

# IMPLEMENTATION OF RESTFUL-API BASED BI-DIRECTIONAL PROCESS MANAGEMENT IMPROVEMENT SYSTEM USING GCM

<sup>1</sup>JOON-YONG KIM, <sup>2</sup>KOO-RACK PARK\*, <sup>3</sup>YOUNG-SUK CHUNG, <sup>4</sup>JINYOUNG JUNG

<sup>1,2,3</sup>Dept. of Computer Science & Engineering, Kongju National University, Chungnam Cheonan Subuk Cheonan-Dearo 1223-24(Budeadong 275), 31080, South Korea

<sup>4</sup>Dept. of Bio Information, Daejeon Health Science College, Daejeon Dong-gu Chungjeong-ro 21, 34504, South Korea

E-mail: <sup>1</sup>musim-kum@hanmail.net, <sup>2</sup>ecgrpark@kongju.ac.kr, <sup>3</sup>Merope@kongju.ac.kr, <sup>4</sup>jyjung@hit.ac.kr

## ABSTRACT

In order to prevent the loss due to the delay of the material input during the process management problems frequently occurring at the construction site, we have designed and implemented a system that can prevent the loss due to the delay of material input occurring due to process management simultaneously informing the head office and site staff of the timing of material input, by using the GCM Push Notification service of Google, messages sent to the database are linked to the database and user app by RESTful-API method.

**Keywords:** *Process Management, ERD, Push Alarm, GCM, RESTful-API*

## 1. INTRODUCTION

The present society, thanks to the remarkable development of the IT industry, is moving our life to a higher level through the 4<sup>th</sup> industrial revolution and artificial intelligence technology. In order to build the infrastructure of these technologies, especially the construction industry, the machinery industry, the power generation industry, and so on, the social infrastructure industry which developed our society by supporting the 20th century has fused with the IT technology and achieved a lot of development.

However, the construction industry, which had been the initial role model for the IT industry in the 21st century, has been not only slowed down due to the ideology of the companies that are leading the market economy principle, but also has become a social issue due to its frequent accidents [1].

In order to solve these problems, the government and research institutes have conducted various studies and concluded that proper process management should be the first priority as a solution [2]. Determining the timing of input of materials, equipment, and personnel that form the basis of this process management is the key of various techniques for controlling the

process.

To effectively manage process, both the head office and the field worker must share the timing of the inputting the elements as required for the process in real time and carry out the project, but most of the systems are currently dominated by one-way processes. In order to solve these problems, we propose a bi-directional process system for management.

In this paper, to implement the "RESTful API based construction site process management improvement model" [3] proposed in the existing research, we established an ERD concept process management database, and have designed and implemented a system that can inform the head office and site staff sending messages by using the GCM Push Notification service of Google, linked to the database and user app by RESTful-API method.

The system proposed in the existing research was effectively implemented, and it will be possible to prevent the problems that may occur in the process by giving push alarm without incurring an additional cost, thereby positively affecting the enterprise and society.

## 2. RELATED STUDIES

### 2.1 Process Management

Process management is a technique for a series of work processes that take place in an industrial field. In particular, in the case of construction site, it is a technique to identify each process to complete the project within the predetermined construction period so that it can be performed as planned before the project execution.

Process management techniques include CPM(Critical Path Method), PERT(Program Evaluation and Review Technique), LOM(Line Of Balance), Bar-Chart and S-Curve. CPM and PERT are applied to large-scale constructions and it needs program construction cost and expert managers. On the other hand, Bar-Chart or S-Curve is a technique that can manage the overall process using Excel, etc., without any additional skill acquisition. It is mainly used in small-scale construction sites, and is also frequently used by big companies as well [1].

In this paper, a combination of Bar-Chart and S-Curve has been studied focusing on the most commonly used process management techniques in the field.

### 2.2 ERD (Entity Relationship Diagram)

In order to construct a database system, it is necessary to analyze and apply mutual relations among the data to be used. ERD makes the relationship clear to the developer [4].

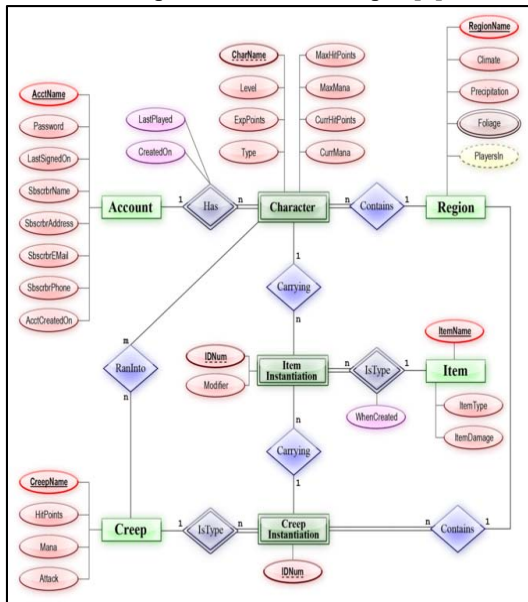


Figure 1: ERD Using Chen's Notation

ERD is widely used since it was originally proposed by P. P. Chen in 1976 as a

basic concept of relational database design [5] [6] [7].

