J. Internet Comput. Serv.
ISSN 1598-0170 (Print) / ISSN 2287-1136 (Online)
http://www.jics.or.kr
Copyright ⓒ 2022 KSII

Design and Implementation of a Service Platform that Recommends the Optimal Shortest Distance as a Patrol Route☆

Yu-min Jo[1] Ye-jin Jang[1] Jong-ho Paik[2*]

ABSTRACT

Recently, interest in living safety and crime prevention is increasing. The reality is that most women have anxiety about social safety and ultimately want a safe return to home. As a result of these issues, the Seoul Metropolitan Government and the National Police Agency are implementing various services to alleviate them. However, there are limitations such as that the service can be used only during a limited time or the process of checking whether the patrol is really completed is complicated. Therefore, in this paper, we propose a service platform that overcomes these limitations and suggests the best and shortest patrol route to the police based on the desired patrol location applied by citizens. It is designed based on the MVC pattern, and the functions are divided for each user. It is hoped that the platform will reduce crime rates and allow citizens to ultimately return home with peace of mind. Also we expect that the police will ablet to find places where they did not know about or need to patrol with more attention through the recommended route of the platform, which will be helpful in their task.

☞ keyword : Patrol Service, MVC Pattern, Database, MySQL, Server, Node.js, Android Application, Recommended Algorithm, Path Finding Algorithm

1. Introduction

With the recent increase in single-person households, interest in living safety and crime prevention has increased. According to the National Statistical Office data in 2018, 35.4 percent of women feel anxious about social security, while 57 percent of women say they are "unstable because of the occurrence of crimes." According to the Anti-Corruption and Civil Rights Commission's analysis of big data on public complaints from 2016 to 2019, 317,683 complaints were related to CCTV in Seoul, accounting for 12.47 percent of the total complaints. Most of the complainants were women. This analysis indicates that women feel a significant threat to social safety[1].

For women suffering from anxiety on their way home, the Seoul Metropolitan Government has been implementing the 'Women's Safe Home Scout Service' since 2013. The Safe Return of Women Scout is a service that helps two scouts selected by the Seoul Metropolitan Government to accompany the applicant to the applicant's home and help them return home safely. However, the service is only available from 10:00 pm to 1:00 am the next day[2]. The time was designated, taking into account the possibility of a crime. Since the period for using the service is limited, it is questionable how much it will have on crime prevention.

In addition, the National Police Agency is currently patrolling using the Enhanced Patrol System. The enhanced patrol system uses a web service called "Sunchal-Sinmungo." In more detail, when citizens apply for the time and place they want to patrol through the web service, the police refer to the data and patrol. However, the service has no way for citizens to know in real-time whether the police have completed a patrol or not. There is a problem in that it is cumbersome to check whether the patrol has been completed or not by e-mail. In addition, it is difficult for the user to know precisely each point of the location where the police have completed patrol. Since all current patrols are done

[1] Major in Software Fusion, Seoul Women's University, Seoul, 01797, Republic of Korea

[2] Dept. of Software Convergence, Seoul Women's University, Seoul, 01797, Republic of Korea

[*] Corresponding author (paikjh@swu.ac.kr)

☆ This work was supported by a research grant from Seoul Women's University(2021-0199).

[Received 03 November 2021, Reviewed 25 November 2021, Accepted 09 December 2021]

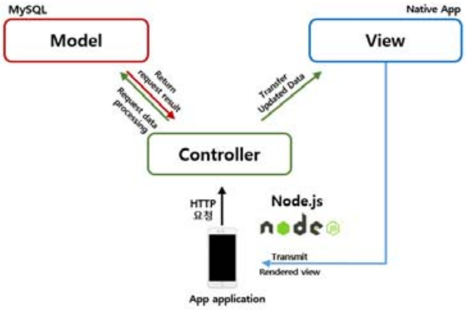
http://dx.doi.org/10.7472/jksii.2022.23.1.1

manually, there is a limitation in that it is difficult even to meet the needs of all users.

In this paper, we propose a service platform that recommends the optimal shortest distance as a patrol route. By improving the National Police Agency's existing system, the proposed platform includes a function that provides real-time patrol completion feedback to users and a route recommendation function. These functions are expected to increase the efficiency of police patrol work. In addition, CCTV-related functions are added to the proposed platform. When the police detect a CCTV blind spot during a patrol or immediately send information about a broken CCTV or street light to the ward office, this function provides the convenience of not having to file civil complaints with public institutions. As users and data accumulate, the opportunity to lower the crime rate will increase.

