http://dx.doi.org/10.5392/IJoC.2013.9.1.011

A Study of the HTML5-based Mobile Order Communication System

Yoon-Ae Ahn
Department of Medical Informatics & Engineering
Korea National University of Transportation, Jeungpyeng, Chungbuk, 368-701, Republic of Korea

Han-Jin Cho
Department of Smart Mobile
Far East University, Gamgok, Eumsung, Chungbuk, 369-700, Republic of Korea

ABSTRACT

Recently, online real-time web accessible mobile devices such as smartphones, tablet PCs and application services have been developing rapidly. Hospitals also need to adopt an efficient information system that can provide decent medical services under mobile computing environments to complete future medical services. This study proposes a system in which a doctor can examine a patient and make out a prescription in a ward on a real-time based on the current Order Communication Systems (OCSs) through mobile interfaces. The proposed system implements mobile web pages using HTML5, instead of a mobile app. Web processing speed can be enhanced using the web socket and web storage functions of HTML5.

Key words*: HTML5, Web Socket, Mobile Web, Order Communication System, Smartphone.*

1. INTRODUCTION

Recently, online real-time web accessible mobile devices such as smartphones, tablet PCs and application services have been developing rapidly. As a result, there has been a great demand for web application services from diverse internet-based devices. To this end, HTML5, the next-generation web standard, provides API, which enables the development of diverse applications.

In a hospital, doctors give orders (ex: Treatment, operation, examination, medication, injection, etc.) regarding hospitalized patients using a hospitalization OCS while the nurse reads the order and cares for patients based on the prescription. Under the hospitalization OCS, unlike outpatient OCS, the medical expenses from the date hospitalized to the date discharged from the hospital are estimated. Therefore, the final calculation of the medical costs for hospitalized patients is performed on the date discharged from the hospital [1].

Because hospitalized patients usually receive treatment in their ward, management could be more efficient if the doctor used the OCS on a real-time basis while on the move. To achieve future medical services in a hospital, it is necessary to introduce an efficient information system that provides a mobile computing environment in order to achieve future medical services.

Thus far, smartphones have been used in diverse sectors in Korea but this new technology is as yet rare in the hospital information system. Considering the convenience and popularity of smartphones, it is necessary to perform a study on application services that guarantee timely and prompt treatment for hospitalized patients using these devices.

Hospitals are also in need of a system that enables the monitoring of orders for hospitalized patients on a real-time basis to enhance patient services and build a rapid work environment. In particular, it is required to minimize order processing time. So far, hospital information systems have been developed in diverse fields using the Internet, web, wireless technology and mobile technologies. New technology in an HTML5-based mobile web service system has, however, been difficult to find.

In Korea, no medical treatments or prescriptions have been received from doctors by using smart devices so far, since the telemedicine was impermissible. In accordance with the medical device law, what is worse, OCSs are not permissible to be developed and used as a medical app [2]. For that reason, in this paper, a study is conducted on an OCS in a mobile web, instead of a mobile app, that is available only in hospital. In particular, this study proposes a mobile OCS based on HTML5 that has recently achieved prominence as the up-to-date web standard for mobile web development.

In particular, one aim was to develop the OCS that makes the efficient management of hospitalized patients possible. This study therefore designs the structure of a mobile OCS, which would be independently applicable to devices anytime and

* *Corresponding author, Email : hjcho@kdu.ac.kr*
Manuscript received Oct. 18, 2012; revised Feb 28, 2013; accepted Mar 08, 2013

anywhere using HTML5 under a web-based environment, and implements server and client interfaces.

2. RELATED STUDIES

Recently, many large hospitals have introduced smartphones and tablet PC-based smart medical examination services to their hospital information system to enhance the quality and efficiency of medical services. In addition, people's interest in Personal Health Record (PHR) services, which control personal medical records through PCs and smartphones, has increased, and related technologies have been under development [3].

