Web-Accessible Maritime Vessel Equipment Smart Monitoring and Maintenance Platform Using MySQL Database and JSP

Stephen Ryan Angsanto, Gyuhyeon Gim, Sung-Phil Heo, Donguk Kwon, and Wansu Lim
Department of IT Convergence Engineering
Kumoh National Institute of Technology
61 Daehak-ro, Gumi, Gyeongbuk 39177, S. Korea
E-mail : {angsanto, gyuhyeon.gim, sungphil.heo, ehddnr0408, wansu.lim}@kumoh.ac.kr

Abstract: This paper presents a web-accessible database adapted for monitoring and maintaining equipment on-board most ships to engineers for easy monitoring and quick data retrieval. A module retrieves relevant and important data then transmits them wirelessly to a mobile device which sends them to an off-site database for more detailed analysis later. Designed mainly with MySQL 5.7 and JSP which offers better security in data access, low cost, compatibility, and accessibility with different operating systems and computer architectures than MS Access or FileMaker.

Keywords—**MySQL, Database, Smart monitoring.**

1. Introduction

Nowadays, the cost of launching and operating database systems have become lower and more frequently used in almost every field[1]. With the development of the next generation of maritime vessels, the complexity of maintaining and monitoring also increases. For that reason, we developed a module that retrieves data from an equipment and transmits them to a mobile device that then send them to an off-site database for further analysis later. So the idea involves a modular device that can be essentially plug-and-play into most equipment and store important data in a database that can be accessible from any place. In this paper, we implemented the database system based on MySQL with the InnoDB storage engine and JSP to render a web page that can be easily accessible for easy monitoring and data retrieval.

2. The Proposed System

2.1 System architecture

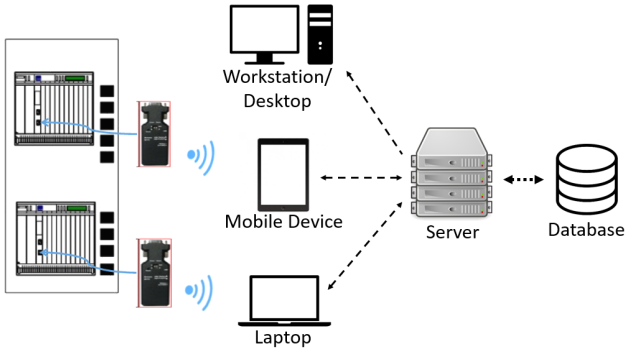

Figure 1. Overall System Design.

As figure 1 shows, a smart terminal dongle module is attached to an equipment. The module retrieve relevant and critical data from the equipment and transmits them to a mobile device, connected to the Internet, then stores the data collected to an off-site database. The design also takes into account the cost, flexibility, and portability similar to [2]. However, as we are passing data to JavaScript for processing on the client-side, we opted to use JSON over XML (PHP). To serve JSP pages to the client, we decided to install an Apache Tomcat Sever [3], to use MySQL 5.7 [4] as a relational database manager system and to use the JSP programming language for developing client-server applications and creating dynamic content for websites. Requests and updates from JSP or the Android application on the mobile device towards the MySQL are carried out by using Structured Query Language (SQL) and JSON.

2.2 System operation process

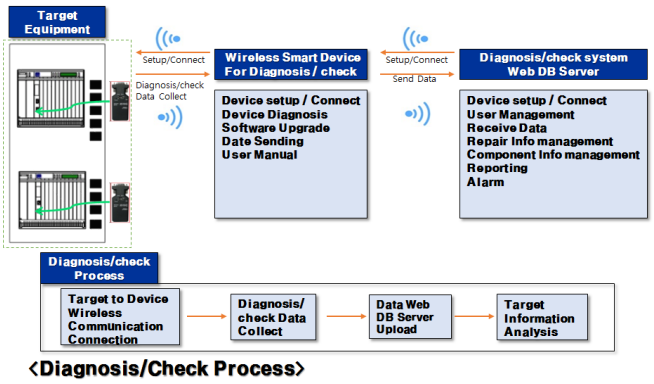

Figure 2. System operating process.