2. Design of Service Platform

The overall configuration of the service platform covered in this paper is based on the MVC pattern, as shown in Figure 1. MVC pattern stands for 'Model-View-Controller' and is a frequently used software design pattern in software engineering[3]. On this proposed platform, the model uses MySQL, the view uses Android's xml, and the controller uses Node.js.



(Figure 1) MVC Pattern Plot

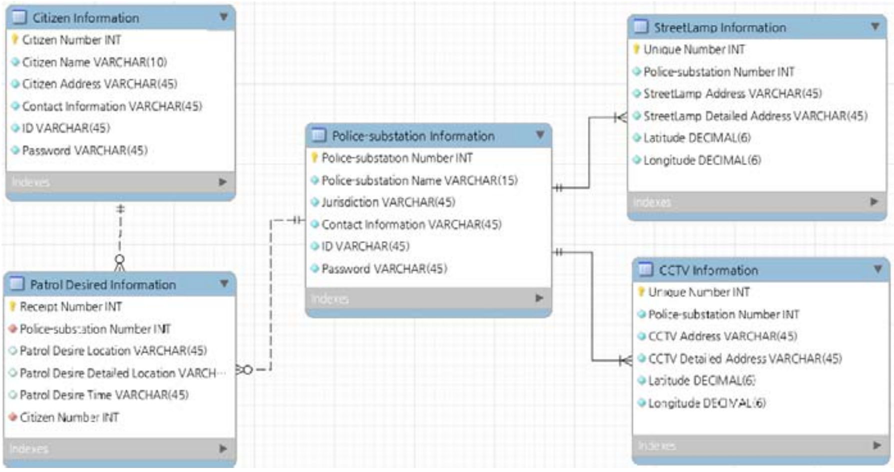
2.1 Model

Model is used when designing a database and enables query processing of data. When designing the platform's database covered in this paper, it consists of 5 tables, as shown in Figure 2. First, we need a table to store and manage the information of citizen who uses the platform. In the citizen information table, citizen number, which can be used as the primary key, citizen's name, citizen address, contact information, ID, and password are required as attributes.

Similarly, a police substation using the platform also needs a table to store information. The police-substation information table requires attributes such as the police-substation name, jurisdiction, contact information, ID, password, as well as the primary key, the police-substation number. When a citizen stored in the citizen information table wishes to be patrolled, a table capable of storing and managing the place and time information is also required. The patrol desire table has the primary key of the registration number, the patrol desire address, patrol desire detailed address, patrol desire time attributes, and the citizen number, which is a foreign key for connection with the member table. As a foreign key for contact with the Police-substation, the police-substation number attribute is also required. At this time, the reception number should have a form that is easy to delete when the amount of data increases later. For example, the first complaint number received on June 12, 2020, can be stored in the form of '202006120001'.

In addition, since it is possible to be patrolled in several places for each citizen, the 1: N relationship is required for the citizen information table. The patrol desired information table and the police substation is determined according to the desired locations of the patrol. Hence, the patrol desired table and the police-substation information table are required N: 1 relationship. Through this, it is possible to create a patrol route after selecting the desired location and time for the patrol, which is the overall first function of the platform.

As the patrol progresses, patrol officers can find the matters such as CCTV or streetlamps and file the complaints directly to the ward office. So a CCTV information table and a streetlamp information table are required separately. Both tables require the unique number as a primary key, address, exact address, latitude, longitude for attributes. The latitude and longitude attributes are needed to set the marker in the View. In addition, since both tables have a


Citizen Information
Citizen Number INT
Citizen Name VARCHAR(10)
Citizen Address VARCHAR(45)
Contact Information VARCHAR(45)
ID VARCHAR(45)
Password VARCHAR(45)
Indexes
Patrol Desired Information
Receipt Number INT
Police-substation Number INT
Patrol Desire Location VARCHAR(45)
Patrol Desire Detailed Location VARCH...
Patrol Desire Time VARCHAR(45)
Citizen Number INT
Indexes
Police-substation Information
Police-substation Number INT
Police-substation Name VARCHAR(15)
Jurisdiction VARCHAR(45)
Contact Information VARCHAR(45)
ID VARCHAR(45)
Password VARCHAR(45)
Indexes
StreetLamp Information
Unique Number INT
Police-substation Number INT
StreetLamp Address VARCHAR(45)
StreetLamp Detailed Address VARCHAR(45)
Latitude DECIMAL(6)
Longitude DECIMAL(6)
Indexes
CCTV Information
Unique Number INT
Police-substation Number INT
CCTV Address VARCHAR(45)
CCTV Detailed Address VARCHAR(45)
Latitude DECIMAL(6)
Longitude DECIMAL(6)
Indexes