In this study, the latest studies on the smart hospital information system are examined. In [4], the development plan for an integrated medical information system and operational duties were mentioned to enhance medical examination services to provide integrated medical information. In [5], a medical information system access control security mechanism was designed as a web-based information usage security measure. In addition, the contents of the software architecture design mapping-based design were proposed. In [3], a gateway system that can integrate and interlock medical information under a multiplatform environment has been designed. In terms of a data mapping technique that is the core function of the gateway system, CCR standards have been applied, and a multi-table-based mapping method has been used. In [6], the improvement of efficiency of PACS using a remote system has been verified via testing. In [7], an inter-operable framework between the grid and PACS has been proposed using web services for the efficient management of medical data. A web services-based grid service mediator, which enables interlocking among PACS and management of medical data, has been developed.

In addition, studies associated with a mobile technology-based hospital information system are examined. In [8], the factors that have an influence on the intention to use mobile information technology were investigated. As a result, it has been verified that as compatibility and usability are high, and individual, technical and organizational preparation is well established in nursing duties, nurse dependence on wireless information technology was high. In [9], the necessities of adequate information exchange among various organizations, hospitals and individuals for the establishment of an advanced medical information system were mentioned. For this, the current development of middleware in the medical information sector was reviewed. In [10], a system has been implemented that allows interpretation doctors or surgeons to access medical images even outside of the hospital setting using PDAs and give orders on patients remotely. In [11], a system has been developed to improve workflow by applying RFID and workflow-centered OCS so as to increase work efficiency in Radiology Departments.

Next, the application of HTML5 technology that has emerged as the next-generation standard for web development is examined. In [12], it was forecasted that because HTML5 has many advantages for the mobile environment, mobile environment-centered web development would spread. In [13], a display system that allows the checking of AIS information using an online commercial map was implemented. For display processing, the next-generation web standard 'HTML5 technology' was used. In [14], functions were implemented to store results searched for on the website in the local storage of the browser using the website storage of HTML5 and search data even when the network is disconnected. In [15], it was mentioned that HTML5 is appropriate for application to the mobile environment, which is sensitive to network speed, because it offers various functions such as offline web application, web form, and web socket.

So far, the hospital information system has been examined in diverse sectors using the Internet, web, wireless technology and mobile technology. However, its application to the mobile web-based system using HTML5 is still rare. Therefore, this study proposes an HTML5-based mobile OCS. If web storage functions are used using HTML 5, it does not need to call up basic personal patient information from the server every time. The time to access to the server is shortened because newly updated information only is loaded. In addition, access to the following information (ex: Patient history, drug code, name of drug, etc.) becomes faster.

3. DESIGN OF AN HTML5-BASED MOBILE OCS

The operation of the OCS is mostly one in which the doctor enters orders in a computer in his or her office. However, smart diagnosis services would be available if doctors are able to enter orders on a real-time basis by checking the patients' status in the ward.

In this paper, HTML5 is used for mobile web development, instead of existing HTML forms. HTML5 supports functions, such as multimedia, graphics, location information and web-based communication, which could be unsupported by existing HTML. First of all, HTML5 is available in all devices, irrespective of smartphone platforms, for example, android or iPhone OS. It is also more advantageous to be available with no program modification even in tablet PCs or general personal computers. Accordingly, HTML5 has been in the spotlight as a next-generation web standard [12].