The mobile device application transmits the received data from the modules to the database using JSON. Once the data has been store on the database, anyone with proper authorization can access the data by serving a responsive web page that can be accessed by most devices with an Internet connection. Aside from searching the database, the database can also store data relevant to the vessel such third party companies that maintain or repair ships, its engineers, as well as keep a history log for future references. The system operating process is shown in figure 2. The mobile has functions such as device setting and connection, dignosis, software upgrade, data transmission, user manual. The Web DB server has the functions setting and connection, user management, data receiving, repair and component management, reporting, alarm.

2.3 Design of Database

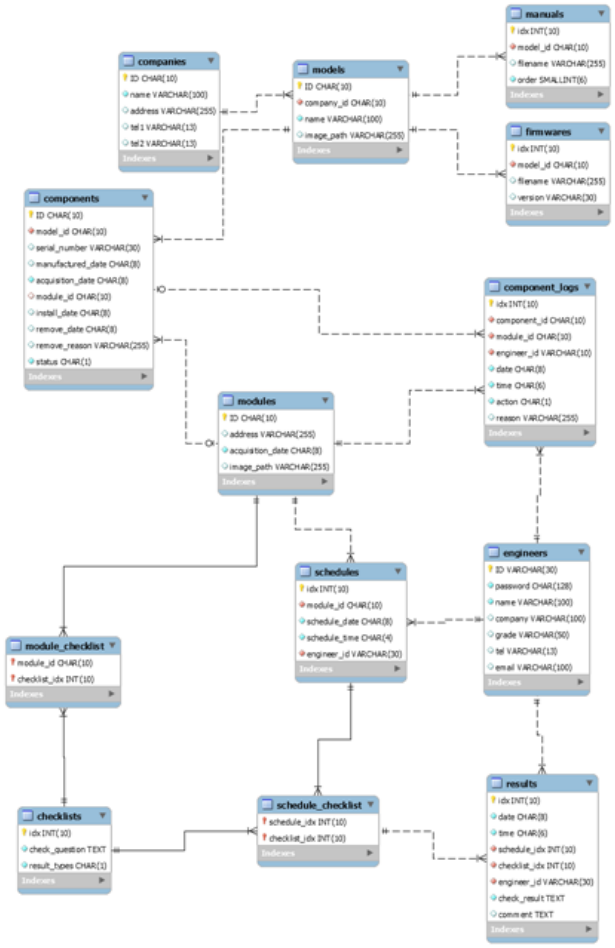

Figure 3. Datebase design.

In this paper, the database was developed by using MySQL. As shown figure 3, the main tables in the database are modules, componenets, schedules, results, engineers, checklists, and component logs. The modules table contains the list of the modules or devices used to acquire the data for maintenance and later diagnosis. The components table contains the components of a machine the module is connected to. Before diagnosing a machine, the engineer needs to first make a schedule which is stored in the schedules table and the corresponding data are stored in the results tables. A log file of all activities for future references are stored in the component logs table. The tables like companies, models, manuals, firmwares

2.4 Design of Mobile Application

The design of the mobile application and each page is shown in figure 4. First, the engineer connects to the module by setting up a bluetooth connection. The mobile scans for nearby modules that could be connected. After setting up the bluetooth connection, the engineer login to the connected module. A list of all the components that had been previously scheduled to be diagnosed is shown on the application. The engineer has the option to select which components he wants to be diagnosed. After retrieving the data from the modules, the results is then sent to the webserver.

Figure 4. Design of the Mobile application.

Figure 5. Bluetooth connection.

In bluetooth setup as shown in figure 5, there are three functions. The log file save tap check if save the diagnosis results together or not. The device module search tap has function of turning on bluetooth and scanning the device that can be connected. And this tap shows the device list. Next, in login display, the connected device name comes up and application wants the password. After setting up the bluetooth connection, user can see the module diagnosis screen. In this screen, there are serveral features and shows the detail results of diagnosis as shown in figure 6. The left side check box is for selection of device modules. The right top box is diagno-

Figure 6. Module diagnosis and results.

sis button, so if user click this button, mobile send a diagnosis order and wait the response. The list of red and green boxes shows diagnosis results. This list presents the response about prior order and if success, the box will be green, if not, will be red. The center bottom button has feature of transmission the results. By this button, user can send the results data to web server and save in database.

