most convenient to use) restroom.
- Reminding the time schedule of the field trip to the members of the group.
- Recommending points of interest (POIs) that are worth of visiting considering user's context
- Allowing members of the group to talk to each other.
- Allowing members of the group to talk to the teacher, and vice versa.



**Figure 1. A Conceptual Model of Our Application System**

Considering the functional requirements of GMSs, we devised conceptual description of our GMS shown in Figure 1. Our GMS is a client-server system. In the server side, a positioning server, a database server, a web server and a GMS server can be found. In the database, information of students, floor maps, received signal strength indicator data, teachers, parents, and so on is recorded. The positioning server communicates with the students to collect sensor values from their smartphones and determines their current positions referring to the data, for example a look-up table if the positioning method is fingerprinting. The coordinates of the locations of students are sent to the database server in order to be stored in the database. The web server converts the coordinates into the geolocations referring to the electronic maps and displays the locations of the students on

the map if the teacher accesses the web site. The parents of a student can also find out where their child is located by accessing the web site.

The GMS server analyzes the data stored in the database in order to generate value added information such as the most popular point of interest, an optimal tour route, an optimal duration time of visiting, most demanded facilities, and so on. The GMS provides graphical user interfaces through which teachers can efficiently access the value added information. The value added information can also be accessed through the web site. Parents may find out which subject their child is most interested in through the web site.

For outdoor positioning, the global positioning system (GPS) is used. Using the GPS, smartphones can find out their current locations. For indoor positioning, many methods have been developed, but there is no general solution for indoor positioning, yet. The fingerprinting method [12] is adapted for our positioning server. A set of received signal strength indications (RSSI) from the beacons collected at a position is used as the fingerprint of the position. In other words, the general form of a fingerprint is a set of (RSSI, beacon ID) pairs:  $\{(RSSI_1, beacon_1), \dots, (RSSI_n, beacon_n)\}$ , where the minor value can be used as the beacon ID. Given a fingerprint, the `getCoordinate( )` method described in Figure 2 finds K nearest reference points and returns the weighted average of the K points as the coordinates of the position where the fingerprint is obtained.

```

public int[] getCoordinate(ArrayList<AverageRSSI> fingerprint, double[][] lookupTable,
double[][] var_Arr, int K) {
    For each row r in the lookupTable{
        For each Minor in the fingerprint {
            Find the index, col, corresponding to Minor;
            gap += getGap(... rssi, r, col);
        }
        setSmallGapIndexArr(small_gap_index_list_Arr, row_index, gap);
    }
    return gapAverage(small_gap_index_list_Arr);
}
    
```

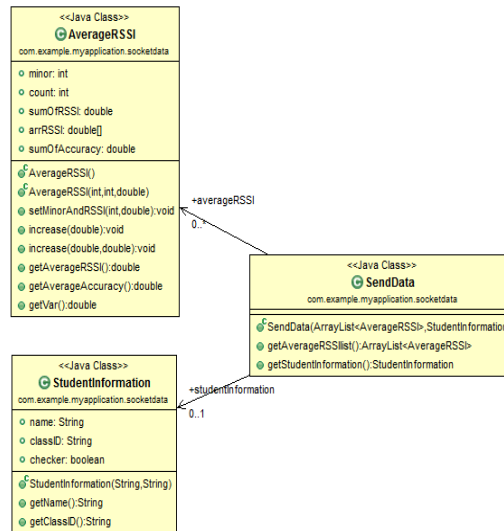
**Figure 2. Our Algorithm to Determine the Current Position of a Smartphone**

The uniqueness of our algorithm can be found in the `getGap( )` method. It uses the standard deviation of RSSI values received from the i-th beacon as the weight of the i-th beacon [13]. The strategy employed by the `getGap( )` method to estimate the difference between `fingerprint[Minor]` and `lookupTable[r, col]` is summarized in Table 1. In the table, `F[M]` represents “fingerprint[Minor]” and `IT` represents “lookupTable”. For example, the first row represents that if `fingerprint[Minor]` is greater than -85 and `lookupTable[r, col]` is NULL then the gap, or difference, is 10,000.