The proposed system is available in a HTML5-based mobile web service form. In addition, it can be interlocked with the OCS of a hospital information system. The proposed system is structured as follows:

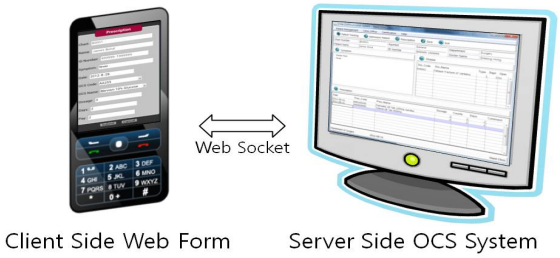

Fig. 1. System Structure

The client side part in Fig. 1 reveals a system domain used by the doctor to examine patients in the ward. The doctor can

enter orders on a real-time basis through the web page of mobile devices. The entered orders are transmitted to the OCS server using the web socket function of HTML5. The server side part is the server of the hospital information system. The OCS receives the orders entered over the mobile web through a web socket on a real-time basis. The received data are saved in the hospital database and can be viewed through the order screen in the doctor's office.

HTML5 provides an API to make direct socket communication using JavaScript in a browser. Here this function is supported by web socket. For web socket implementation, it is required to implement a separate API and construct a web server. The web socket has the same features as in general TCP socket communication [13]. Accordingly, without implementation of the web socket server, HTML5 socket communication is not allowed on a mobile web. A mobile browser, moreover, connects the web server and receives web pages and scripts. Without the web server, no web page is served to users.

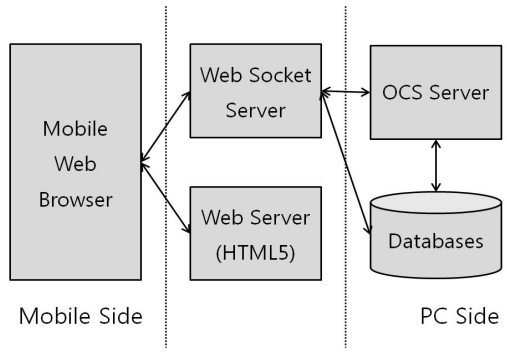

Fig. 2. Software Module Operating Flow

In Fig. 2, a mobile web browser receives web pages and script after accessing the web server. In order for a mobile browser that has been implemented with HTML5 to carry on a socket communication with different client server system, there needs to be a web socket server in between. If OCS is developed in a form of a web page, then web socket server is unnecessary for data can be transferred through a web server. However, most of the OCSs that are being used domestically are developed as client-server form. Therefore in order for a mobile web communication there need to exist a separate communication server.

The OCS server saves the information received from the mobile browser in the hospital database. The database then saves and manages all hospital information using the MS SQL server or Oracle.

4. DESIGN AND IMPLEMENTATION OF THE SERVER SYSTEM

The OCS makes the orders created in the process of examining patients available to all related departments in a hospital. It connects all orders in a hospital and communicates them on a real-time basis.

4.1 OCS Server

The orders delivered through the mobile interface are received by the OCS server and saved in the hospital database. They are then designed to be searched. First, order input and search pages that are used in a doctor's office through JAVA are implemented to provide a simple OCS server page.

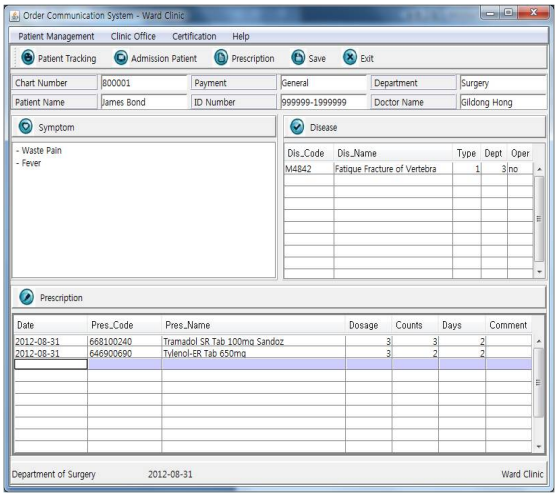

Fig. 3. Order Input & Search Page

Fig. 3 shows the input and search of orders in the OCS. The orders, which were entered through the mobile browser, are sent to the server as message through data communication with a web socket server and saved in the hospital database. These orders, then saved in the server system as shown in Fig. 3, can be searched.