The main content of the ERD is to design the database by constructing an object-relational model by identifying the relationships among the entities. The ERD's representation techniques include P. Chen's Notation, Crow's Foot Notation, Rein85 Notation, and IDEF1X notation [8]. Even though it has continued to evolve to object-oriented analysis design method, it is difficult to use it if one lacks technical understanding about object-oriented method [9].

The figure 1 and figure 2 are conceptual diagrams of the representative ERD technique: P. Chen's Notation and Crow's Foot Notation [10] [11].

Chen's Notation is mainly used in database design, and it designs databases with three elements: Entity, Relationship, and Attribute. It is used to determine the relationship between Entity and Entity, and to construct a relational database.

The Entity is represented by a rectangle, the Relationship by a diamond, and the Attribute draws a circle connecting a line to Entity or Relationship.

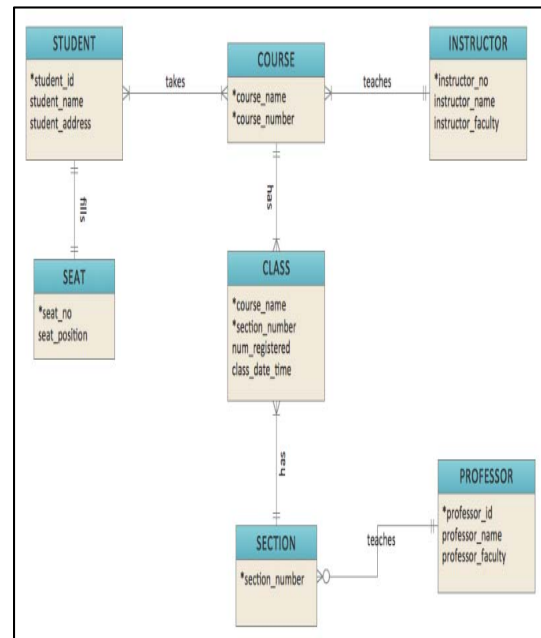


Figure 2: ERD Using Crow's Foot Notation

Crow's Foot Notation is also referred to as Information Engineering Notation. The basic concept is the same as Chen's Notation. For more explicit effects, Entity is represented by a

rectangle, Relationship is represented by a line connecting Entity, and Attribute is written in Entity.

In this paper, we designed the database using SQL (Structured Query Language) with Crow’s Foot Notation as the basic concept in order to secure the scalability of future research.

**2.3 Push Alarm Service**

Unlike existing SMS(Short Message Service) or MMS (Multimedia Message Service), the push alarm service can be used free of charge, and it is a service technology that can access any application using a server without user's request [12].

There are push services to announce messages, emergencies or advertisements which include GCM(Google Cloud Massaging), APNS(Apple Push Notification Service), AOM(Always On Management) of SKT, and

Apps	Features
GCM	<ul style="list-style-type: none"> <li>● Provide Cloud Server</li> <li>● Less likely to have traffic (speedy)</li> <li>● Free of charge</li> </ul>
APNCS	<ul style="list-style-type: none"> <li>● Apple's dedicated platform for mobile services</li> <li>● Easy to develop iphon app since iOS does not support background process,</li> <li>● One-way push method</li> </ul>
AOM	<ul style="list-style-type: none"> <li>● Troubleshoot traffic issues</li> <li>● Requires a separate authentication server</li> <li>● Personal information at risk</li> </ul>
nPush	<ul style="list-style-type: none"> <li>● NHN's integrated mobile push service</li> <li>● Requires a separate server</li> <li>● Easy to process large-size data</li> </ul>

nPush (NHN Mobile Push) [13].

Table 1 below compares the features of these push notification services [14].

Table 1: Push Alarm Service types and features

**2.3.1 GCM(Google Cloud Massaging)**

In this paper, we adopt GCM method to implement numerical data push service in cloud environment based on Android

GCM is developed for users who use Android operating system in Google, so that they can easily provide push service for free [15]. GCM allows using Google server through a separate API that enables developing a push

service that can send messages to individual devices.

Developed applications receive a unique ID that allows them to use a value that identifies the device using this application on a Google server without constructing a separate server [12] [16]. The following figure 3 shows the GCM push service process [12].