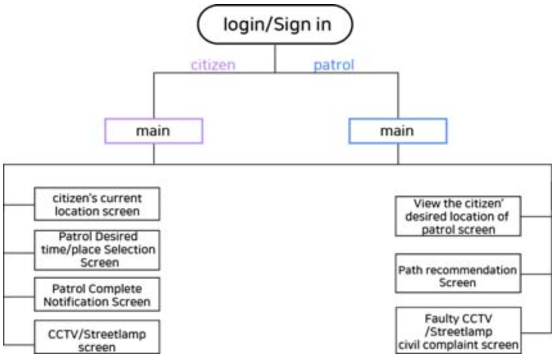
(Figure 2) E-R Diagram

relationship about one police-substation corresponding to multiple CCTV / streetlamps, the relationship between each table and the police-substation information table can be defined as N: 1 with the foreign key, which is the police-substation number attribute. Therefore, it can be seen that the model covered in this paper requires a relational database management system that uses a query language (SQL). In conclusion, we proposed to use MySQL, which is easy to use, flexible, and typically widely used[4].

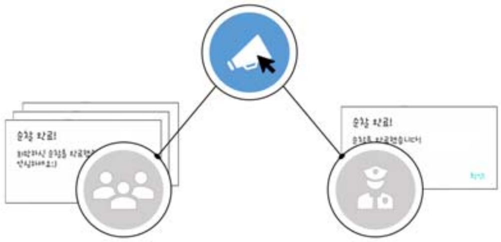
2.2 View

A view is a screen that the user can see directly. The service platform proposed in this paper provides the user with a screen designed based on the data stored in the MySQL DB at the request of the client. The proposed service platform uses Android apps for smartphones. Therefore, the screen shown to the user is implemented in XML.

The overall screen configuration is shown in Figure 3. As shown in Figure 3, the screen composition is determined through login and login according to citizens or patrol members. The screens used by citizens are divided into the current location screen, the viewing screen such as CCTV/street lights, and the desired place and time. Selection screen, patrol completion notification screen. The screen used by the patrol unit is divided into the desired patrol screen location view, route recommendation screen, and CCTV/street light defect report screen. Figure 4 is a representative screen design example, showing the patrol completion notification message displayed to each user after the patrol is completed.


login/Sign in
citizen
patrol
main
main
citizen's current location screen
Patrol Desired time/place Selection Screen
Patrol Complete Notification Screen
CCTV/Streetlamp screen
View the citizen' desired location of patrol screen
Path recommendation Screen
Faulty CCTV /Streetlamp civil complaint screen

(Figure 3) Structure of the Screen Classified by the Function.

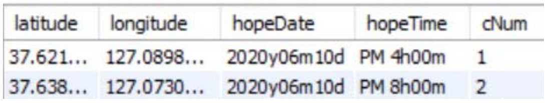


(Figure 4) Patrol Completion Notification Screen for Each User

2.3 Controller

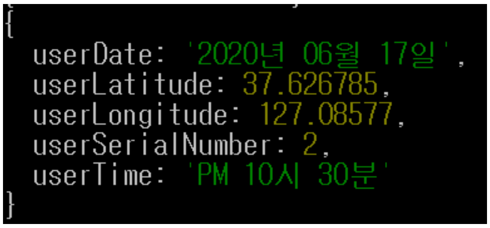
To create a good performance platform, we need to build a server with a structure that can handle parallelism in real time[5]. In this respect, Node.js is an asynchronous I/O-based architecture suitable as a real-time server capable of parallel processing processes. The targets of the platform covered in this paper are classified into citizen and patrol officers (or police). The proposed service platform uses the Node.js server to enable immediate requests and responses. This Node.js advantage allows for fast interaction between two users.