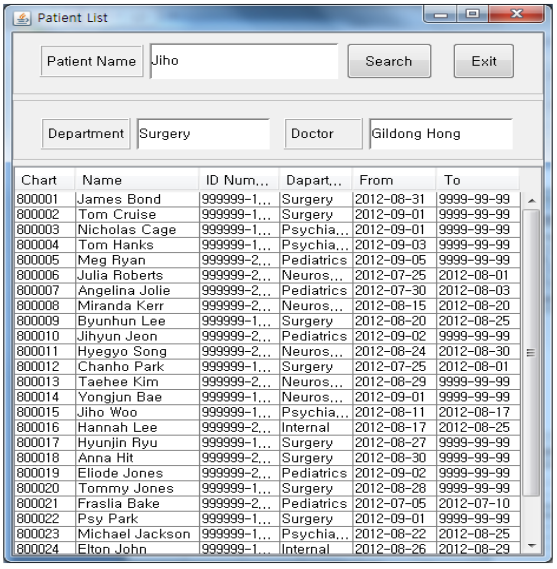

Fig. 4. Patient List

Fig. 4 shows the search of patient information to enter orders. In terms of patient information, chart No., name, resident registration number and treatment period are saved. These contents can be checked on a real-time basis in the mobile browser.

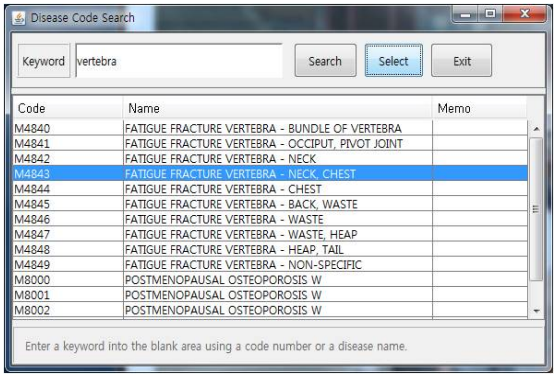

Fig. 5. Disease Code Search

Fig. 5 shows the search of disease code and name to enter orders. Patients' disease code and name should be recorded exactly in accordance with the Korean Standard Classification of Diseases. It should be possible to search these data in the mobile browser.

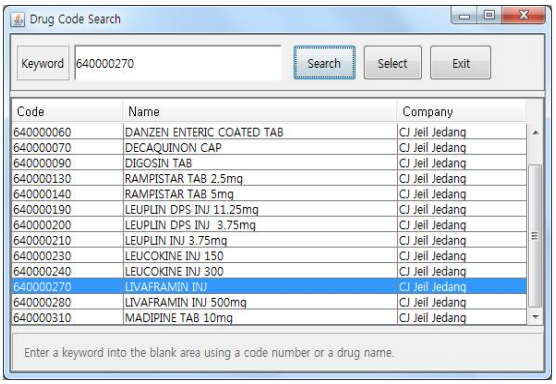

Fig. 6. Drug Code Search

Fig. 6 shows the search of drug code and name saved in the hospital database. The drug code and name should be exactly recorded in accordance with the Health Insurance Review & Assessment Service. The information should be available in the mobile browser as well.

4.2 Web Socket Server

A web socket server can be implemented into a Web Application Server (WAS) module type or stand-alone server. In this study, a stand-alone type server is implemented using jWebSocket [16]. Fig. 7 shows the process of sending and receiving messages between a mobile browser and OCS server through the web socket.

As shown in Fig. 7, a web socket server always waits for connection between the mobile client and the OCS server. First, a mobile browser attempts to be connected with the web socket server. Once successfully connected, it sends a message to the web socket server. The web socket server then resends the message to the OCS server. In this kind of process, data are communicated between the mobile browser and OCS. Regarding client connection, server events are processed (listener), and functions are provided as follows: First, a listener class is created in the server. Second, the server listener is registered in the jWebSocket server. Lastly, the functions of the listener are used in the webpage.