**Table 1. Our Strategy to Estimate the Difference between fingerprint[Minor] and lookupTable[r, col]**

F[M]!= NULL	IT[r, col]== NULL	F[M]>= -85	F[M]-IT[r, col] >= 20	gap
O	O	O		10,000
O	O	X		(200-F[M])/divisor
O	X		O	10,000
O	X		X	(IT[r,col]-F[M])/divisor
X	O			0
X	X			(IT[r,col]-200)/divisor

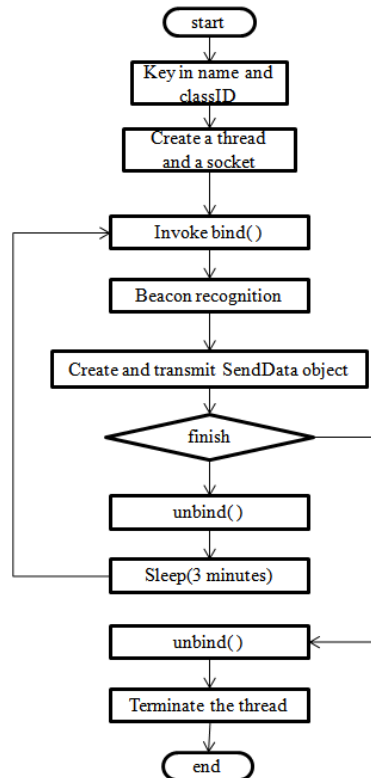
In the database, we have studentTable, parentsTable, teachersTable, and permissionTable for user management. For the purpose of storing students' position information, we have currentPositionTable and allPositionsTable. The attributes of both of these two tables are studentName, X, Y, and Date. A tuple of this table represents that the coordinates of the position where the student was located at Date is X, Y. However, the manners of updating these two tables are different. When the currentPositionTable is updated, it finds the tuple with the same studentName and modifies the X, Y, and Date of this tuple with the new values so that only students' current locations are recorded. On the other hand, updating the allPositionTable appends the new tuple to this table so that all students' movement paths can be recorded.



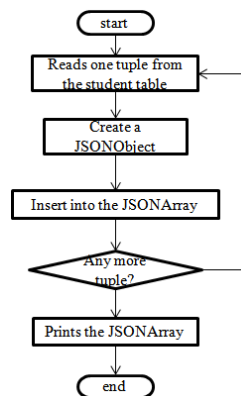
**Figure 3. A Class Diagram for the SendData Class**

The main task of the client application running on students' smartphones is to periodically collect RSSI values and send them to the server. The SendData class is used as the container for the data to be sent. A class diagram for the SendData class is shown in Figure 3. An object of the SendData class consists of objects of AverageRSSI and StudentInformation. The process of the StudentClient that transmits the data to the server is described in Figure 4. From Android 3.0, a NetworkOnMainThreadException exception is thrown when the main thread creates a socket. Therefore, our StudentClient creates a thread. The thread repeats the process of collecting RSSIs from the beacons, transmitting the collected RSSIs, and sleeping for 3 minutes until the finish button is clicked.

StudentLocationPage1, 2 read all data from currentPositionTable and allPositionsTable, respectively and print it to the web page in JSON format. All tuples are put into a JSONArray and an object of this JSONArray is printed out as shown in Figure 5.



**Figure 4. A Flowchart Describing the Process of the StudentClient**



**Figure 5. A Flowchart Describing the Process of StudentLocationPage1 and 2**

## 5. Implementation of a Prototype

Android system provides API (Application Program Interface) with which we can obtain the location of an Android mobile device. We have to specify permission in the manifest file.

```

<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" >
</uses-permission>
<uses-permission android:name="android.permission.ACCESS_MOCK_LOCATION">
</uses-permission>
    
```

```
public double getGap(  
    ...  
    double divisor= Math.sqrt(Math.sqrt(var_Arr[row_index][column_index]));  
    if (rssi != -9999) {  
        ...  
        gap=Math.abs((average_Arr[row_index][column_index] + (-1 * rssi)) / divisor);  
    }  
    ...  
    gap=Math.abs((average_Arr[row_index][column_index]+(-1*-200))/divisor);  
    ...  
    return gap;  
}
```

**Figure 6. Part of the getGap( ) Method**

```
public class SendData implements Serializable {  
    public ArrayList<AverageRSSI> averageRSSI;  
    public StudentInformation studentInformation;  
    public SendData(ArrayList<AverageRSSI> averageRSSI, StudentInformation  
studentInformation)  
    {  
        this.averageRSSI=averageRSSI;  
        this.studentInformation=studentInformation;  
    }  
    ...  
}
```