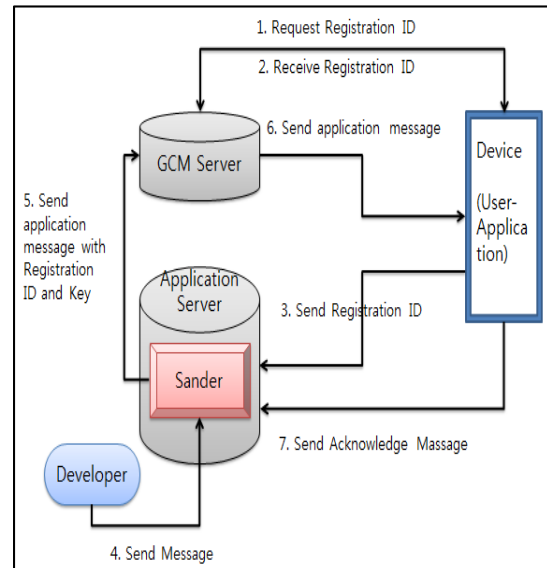


Figure 3: GCM Service Process

First, it requests the register ID and Key value to receive the message to Google GCM server. Second, it sends the generated ID to the GCM server and to the user device. Third, the user application registers the register ID issued by the GCM server in the message sender. Fourth, the developer or administrator sends a message to the application server. Fifth, the register ID and the message sent by the developer to the application server are sent to the GCM server. Sixth, the GCM server senses activation of a certain period predetermined or activation of application and sends a message to the device that matches the register ID. Seventh, when a message is sent from the GCM to an individual device, the user application sends a notification message to the user and transmits an acknowledgment message to the application server that the message has been delivered.

**2.4 RESTful-API(Representational State Transfer-Application Programming Interface)**

The RESTful-API is a technology required for structural linkage between the resources of the database installed in the local and the application, and the RESTful-API of JSON

method is widely used as a standard [17]. The RESTful-API is a web service standard based on ROA (Resource Oriented Architecture). It is a technology that converts a resource on the Web to a URL (Uniform Resource Identifier) and accesses the URL using HTTP (Hypertext Transfer Protocol). It is a service that directly transfers requested resources to a user's app based on ROA [18] [19]. The following figure 4 is the architecture of the RESTful API [20].

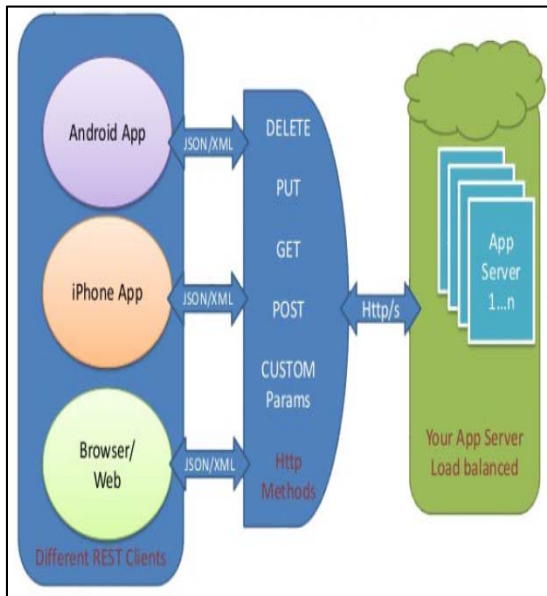


Figure 4: RESTful API architecture

### 3. EXPERIMENTS AND DISCUSSION

In this paper, in order to implement the "RESTful-API-based process management improvement system" proposed in the existing research, ERD technique was used to construct the database using the process of installing the bridge in the civil engineering construction site.

We connected this to sender server of Google, and GCM server which provide push service to notify user device and configure a series of processes to check messages stored in user's database device through RESTful API.

The following figure 5 is the overall system proposed and to be implemented in this study.

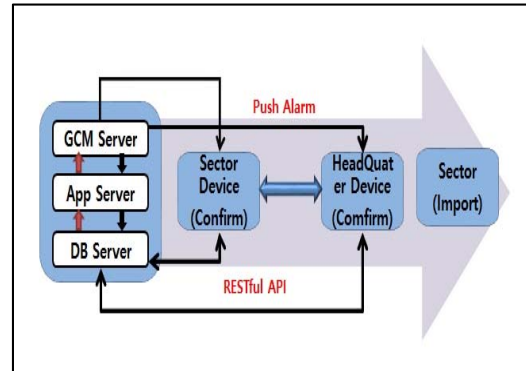


Figure 5: Implemented system Process

First, Construct ERD using SQL on database server.

Second, the GCM server receives the device register ID and the key value of the field user and the head office personnel.