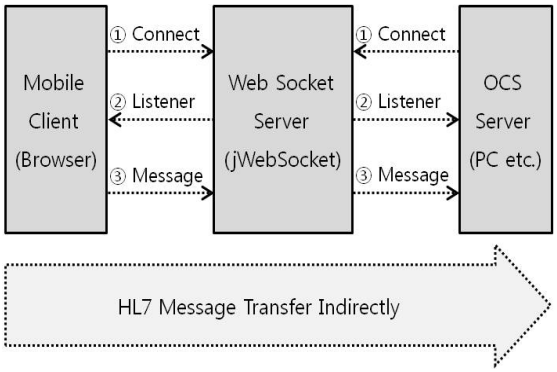

Fig. 7. Web Socket Server and Client Operation

```
//Step 1: Create a Server Side Listener
public class OCSTockenListener implements
                WebSocketTokenListener {
  private static Logger log =
            Logging.getLogger(OCSTockenListener.class);
  public void processOpened(WebSocketEvent e) {
     log.info("Client '"+e.getSessionId()+"'connected."); }
  public void processPacket(WebSocketEvent e,
          WebSocketPacket p) { }
  public void processToken(WebSocketTokenEvent e,Token t) {
     log.info("Client '"+e.getSessionId()+"' sent Token: '"+
                t.toString()+"'.");
     String ns = t.getNS();
     String typ = t.getType();
     if (typ != null && "my.namespace".equals(ns)) {
       Token res = e.createResponse(t);
       if ("getInfo".equals(typ)) {
         res.put("vendor", JWebSocketConstants.VENDOR);
         res.put("version", JWebSocketConstants.VERSION_STR);
         res.put("copyright", JWebSocketConstants.COPYRIGHT);
         res.put("license", JWebSocketConstants.LICENSE);
     } else {
         res.put("code", -1);
         res.put("msg", "Token type '" + typ + "' not supported
                  in namespace '" + ns + "'.");
     }
     e.sendToken(res); } }
  public void processClosed(WebSocketEvent e) {
     log.info("Client '"+e.getSessionId()+"' disconnected."); }
}
```

Fig. 8. Creation of Listener in the Server

Fig. 8 shows the algorithm for creating a listener class in the web socket server. The OCSTokenListener implements and prepares the WebSocketTokenListener interface provided from the jWebSocket. In the OCSTokenListener class, the client connection waits and the connected clients are checked.

```
//Step 2: Register Listener at the jWebSocket Server
TokenServer sv = (sv)JWebSocketFactory.getServer("ts0");
if(sv != null) {
   sv.addListener(new OCSTockenListener());
}

//Step 3: Use the Listener Capabilities in the Web Page
var tt = {
   ns: "my.namespace",
   type: "getInfo"
};
jWebSocketClient.sendToken(tt, {
   OnResponse: function(t) {
      log("Server responded: " + "vendor: " + t.vendor +
          ", version: " + t.version);
   }
});
```

Fig. 9. Registration of Server Listener

As shown in Fig. 9, the OCSTokenListener class is registered in the token server of jWebSocket. The server listener is then activated by the JavaScript source code of the client.

5. DESIGN AND IMPLEMENTATION OF MOBILE CLIENTS

In this study, a mobile web page is designed using HTML5. It is tested using a Chrome browser on a PC, and the mobile browser is executed through an Android-powered smartphone.

5.1 Mobile Browser

If the web form of HTML5 is used, improved options over conventional web pages can be utilized. The web form presents markups, attributes and events to get the user's input information.