When the Controller receives a request from the client, it requests the Model to process the data. At this time, for security reasons, Android cannot access MySQL DB independently and directly. At this point, Node.js acts as an intercessor that connects DB and Android. The server loads the data stored in each column of the DB and converts it into JSON format. Figure 5 is an example of a column stored in the DB. And Figure 6 shows the screen displayed on the console window by changing the data stored in the DB to JSON. As shown in Figure 6, we can freely use the refined data in Android.



latitude	longitude	hopeDate	hopeTime	cNum
37.621...	127.0898...	2020y06m10d	PM 4h00m	1
37.638...	127.0730...	2020y06m10d	PM 8h00m	2

(Figure 5) Example of DB Data



```
{
  userDate: '2020년 06월 17일',
  userLatitude: 37.626785,
  userLongitude: 127.08577,
  userSerialNumber: 2,
  userTime: 'PM 10시 30분'
}
```

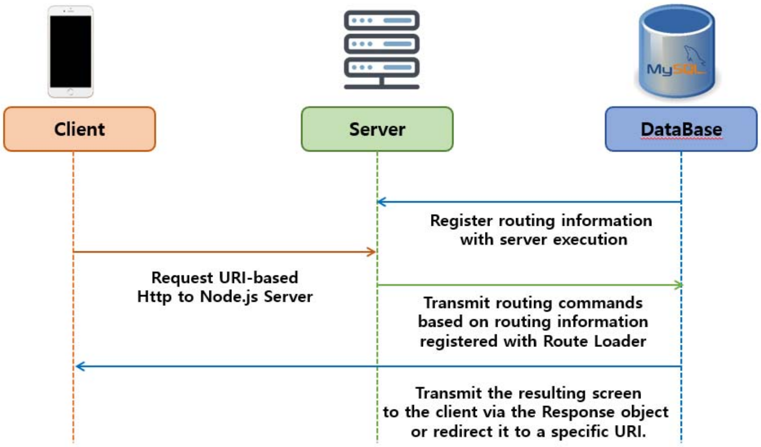
(Figure 6) JSON Format Data Displayed in the Console

Basically, the routing module branches the URI or HTTP request through the user's web browser, with the help of the "Express" module[5]. A sequence diagram of the process in which the client requests the server be routed in that structure is shown in Figure 7.

2.4 Prior Study of Algorithm

2.4.1 Dijkstra Algorithm

Dijkstra's algorithm is an algorithm that searches all neighboring paths from the starting point. Search by expanding the area based on the start point and checking whether it is the destination point. Repeat this process until the destination is found[6]. Dijkstra's algorithm is the most suitable in the real world because it can be used when all edges connecting vertices have positive values. Therefore, it is widely used in software using GPS. However, Dijkstra's


MySQL
Client
Server
DataBase
Register routing information with server execution
Request URI-based Http to Node.js Server
Transmit routing commands based on routing information registered with Route Loader
Transmit the resulting screen to the client via the Response object or redirect it to a specific URI.

(Figure 7) A Sequence Diagram of the Service Platform Process algorithm has the disadvantage that it does not work if the weights are negative. Dijkstra’s algorithm has a time complexity of O(n^2) unless a priority queue is used to check all edges and n means the number of vertices.

2.4.2 BFS

The BFS algorithm is one of the methods to search the entire graph by starting at the root node and searching the neighboring nodes first. It explores the node extensively, visiting vertices closer to the starting vertex first and later more distant vertices. In other words, we use an algorithm when we want to find the shortest or random path between two nodes. This algorithm guarantees the shortest path even if there are multiple paths, which is the answer to search the width first, and if there is a shortest path, it has the advantage of finding the shortest path even if one path is infinitely deep. However, there is also a disadvantage that it is unrealistic because it requires a lot of storage space and the number of nodes to be searched also increases as the number of nodes increases. BFS algorithm has time complexity of O(v^2) or O(v+e) when it has implemented by adjacency matrix or adjacency list, respectively. Also, v means the number of edges and e means the number of vertices.