Third, register the register ID issued from the GCM server at the sender in the App server.

Fourth, the administrator sends an alarm message prepared to the App server.

Fifth, the message and the register ID delivered to the App server are transmitted to the GCM server.

Sixth, find the same device as the register ID registered in GCM server and transmit the message registered

Seventh, connect server and application of database by RESTful-API method

Eighth, check the message stored in the database and confirm by connecting with the device.

Ninth, check and process the alarm messages transmitted to the field users and the head office personnel

#### 3.1 Process Table and ERD (Entity Relational Diagram)

In this paper, the process table of the construction site used in this paper is extracted from the pier installation process which is part of the 00 bridge construction process project. And we extracted the timing and quantity of the concrete materials, concrete, re-bar, and form, which are essential materials for the process.

The material input timing is configured of a system that gives a message notification 2 days before concrete, 5 days before Re-bar, and 10 days before form. The following figure 6 is the process table for system implementation.

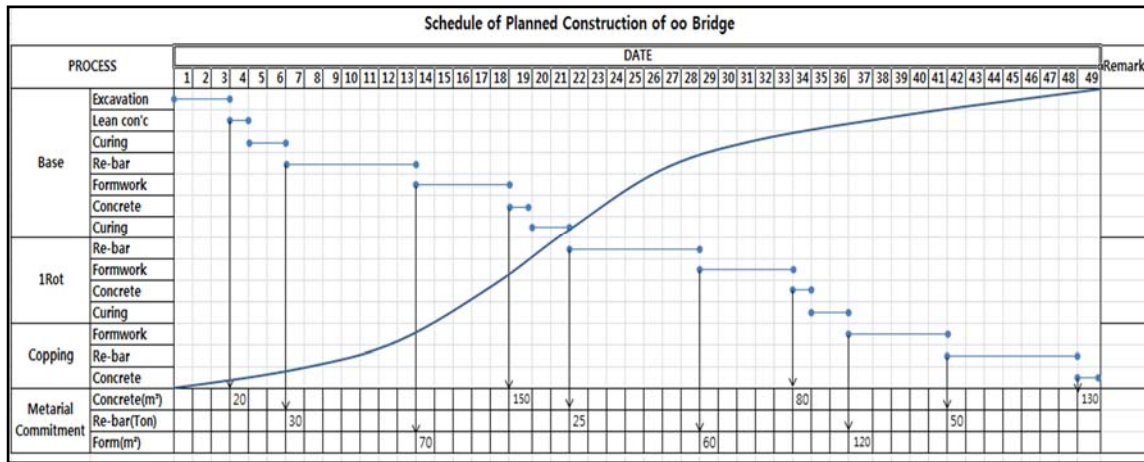


Figure 6: Process table

The ERD for constructing the database using the created process table was constructed as shown in the figure 7 below.

Database Design		Step	design	Activity	Database design		
		Writer		Reviewer			
Doc No		Date	2017-10-11	Ver	1.0		
Database	ALRAM_MASTER	Table	ALRAM_SETTING				
Table Space		Entity					
Function							
NO.	Column	Property	Key	Null	Type	Len	Remark
1	Item Code			NO	int		
2	Number of alarms			YES	int		
3	Advance alarm			YES	int		
4	Sector code			YES	varchar	50	
5	Device ID			YES	varchar	250	
6	Creation Date			YES	datetime		
7	Change Date			YES	datetime		
8	Use or Not			YES	varchar	1	
9	Alarm Message			YES	varchar	60	
10							
11							
12							

Figure 7: ERD blue print

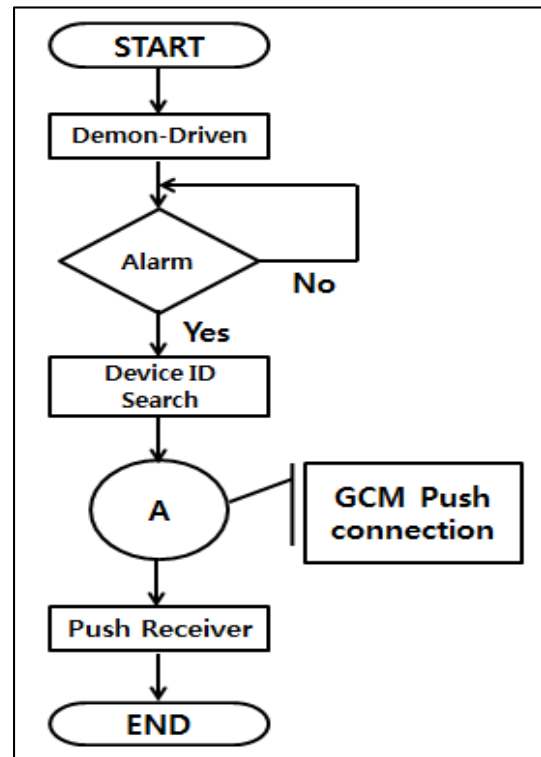


Figure 8: GCM flow chart

First, run the demon check until message notification is confirmed  
 Second, if there is a message, find the device ID and start the push service by linking with GCM.  
 Third, the message is received alarm\_manager, the demonstration source for running this system, is shown in figure 9 below.