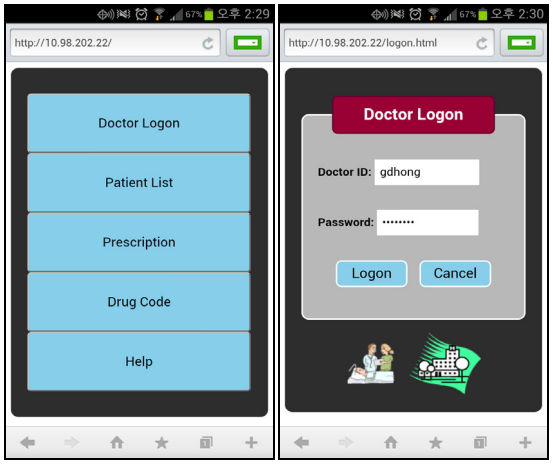

Fig. 10. Setup Menu and Logon

Fig. 10 shows the default page of the website connected to the mobile OCS. The menus consist of Doctor's Logon, Patient List, Prescription, Drug Code and Help. In order for a doctor to enter orders, he/she should log on using his/her ID and password.

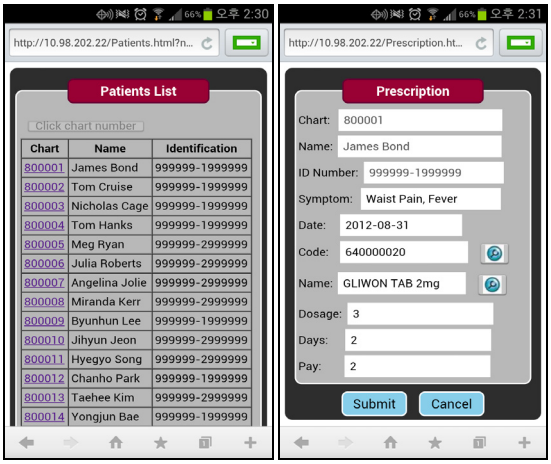

Fig. 11. Patient Information Search and Order Input

Fig. 11 displays a search for patient information before orders are entered. If a chart number is selected in the patient list, an order input page pops up. The drug code and name can be searched by clicking the search button.

The local storage of HTML5 is used to enhance conventional cookie functions. It provides these functions to save data comprised of a pair of key and value in the web client. The patient information in Fig. 11 can be quickly searched using the local storage.

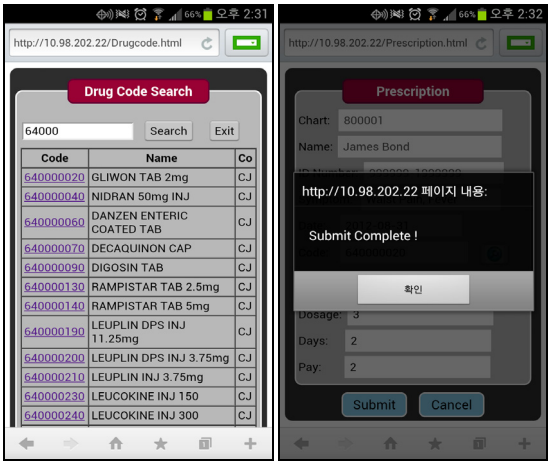

Fig. 12. Drug Code Search and Order Communication

Fig. 12 shows a search of drug codes when orders are entered. As soon as an order is submitted, the input process is completed. The data are then sent to the OCS via a web socket server. Because the web socket provides a two-way communication channel in the mobile browser, once a web socket is obtained, data are exchanged only between the mobile browser and server. Therefore, this function is essential in a speed-sensitive mobile web environment.