2.4.3 A*

The A* algorithm finds the shortest route from the origin to the destination. This algorithm is characterized by using heuristics, that is, estimated weights. Estimated weights generally use the straight-line distance from the node currently being searched to the destination. The operation method is to first put the nodes adjacent to the starting point in the open list and find the evaluation function f. Here, f is the sum of g and h, where g is the actual weight between the current nodes from the source and h is the estimated weight between the current node and the destination. Among the nodes in the open list, the node with the smallest f, that is, the node estimated to be the shortest path, is removed from the open list and added to the close list. Then, using that node as a search criterion, repeat this process until the destination is reached. When the destination is reached, the shortest path can be found using parent node information The great advantage of this algorithm is that it is closest to an interpersonal space where information about almost every right is almost infinite. We recommend this method. The decision was made in the shortest possible way because it responds flexibly. The time complexity of A* depends first of all on the heuristic. If the search space is a tree, there is a single goal state, and the heuristic function h satisfies the following condition, then the time complexity is the |h(x) - h^*(x)| = O(\log h^*(x)).h* is the exact cost of reaching the goal at x in the best heuristic. That is, the error of h does not grow faster than the logarithm of the perfect heuristic h*, which loops the actual distance from x to the target[7].

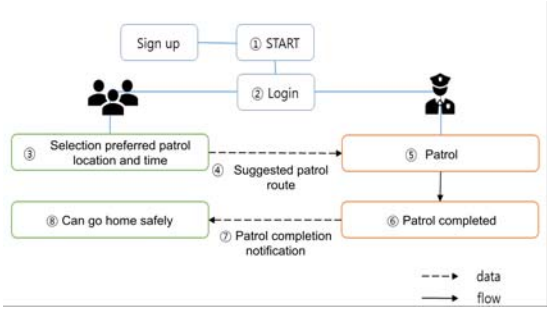
2.4.4 Floyd-Warshall Algorithm

The Floyd-Warshall algorithm is an algorithm that can find all the shortest distances from every vertex to every vertex in a graph without negative cycles. It doesn’t matter whether there are edges with negative weights as long as there are no negative cycles in the graph. That is, the corresponding algorithm is an algorithm using a dynamic programming technique, and the minimum cost between each node is calculated using an adjacency matrix. An algorithm that progresses while updating the minimum cost step by step from every node to every other node. The time complexity of the algorithm is O(n^3). This algorithm has the advantage of being concise in the source and finding the distance of the shortest path between all vertices, but it also has the disadvantage of being very time consuming. Floyd-Warshall algorithm has a time complexity of O(v^3) and v means the number of vertices.

3. Implementation of Service Platform

3.1 Principle of Operation by User

The overall flow chart of the proposed service platform is shown in Figure 8. In Figure 8, the overall service flow is indicated by a solid line. And the data flow in the platform is indicated by dotted lines. The main functions of the service platform proposed in this paper are the patrol recommendation route presentation and patrol completion notification function. The recommended route is calculated and presented through an algorithm in the Node.js server, and the patrol completion notification function also transmits data in real time using socket.io of Node.js[8]. The detailed system flow in Figure 8 is shown in Table 1.



(Figure 8) System Flow Chart

(Table 1) The Flow of the Bidirectional Patrol Route Service Platform

Order	Content
①	Users (citizen/police) can use the service platform after registering as a member
②	Each user accesses the application through a login method.
③	Citizens choose where and when they wish to be patrolled in the service platform.
④	Integrate the place and time requested by the citizen into the service platform and present the patrol recommendation route to the police
⑤	Patrol is proceeded according to the recommended route presented in ④
⑥	Complete the patrol
⑦	A notification is sent to the service platform that the patrol has been completed only for citizens who correspond to the route presented in ④.
⑧	According to ⑦, citizens can go home safely

3.2 Implementation of Function for Citizen

3.2.1 Application for Patrol Location and Time

The function to request a desired patrol location and time is the most essential in the proposed platform. Based on the existing 'Sunchal-Sinmungo' function, citizens select the place and time of day they want the police to patrol in the application. The UI of the place selection and application screen was configured using the Google Map API. When the citizen touch the desired patrol location on the map, a dialog box appears where the citizen can select the desired date and time for the patrol. This can be implemented with Android's Date Picker and Time Picker, respectively Location, date, and time data selected by citizens are stored in the DB through the server. After that, when the police search for patrol hope data applied by citizens, it is exposed to the police application.