### 3.2 GCM (Google Cloud Messaging) Operation

The following figure 8 is a flowchart of the GCM Push service applied in this paper.



```

public static void alram_manager()
{
    Connection con = null;
    con = DriverManager.getConnection("jdbc:mysql:
        //localhost","root", "1234");
    java.sql.Statement st = null;
    ResultSet rs = null;
    st = con.createStatement();
    rs = st.executeQuery("select * from alram_master
        where use_yn='Y' and TO_DAYS(curdate()) >=
        TO_DAYS(curdate()-before_alam) ");
    Calendar oCalendar = Calendar.getInstance( );
    while (rs.next()) {
        String device_id = rs.getString("device_id");
        String alram_msg = rs.getString("alram_msg");
        int before_alam = rs.getInt("before_alam");
        if( oCalendar.get(Calendar.DAY_OF_MONTH)
            - before_alam == 0 )
            {
                send_alam(device_id,alram_msg);
            }
    }
}
}

```

Figure 9: *alarm\_manager Demon Source*

The demon source shown in figure 9 is implemented with JAVA and is a module that sends push notifications to users who need to receive alarms automatically executed by the server according to schedule.

As an SQL condition, first, only the current users who are going to send alarms are able to read

Second, if the data condition is the date of which has to notify of an alarm, the message is delivered to the GCM server.

The module that delivers the message to the user's app is implemented as shown in figure 10 below.

```

public static void send_alam(String device_id,
                            String alram_msg)
{
    //google gcm shipped
    Sender sender = new Sender("ekghdk.....");
}

```

```

String regId = device_id;

    Message message = new Message.Builder()
        .addData("msg", alram_msg).build();

    List<String> list = new ArrayList<String>();

    list.add(regId);

    MulticastResult multiResult;

    multiResult = sender.send(message, list, 5);

    if (multiResult != null) {

        List<Result> resultList = multiResult.
            getResults();

        for (Result result : resultList) {

            System.out.println(result.
                getMessageId());

        }

    }
}
}

```

Figure 10: *send\_alam Module Source*

The source of figure 10 is a module that allows Google to send messages to users' app using GCM. The sender is a server key license received from Google, and is device\_id, in other words, regid is the ID of the terminal. In addition, alram\_msg received from the alarm\_manager is configured to send to each individual via the Message method.

### 3.3 Implementation results

In order to send the final message to the user's device, the setting flow of the manager app is designed as shown in figure 11 below.

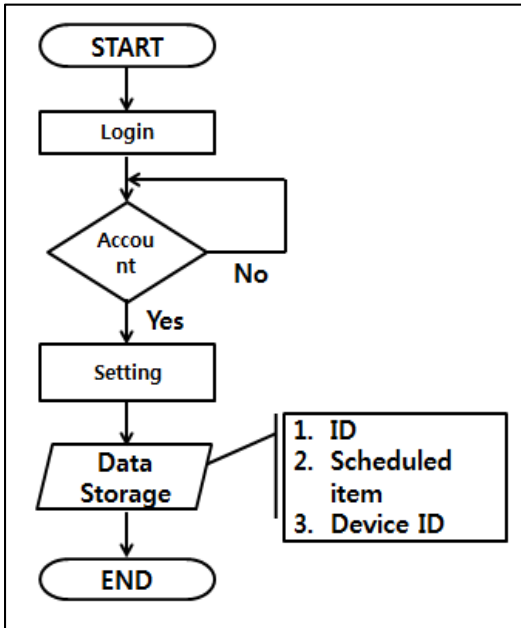


Figure 11: Manager App setting flow chart

First, log in and verify your account.

Second, via the setting process, save the account ID, the scheduled item, and the device ID.

Third, when the update and insert are completed, the app sends a save completion notification.