5.2 Web Socket Client

Once the server listener class is implemented, a client code is created. In the mobile browser, the web socket of HTML5 is used to send data to the OCS server. All these processes are executed through a total of six stages as shown in Fig. 13 below:

```
//Step 1: Embedding the jWebSocket Script
<script type="text/javascript" src="../../res/jWebSocket.js" />
//Step 2: Creating the jWebSocketClient instance
if (jws.browserSupportsWebSockets()) {
   jWebSocketClient = new jws.jWebSocketJSONClient();
} else {
   var msg = jws.MSG_WS_NOT_SUPPORTED;
   alert(msg);
}
//Step 3: Connect and Logon
log("Connecting to "+ocsURL+" and logging in as '"+uname);
var res = jWebSocketClient.logon(ocsURL, uname, pswd, {
   OnOpen: function(e) {
      log("<font style='color:#888'>jWebSocket connection
      established.</font>"); },
   OnMessage: function(e, t) {
      log("<font style='color:#888'>jWebSocket '"+t.type+"'
      token received, full message: '"+e.data+"'</font>"); },
   OnClose: function(e) {
      log("<font style='color:#888'>jWebSocket connection
      closed.</font>"); }
});
//Step 4: Sending Tokens
if(msg.length > 0) {
   var res = jWebSocketClient.sendText(ocsTarget, msg);
   if(res.code != 0 ) { }
}
//Step 5: Processing Incoming Messages
OnMessage: function(e, t) {
   log("<font style='color:#888'>jWebSocket '" + t.type +
       "' token received, full message: '" + e.data + "'</font>");
}
//Step 6: Logoff and Disconnect
if( jWebSocketClient ) {
   jWebSocketClient.close();
}
```

Fig. 13. Client Web Socket Processing

Fig. 13 shows an algorithm to process the web socket in the mobile browser. First, 'jWebSocket.js' file is embedded in the web page using the HTML <script> tag. Second, clients are activated after creating a jWebSocketClient. Third, this is logged on to the ocsURL (jWebSocket server). Fourth, a message is sent to the ocsTarget client through jWebSocket. Fifth, the received message is managed. Lastly, client access is terminated.

5.3 Analysis

To analyze advantages and disadvantages of the mobile-based OCS system proposed in this paper, the differences in feature between the existing client-server system and the proposed system are compared as follows.

Table 1. Analyzed features of the proposed system

Item	Existing System (Client-Server Only)		Proposed System (Mobile Web + Client-Sever)	
Mobile Prescription	Available only in consultant room	×	All available in hospital	○
Real-time Prescription	Unavailable in ward	×	Available in ward	○
Data Communication	Available in Socket Communication	○	Available in socket communication using Web Socket	△
Mobile Scalability	Platform-dependent Complete redevelopment required to be suitable for features of devices	×	Mobile platform-independent Available in smartphones, laptops, PCs, etc. with simple changes	△
u-Healthcare Extension	Partly suitable	△	Very suitable	○
○(Excellent) △(Good) ×(Bad)				

The proposed system is advantageous to prescribe medicines for patients not only in a consultation room but in a ward, even while direct medical treatment. It also allows socket communication via web socket, like the existing client-server system, to have the advantage in terms of communication speed. Due to HTML5-based design, particularly, it shows very high scalability to different mobile devices. The existing client-server type OCSs, on the other hand, require many operations to change into a system suitable for u-Healthcare environments in the future. However, the proposed system is easy for system extension with some changes as it was designed in view of u-Healthcare.

6. CONCLUSION

This study has designed a structure of a mobile OCS that interlocks HTML5 and an OCS server and implemented server and client interfaces. If a web storage function is used utilizing HTML5, it will not have to call the patient's basic personal information from the server every time. Because only newly updated information is loaded, in addition, the time needed for connection to the server is shortened. In addition, access to the patient's history information, drug codes and drug names is made more rapidly.

Because both the web socket server and web server have various implementation techniques, optional development is enabled. As a result, the connection between the latest mobile technology and hospital information system would be promoted.

Implementation and testing of a system connected to the OCS server of a hospital is needed in the future. In addition, investigation of the development of diverse mobile browser pages is also required.

REFERENCES

[1] S.H. Kim and M.S. Go, *Hospital Information System Utilization*, Hyun Moon Publisher, pp. 19-24, 2010.