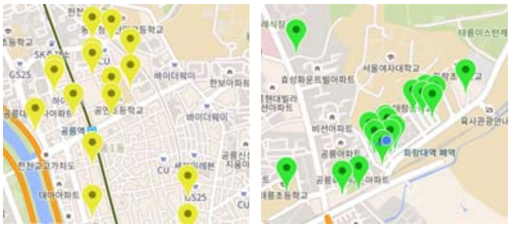
3.2.2 Confirmation of Patrol Completion in Real-time

The existing "Sunchal-Sinmungo" had a problem in that it was difficult for the user to know whether the patrol was completed in real time. We thought that if citizens could check the end of the patrol in real time, they would be able to go home with more peace of mind. After completing the patrol based on the patrol hope data applied by the citizens, the police press the 'patrol completed' button in the police application. When the patrol completion data is sent to the server, the server immediately sends a message to the citizen application that the patrol is complete. Citizens in this category are the citizen included in the patrol route. Also, they can receive real-time notifications and go home safely.

3.2.3 Checking about CCTV and Street Light Location

We thought that some citizens would want to go home along the road with CCTV or streetlights installed. Therefore, we decided to add a function that informs the location of CCTV and street lights. In order to know the location information of CCTV and street lights, APIs provided by each local police agency were used. Each latitude and longitude data included in the API was firstly purified and then stored in the DB. When citizens press the CCTV checking button or street light checking button, each latitude and longitude data is checked and the CCTV abd street lights corresponding to the area are displayed on the map respectively. The screen where you can check the information is shown in Figure 9. The left side of the figure

7 shows the CCTV, and the right side shows the location of the street lamp.

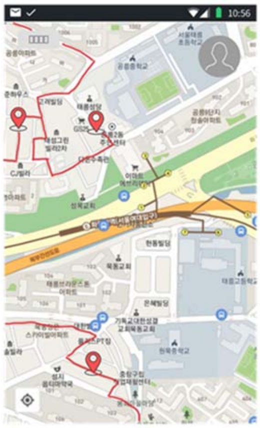


(Figure 9) CCTV/Street Light Location Confirmation Screen

3.3 Implementation of Function for Police

3.3.1 Confirmation of Requested Patrol Location and Recommended Patrol Route

Previously, all methods of police selection of patrol routes were done manually. Therefore, the work efficiency may be lowered. To solve this problem, the platform proposed in this paper recommends a patrol route to the police. The screen that recommends a patrol route is shown in Figure 10. The screen used the T-MAP API, not the Google Maps API. As previously explained, this patrol route is calculated based on the desired location and time applied by the citizens. When the calculated patrol route is recommended to the police, the police perform patrol tasks by adding another patrol location to this route. When the patrol task is over, police press the patrol complete button to notify the citizens included in the recommended patrol route. This allows citizens to return home safely and the police to patrol efficiently.



(Figure 10) Route Guidance Screen

3.3.2 Complaints about Abnormal CCTV and Street Lights

When citizens return home, they often get help from CCTV or street lights. Therefore, it is essential to check that there are no abnormalities in CCTV and street lights. However, as we pass by the road, there are often more CCTVs and street lights that has abnormalities than you think. Since the police patrol several times a day, there are many opportunities to check CCTV and street lights for abnormalities. Therefore if the police find abnormalities in CCTV or street lights, they can file a complaint with local ward office. This is directly related to the safe return for citizens. In addition, it is expected that complaints can be handled more quickly then when a large number of individuals file complaints respectively through the linkage between these public institutions.

3.4 Implementation of Patrol Route Recommendation Algorithm

The service platform proposed in this paper has the characteristic that all citizens have to go through all the places requested for patrol by time. Therefore, among the algorithms mentioned in previous studies, the service platform and suitable algorithm we propose can be seen as the Floyd-Warshall Algorithm. Also, in terms of time complexity, the Floyd-Warshall Algorithm is superior even under the condition that the factors to be weighted such as the shortest distance, convenient route, and real-time traffic information are the same. Breaking it down a little bit by case, if it takes longer to go straight to the next node than if it went through the current one, the current path is maintained and followed. However, in the opposite case, the shortest distance between each node is selected. If the node through which it passes is not included, the shortest distance is selected among the included distances and the number of times is calculated. Finally, we propose that the total time complexity can be obtained by multiplying the individual time complexity by the number of times previously calculated.