**Figure 7. Part of the SendData Class**

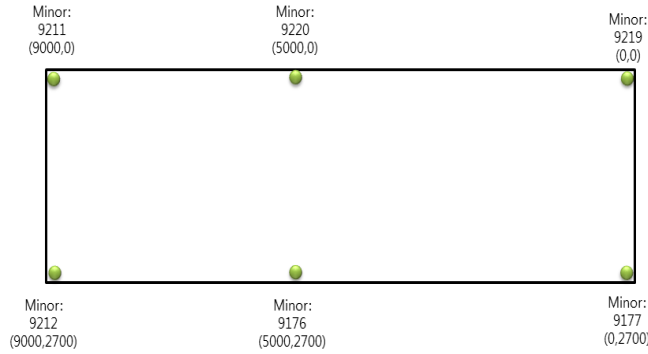
In the Android application, we invoke the requestLocationUpdates method in the LocationManager class. Before the invocation, the GpsLocationListener should have been implemented.

```
mGpsLocMan = (LocationManager) getSystemService(LOCATION_SERVICE);  
mGpsLocMan.requestLocationUpdates(LocationManager.GPS_PROVIDER, 5000, 3,  
GpsLocationListener);
```

Part of the getGap( ) method is shown in Figure 6. The divisor variable is used as the weight of beacon. If we use a constant as the divisor, then the gap is the Euclidean distance. Part of the SendData class is shown in Figure 7. As the container of the data to be sent to the server, it has an ArrayList consisting of averages of RSSIs and StudentInformation.

## 6. Experiments

For indoor positioning, we use RSSIs received from beacons. Using the API provided by the beacon manufacturer, RSSIs can be received. We installed six beacons on the walls as shown in Figure 8. Distances between two neighbor reference points were set to 1 meter. At each reference point, RSSIs were received 50 times. A fingerprint of a reference point is a set of averages of these 50 RSSIs. For example, the fingerprint of the reference point (100, 100) can be {(9211, -73.28), (9212, -88.16), (9220, -85.8), (9176, -85.79), (9219, -90.56), (9177, -91.37)}.

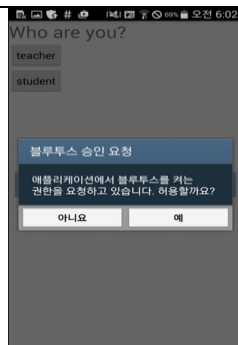


**Figure 8. Six Beacons were Installed on the Walls**

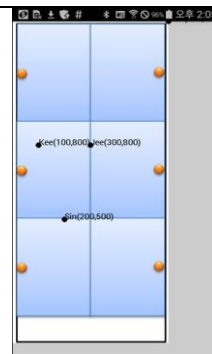
We have tested our indoor positioning with various divisors (refer to Figure 6) such as standard deviation (SD), square of the SD, and so on as shown in Table 2. We obtained the best result when we used the 4th power of the variance as the divisor.

**Table 2. Summary of Our Test Results**

divisor	Error
1	163 cm
Variance (Var)	196 cm
Standard Deviation	226 cm
4 <sup>th</sup> power of Var	130 cm
8 <sup>th</sup> power of Var	134 cm



**Figure 9. Asking for the Bluetooth Communication**



**Figure 10. Displaying the Current Locations of all Students**

The screenshot shown in Figure 9 asks permission from the smartphone owner to activate the Bluetooth communication while the screenshot in Figure 10 displays the current positions of all students. We have many other screenshots including a screenshot displaying the path taken by a given student and a screenshot that shows a list of the students who belong to the “dg1” class along with their current locations if a user connect to “Where Are You” and specify “dg1” as the class name.

## 7. Conclusion

In the early stage of location-based service, fleet management was one of the hottest research topics. After reviewing fleet management systems, this paper proposed and designed a group management system (GMS). The designed system can be considered to be comprehensive in that most of the features provided by existing fleet management systems are adapted for our GMS. A prototype of the designed GMS has been



implemented. Most of the techniques needed to implement the designed GMS have been exercised in the implementation of the prototype. An indoor positioning system, mobile applications for teachers and students, web systems for teachers and parents, and a mobile messenger application are included in the prototype. For the future work, we are developing a practical group management system.