3. Implementation

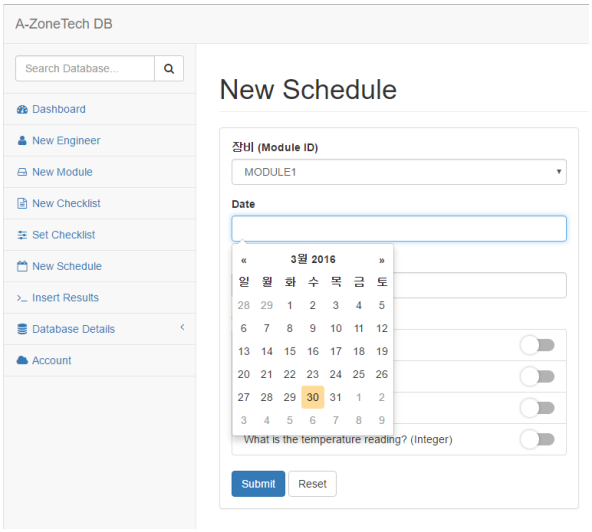

Figure 7. Example of an Input Form.

The web server provides the services and has access link like fcsldb.kumoh.ac.kr:8080/fcsl. When developing the web-based interface as shown in figures 7 and 8, Eclipse Mars Java EE IDE for Web Developers was used. At this moment, the version of the running software are: Apache Tomcat 8.0.30, MySQL 5.7.10.0-Community Edition and Java Server JRE 8u71. Sending data to the database as shown in figure 7 and searching as shown in figure 8 was made using the Bootstrap framework, which provides a responsive, intuitive, and powerful mobile first front-end framework.

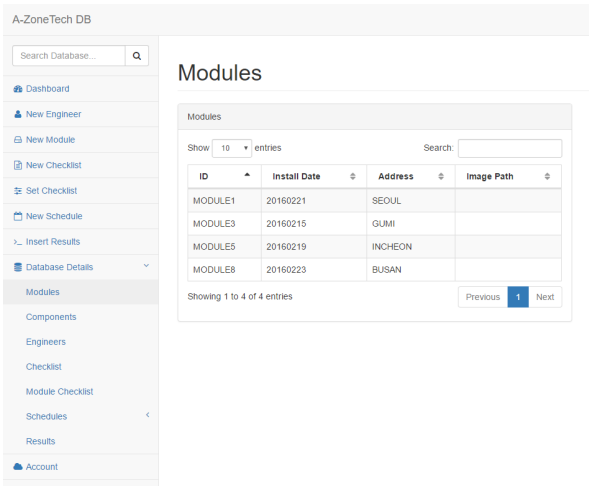

Figure 8. Example of Searching the Database.

4. Discussion and Conclusion

This project emerges from the need to easy monitoring of equipment in a maritime vessel. A web-accessible database adapted for monitoring and maintaining equipment on-board most ships to engineers for easy monitoring and quick data retrieval was made. A module retrieves relevant and important data then transmits them to a mobile device which sends them to an off-site database for more detailed analysis later. The web-accessible database has a friendly interface to support engineers, whether on-site or off-site for easy vessel equipment monitoring.

References

[1] N. Miyamoto et al., "Incremental Data Migration for Multi-Database Systems Based on MySQL with SPIDER Storage Engine," *2014 IIAI 3rd International Conference on Advanced Applied Informatics, 2014* ©IEEE. pp.745 - 750, doi: 10.1109/IIAI-AAI.2014.151.

[2] S. Carrasco et al., "Sharing Acute Myocardial Infarction Databases through the Internet with MySQL and PHP: A Web-Accessible Database for Clinical Research Networks," *Computers in Cardiology, 2007* ©IEEE. pp.605 - 608, doi: 10.1109/CIC.2007.4745558.

[3] Apache Tomcat Server. [Online]. Available: http://tomcat.apache.org/

[4] MySQL 5.7 Database Server - Community Edition. [Online]. Available: http://dev.mysql.com/downloads/mysql/