4. Conclusion

In this paper, we propose a service platform that recommends the optimal shortest distance as a patrol route. The patrol route is calculated through an algorithm by reflecting the location and time that citizens directly apply through this service platform. This is referred to when the police carry out patrol tasks. After that, citizens can check whether or not the police have completed their patrol tasks through the platform in real time. In addition, as an additional function, location data of CCTV and street lights can be checked in this service platform. This service platform was designed and implemented based on the MVC pattern. First, the Model needs a table to store not only basic information about citizens and police, who are the subject of the platform, but also the places where citizens want to patrol. In addition, the tables are also needed to store information including latitude and longitude to show CCTV and streetlights, respectively. It is suggested to use MySQL as a DBMS for flexibility as well as relation between the tables. Second, in View, this service platform composes the screen using xml based on Android application. The screens are divided into citizens and police, and are displayed according to their respective functions. Lastly, in the Controller, since the patrol process proceeds in real time, Node.js, an asynchronous I/O-based architecture suitable as a real-time server, is used. This enables immediate requests and responses between citizens and the police. Also, Node.js acts as a mediator that connects the DB and Android, and the express module is used in whole process. And for the optimal patrol route, which is the core of the service platform, we use the floyd-warshall algorithm among many route finding algorithms. It allows all the places that was requested to be patrolled by citizens. Through this study, it is expected that the citizen will be able to return home safety, then the crime rate will be decrease.

Reference

[1] Seong-jin Park and Seong-hwi Kang, “The biggest concern in the 60s is ’job’... A woman in her 20s is ’CCTV’”, DongA.com, 2020. https://www.donga.com/news/article/all/20200219/99768649/1

[2] Yu-min Jo, Ye-jin Jang and Jun Hwang, “Design and Development of Two-way Communication Patrol Route Recommendation Service System”, proceeding of KSII Fall Conference 2020, Vol.21 No.2, 2020.

[3] Hajeong Yoo, Yumin Jo, Yejin Jang, Jongho Paik, “Design of Personalized Creative Works Recommendation Service Platform based on Five-Senses”, proceeding of KSII Spring Conference 2020, Vol.21 No.1, 2020.

[4] Seokho Lee, “DataBase Concepts”, Jungiksa, 2010.

[5] SongYeon Lee and JongHo Paik, “An Asynchronous-Driven Node.js Based Intermediary-free Direct Deal Distribution Platform Converged with Cloud Service,” TIIS, VOL. 13, NO. 8, pp.4212-4226, 2019. https://doi.org/10.3837/tiis.2019.08.022

[6] Yong-Hu Lee and Sang-Woon Kim, “A Hybrid Search Method of A* and Dijkstra Algorithms to Find Minimal Path Lengths for Navigation Route Planning”, Journal of the Institute of Electronics and Information Engineers, Vol 51, No.10, pp109-117, 2014. https://doi.org/10.5573/ieie.2014.51.10.109

[7] Cho, Sung Hyun, “A Pathfinding Algorithm Using Path Information”, Proceedings of the Korea Game Society, 13(1), 31-40, 2013. https://doi.org/10.7583/JKGS.2013.13.1.31

[8] Jae-Gon Jung, “Do it! Node.js Programming”, Easy-Publishing, 2018.

◐ 저 자 소 개 ◑

Yu-min Jo

2021 B.S in Mathematics from Seoul Women's University
2021 B.S in Software Convergence from Seoul Women's University
2021~present M.S in Software Fusion from Seoul Women's University
Research Interest : Database, Algorithm, Machine learning, BigData, Intelligence Platform, Recommendation System
E-mail : rosemin97@swu.ac.kr

Ye-jin Jang

2021 B.S in Software Convergence from Seoul Women's University
2021~present M.S in Software Fusion from Seoul Women's University
Research Interest : Server, Database, Intelligence Platform, Android Application, Programming
E-mail : yuvm02@swu.ac.kr

Jong-ho Paik

1994 B.S in Electrical and Electronic Engineering from Chung-Ang University, Seoul, Korea
1997 M.S in Electrical and Electronic Engineering from Chung-Ang University, Seoul, Korea
2007 Ph.D in Electrical and Electronic Engineering from Chung-Ang University, Seoul, Korea
1997~2011 Director, Mobile Terminal Convergence Research Center, Korea
2011~present Professor in Software Convergence, Seoul Women's University
Research Interest : web-based communication, intelligence platform, machine learning, software testing, wireless/wired communications system design, video communications system design and system architecture for realizing advanced digital communications system and for advanced mobile broadcasting networks
E-mail : paikjh@swu.ac.kr