The source for driving the designed flow is shown in figure 12 below.

```

public static void alam_manager()
{
    rs = st.executeQuery("select item_cd from
        alam_master where device_id='
        "+device_id+" and item_cd='"+item_cd+" '");
    if (rs.next())
    {
        String Query = "UPDATE alam_master set
            item_cd
            ='"+item_cd+"', alam_qty=
            "+alam_qty+", before_alam=
            "+before_alam+",alam_msg=
  
```

```

        "+alam_msg+",
        use_yn='"+use_yn+"' where device_id=
        '"+device_id+"";
        Statement stmt = con.createStatement();
        rowCount = stmt.executeUpdate(strQuery);
        //number of update
    }
else
{
    String Query = "INSERT INTO alam_master
        (item_cd,device_id,alam_qty,.....";
        Statement stmt = conn.createStatement();
        stmt.executeUpdate(strQuery);
    }
    request_msg("saved");
}
  
```

Figure 12: Administration Manager Source

The figure 12 shows a procedure in which an administrator sets and saves Push notification for each user. First, check whether existing data is registered through st.execute Query.

Second, if the item data for corresponding user already exists, update the data, if it does not exist, the app will send a save completion notification when the procedure to register the data is updated.

The figure 13 below shows the json source, which is a RESTful-API that connects the manager app to the database and implements the source in a json way.

If you click on the per-user preference, the data that was previously set is imported into the app, the necessary information is modified, and the message is sent to the person in charge.

```

public static String user_setting(String corp_cd, String
        item_cd, String user_device) {

    db_con();

    JSONObject jsonObject = new JSONObject();

    JSONArray personArray = new JSONArray();

    JSONObject personInfo = new JSONObject();

    try{

        String qu="SELECT * FROM alram_master where

            corp_cd='"+corp_cd+"' and device_id=

            '"+user_device+"'and item_cd='"+item_cd+"' ";

        ResultSet rs = statement.executeQuery(qu);

        while ( rs.next() ) {

            personInfo = new JSONObject();

            personInfo.put("item_cd", rs.getString("item_cd"));

            personInfo.put("alram_qty",rs.getString("alram_qty"));

            personInfo.put("before_alram",rs.getString

                ("before_alram"));

            personInfo.put("corp_cd", rs.getString("corp_cd"));

            personInfo.put("device_id", rs.getString

                ("device_id"));

            personInfo.put("use_yn", rs.getString("use_yn"));

            personInfo.put("msg",rs.getString("msg"));

            personArray.add(personInfo);

            jsonObject.put("DATA", personArray);

        }

    }catch (Exception e) {

    }

    return jsonObject.toString();

}
    
```

Figure 13: Manager app JSON Source

The following figure 14 is the result of implementing the system and is the administrator environment setting screen.

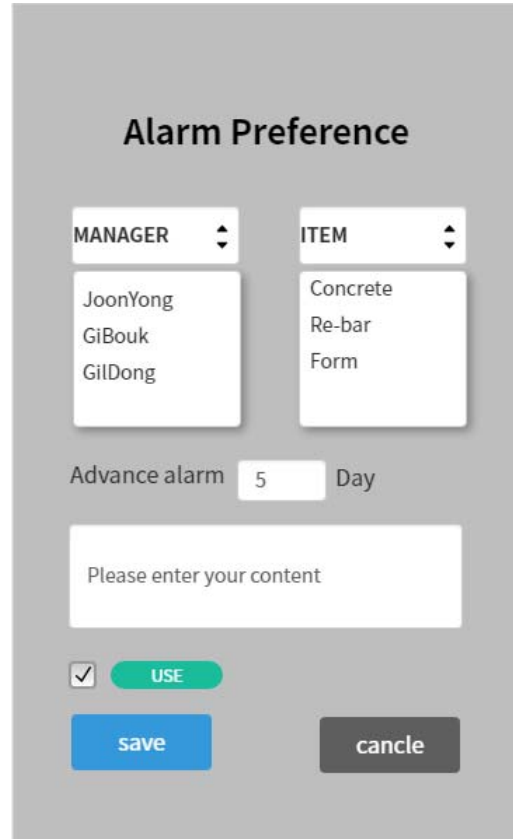


Figure 14: Administrator environment setting app display

Select the registered user and material through the MANAGER and ITEM window, select the alarm scheduled date, and enter the message to be transmitted to the user device in the input window.

The following figure 15 shows the final message received from the user app.