## Acknowledgments

This research was supported by the Dongguk University Research Fund of 2016, by the Ministry of Education (NRF-2011-0006942), and by the Ministry of Knowledge Economy (10037393).

## References

- [1] C. Feijoo, J. Ramos, F. Perez, 'A system for fleet management using differential GPS and VHF data transmission mobile networks', Proceedings of the Vehicle Navigation and Information Systems, (1993) October 12-15; Ottawa, Canada, pp. 445-448.
- [2] H. E. Gerland, 'ITS Intelligent Transportation System: fleet management with GPS dead reckoning, advanced displays, smartcards, etc', Proceedings of the Vehicle Navigation and Information Systems, (1993) October 12-15; Ottawa, Canada, pp. 606-611.
- [3] J. Backman, J. Väre, K. Främling, M. Madhikermi and O. Nykänen, 'IoT-based interoperability framework for asset and fleet management', Proceedings of the IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), (2016) September 6-9; Berlin, Germany, pp. 1-4.
- [4] E. Natale, M. Tufo and A. Salvi, 'A fleet management service for smart cities: The S2-move project', Proceedings of the International Smart Cities Conference (ISC2), (2016) September 12-15; Trento, Italy, pp. 1-6.
- [5] A. Polo, F. Viani, F. Robol, A. Massa, S. Marchesi and L. Zappini, 'Optimization strategies for fleet management based on wireless terminals localization in smart cities scenarios', (2016) September 12-15; Trento, Italy, pp. 1-6.
- [6] L. Hajibabai, Y. Ouyang, 'Dynamic Snow Plow Fleet Management Under Uncertain Demand and Service Disruption', IEEE Transactions on Intelligent Transportation Systems (2016), Vol. 17, Issue 9, pp.2574-2582.
- [7] S. Lee, M. Kim, Y. Kyung, K. Jung, J. Kim, Y. Lee and K. Eom, 'A Design of U-system for Group Management Using Wireless Sensor Network and Android Device', IJAST (2011), Vol.35, pp.61-72.
- [8] R. S. Tolentino and S. Park, 'A Study on U-Healthcare System for Patient Information Management over Ubiquitous Medical Sensor Networks', IJAST (2010), Vol. 18, pp.1-10.
- [9] J. Won, J. Kim and Y. Kwon, 'Development for the Integrated Management System for the Safe Transport of Dangerous Goods Vehicles', IJCA (2015), Vol. 8, No. 9, pp.371-380.
- [10] B. Kang, H. Kim and H. Yoe, 'A Study on the Android Based Livestock Vehicle Management System', IJMUE (2014), Vol.9, No.2, pp.87-94.
- [11] S. S. Lee, 'A Group Management Application in Consideration of Software Engineering: Design and Implementation', IJSEIA (2015), Vol. 9, No.2, pp.107-124.
- [12] J. Yim, 'Introducing a decision tree-based indoor positioning technique', Expert Systems with Applications (2008), Vol. 34, Issue 2, pp. 1296-1302.
- [13] J. Yim, 'Introducing Weighted Fingerprint Indoor Positioning', International Journal of Smart Home (2016), vol.10, No.10, pp.193-202.

## Authors



**Kyubark Shim**, He received the M.S. and the Ph.D. degrees in Statistics from the Dongguk University in 1986 and 1993, respectively. He is a Professor in the Department of Applied Statistics at Dongguk University at Gyeongju, Korea. His current research interests include Computational Statistics, Reliability Test, Analysis of Statistical Data.



**Jaegeol Yim**, Jaegeol Yim received an MSc and PhD in Computer Science from the University of Illinois at Chicago, in 1987 and 1990, respectively. He is a Professor in the Department of Computer Engineering at Dongguk University at Gyeongju Korea. His current research interests include Petri net theory and its applications, location-based services, AI systems, and multimedia systems. He has published more than 50 journal papers, 100 conference papers (mostly written in the Korean Language), and several undergraduate textbooks.



**Sangheon Kim**, Sangheon Kim received a B.E. in Computer Engineering from Dongguk University at Gyeongju in 2017. He worked with ITSEN Cooperation in Korea. His current research interests include mobile application and system programming.