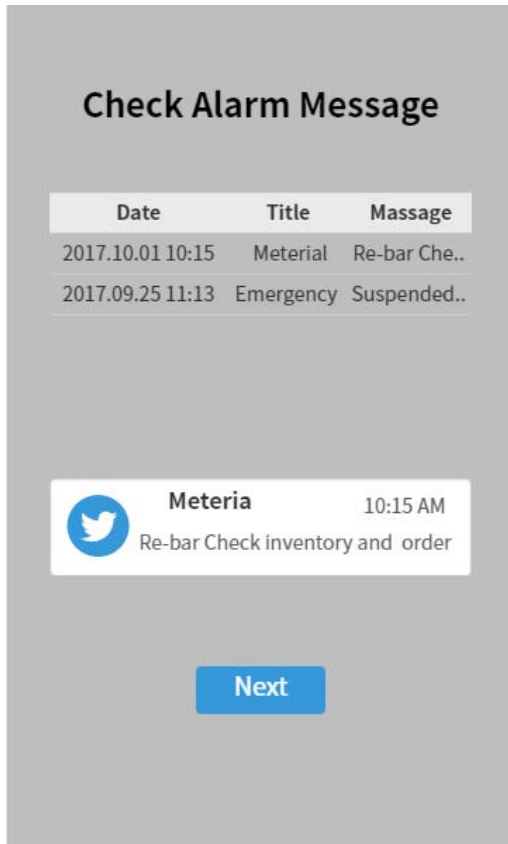


Figure 15: User's App display

### 3.4 Discussion

The management system of material injection time, which is an important part of the process management system of the construction site, is proceeding as a one-way process to request the head office for the materials and input timing required at the site. Such a system suffers from the problem that it is possible to delay the construction period due to a mistake such as missed a request for input of materials at the site. In this paper, I have implemented a system that notifies the head office and the site manager of the timing of material input so that these problems can be prevented in advance.

The main difference between the existing system and the improvement system is that the existing uni-directional system is improved to the bi-directional system to solve the problems of the existing system. In addition, by informing the head office and the field person at the same time, the process of requesting purchase of materials from the site to the head office is shortened, thereby shortening the construction period.

The following figure 16 and figure 17

show the process flow of the existing system and the improvement system[3].

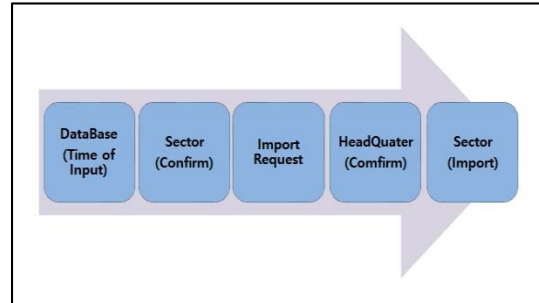


Figure 16: existing system process

The figure 16 above shows a one-way system that proceeds according to the flow of time through the process of the existing system. As shown in the figure, if you request a material purchase at the site, the head office approves it and puts the material into the site.

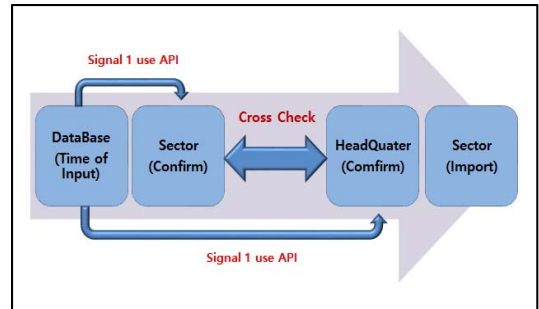


Figure 17: improvement system process

figure 17 above shows a system that can prevent the loss caused by not knowing the timing of input, reducing the process of requesting the head office to notify the signal to the head office and the field person at the same time when inputting the material input into the database.

In order to verify the effectiveness of the system implemented in this paper, the effect of the improvement system on the headquarters and field officers of the small and medium construction companies was verified and the results as shown in table 2 were obtained.

Table 2: Improvement system effect evaluation table

Div	Survey Contents			Effect rate(%)
	Effective	Ineffective	Abandon	
H/O manager	93	3	4	93%
Field manager	96	2	2	96%
sum	189	5	6	94.5%

In addition, the result of checking the cost reduction is as shown in figure 18

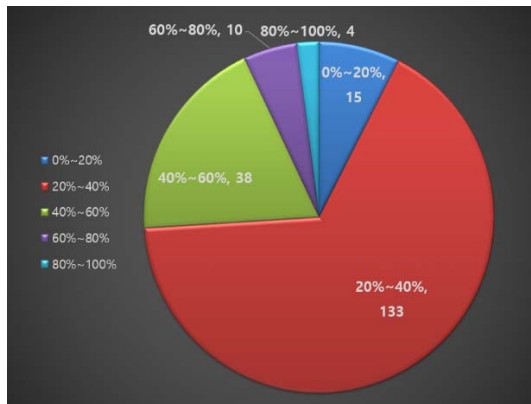


Figure 18: cost reduction rate diagram

As shown in Table 2 and Fig. 18 above, it is proved that the improvement effect is 94.5% and the cost saving effect is close to 40%.

#### 4. CONCLUSION

In this paper, we design and implement the most important process management, especially the problem of delay in material input, among various factors known as problems in the construction site.

The implementation system constructed the material input timing process management database based on the ERD concept. We use a system that links database and user app with the RESTful method by relaying alarm messages to the database using GCM, a push alarm service of Google.

As a result of implementation, we could confirm that the same message was sent to both the head office and the field officer who was registered in advance. Through this, it is confirmed that it is possible to cross check at the head office and in the field, thereby preventing the

loss in process due to the delay of the material input.

Future research should be conducted on systems that can be applied to the overall construction process in conjunction with ERP systems that manage equipment, personnel, and other elements throughout the enterprise as well as materials that occur in the processes.

#### REFERENCES:

- [1] Hong, Seongtae. "The Collapse of Sampoong Department Store and Corruption-Accident Society." *ECONOMY AND SOCIETY*, 108 (2015.12): 231-253.
- [2] Lee Kyu-Sun, Park Hee-Sung. "Implementation of Time Management : Based on Road Construction Projects." *JOURNAL OF THE KOREAN SOCIETY OF CIVIL ENGINEERS D*, 31.3D (2011.5): 451-456.
- [3] J.Y Kim, K.R Park, Y.S Chung, J.Y Jung, "A Study on the Improvement Model of RESTful API-Based Construction Site Process Control." *The 2<sup>nd</sup> ICCCA (2017)*, Saipan, UAS.
- [4] Chen, Peter Pin-Shan. "The entity-relationship model—toward a unified view of data." *ACM Transactions on Database Systems (TODS)* 1.1 (1976): 9-36.
- [5] Chen, Peter Pin-Shan. "English sentence structure and entity-relationship diagrams." *Information Sciences* 29.2 (1983): 127-149.
- [6] Al-Masree, Hala Khaled. "Extracting Entity Relationship Diagram (ERD) from relational database schema." *International Journal of Database Theory and Application* 8.3 (2015): 15-26.
- [7] D. Yeh, Y. Li, and W. Chu, "Extracting entity-relationship diagram from a table-based legacy database", *Journal of Systems and Software*, vol. 81, no. 5, (2008), pp. 764-771.
- [8] Won-Kyu Lee, Se-Myoung Park. "A Study on the Analysis Method of the Basic Design Process of a FPSO Crane using ERD." *Journal of the Korean Society of Manufacturing Process Engineers*, 14.4 (2015.8): 108-117.1.
- [9] Watanabe Kouzo et al. "The Software design technology." *Sung An Dang*, 2004, pp.24-69
- [10] ERD Using Chen's Notation. [http://en.wikipedia.org/wiki/File:ER\\_Diagram\\_MMORPG.png](http://en.wikipedia.org/wiki/File:ER_Diagram_MMORPG.png), 16/10/2017.
- [11] ERD Using Crow's Foot Notation <http://www2.cs.uregina.ca/~bernatja/crowfoo>

- [t.html](#), 16/10/2017.
- [12] Hyun-Min Kang, Hyun-Su Choi, Kyung-Ae Cha. "Development of Vehicle Status Alerts System for Personal Information Leakage Protection using the NFC-based GCM Service." Journal of Korea Multimedia Society, 19.2 (2016.2): 317-324.
- [13] Young-Wook Cha, Jun-Ki Jeong, Choon-Hee Kim, Kyung-Gjun Kim. "Research of Event Notification Subsystem Based on Websocket and GCM Push Services." Journal of Korean Institute of Information Technology, 14.10 (2016.10): 123-131.
- [14] Push Service.  
<http://d2.naver.com/helloworld/1119>,  
07/10/2017.
- [15] Jung, Chang-Hun, Ye, Jun-Ho, Kim, Chul-Jin. "A Mobile Customization Technique using Push Service." Journal of Korea Academia-Industrial cooperation Society, 14.9 (2013.09): 4498-4506.
- [16] Young-Min Lee, Kyung-Rak Sohn. "Fabrication of smart alarm service system using a tiny flame detection sensor based on a Raspberry Pi." Journal of the Korean Society of Marine Engineering, 39.9 (2015.11): 953-958.
- [17] Yong-Ju Lee. "Resource Matchmaking for RESTful Web Services." Journal of Korean Institute of Information Technology, 11.8 (2013.8): 135-143.
- [18] Song, Byung-Kwen. "Performance Analysis of Web Service Middleware based on SOAP/RESTFUL." Journal of IKEEE, 18.1 (2014.03): 146-151.
- [19] Krislan B. Ong, Gi-Weon Kim, Woo-Yong Son, Jung-Gil Song. "A REST-based Information Retrieval in a Hybrid SaaS Architecture." Journal of Korean Institute of Information Technology, 7.3 (2009.6): 262-267.
- [20] RESTful-API Architecture  
<https://shareurcodes.com/blog/creating%20a%20simple%20rest%20api%20in%20php>  
02/10/2017.