[2] Y.H. Yoon, I.K. Kim, and B.K. Yi, "Trend Analysis for Establish App-permit System as A Medical Device", Communications of the Korean Institute of Information Scientists and Engineers, vol. 29, no. 4, 2011, pp. 69-77.

[3] W.H. Shim, H.S. Na, and S.C. Park, "Gateway design and performance evaluation based on health information standards in multi-platform environment", Journal of the Korea Society of Computer Information, vol. 17, no. 3, 2012, pp. 33-40.

[4] B.S. Kim, "U-Healthcare & Medical Information System of Status and Operative Challenges for Integrated Medical Information System", Journal of the Korea Society of Digital Policy and Management, vol. 9, no. 5, 2011, pp. 65-75.

[5] S.C. Noh and J.H. Hwang, "A Study of Software Architecture Design Methods for Multiple Access Control under Web-based Medical Information System Environment", Journal of the Korea Information Assurance Society, vol. 11, no. 4, 2011, pp. 43-49.

[6] J.H. Kim, J.W. Lee, S.J. Lee, and K.R. Dong, "Evaluating the Effectiveness of Work with the Remote System in PACS Room", Journal of the Korean Society for Digital Imaging in Medicine, vol. 13, no. 4, 2011, pp. 171-175.

[7] B.H. Lee and H.S. Cho, "Interoperability Framework between GRID and PACS based on Web Services", Journal of the Korea Institute of Information and Communication Engineering, vol. 14, no. 8, 2010, pp. 1799-1808.

[8] E.J. Lee, Y.J. Seo, Y.H. Kim, and J.Y. Oh, "Determinants of the Intent to Use a Wireless Technology of a University Hospital Nurses", Journal of the Korean Health Policy & Administration, vol. 20, no. 3, 2010, pp. 58-72.

[9] S.I. Yang, "A Suitable Medical Information Middleware System for Mobile Computing Environment", The Korea Society of Computer Information Review, vol. 19, no. 1, 2011, pp. 39-47.

[10] Y.S. Ji, K.R. Dong, and C.B. Kim, "Implementation of PACS using PDA System on Medical Images", Journal of the Korea Contents Association, vol. 9, no. 4, 2009, pp. 247-253.

[11] Y.S. Ji, K.R. Dong, and C.B. Kim, "Survey for Patient Satisfaction Rate & Patient Leading System Development through RFID and OCS Worklist Program Connection", Journal of the Korea Contents Association, vol. 9, no. 5, 2009, pp. 197-205.

[12] E.M. Lee, "Impact of HTML5 in the Web Environment", Communications of the Korean Institute of Information Scientists and Engineers, vol. 29, no. 6, 2011, pp. 55-60.

[13] S.J. Lee and J.W. Lee, "A Development of AIS Vessel Monitoring System on online map using HTML5", Journal of Korean navigation and port research, vol. 35, no. 6, 2011, pp. 463-467.

[14] K.K. Lee and J.Y. Choi, "A Study on the Use of the Web Storage HTML5", *Proc. of the Korean Information Science Society Conference*, vol. 38 no. 1(B), 2011, pp. 430-433.

[15] W.S. Lee, "HTML5 and Mobile Web", TTA Journal, sno. 128, 2010, pp. 50-54.

[16] GNU, "jWebSocket", http://jwebsocket.org/, 2012.

Yoon-Ae Ahn

She received the B.S., M.S in computer science from Hnanam University, Chungbuk National University, Korea in 1993, 1996 respectively and also received Ph.D. in computer science from Chungbuk National University, Korea in 2003. Since then, she has been with the dept. of Medical Informatics & Engineering, Korea National University of Transportation. Her main research interests include mobile applications and medical information systems.

Han-Jin Cho

He received the B.S., M.S and Ph.D. in Computer Engineering from Hannam University, Korea in 1997, 1999 and 2002. Since then, he has been with the dept. of Smart Mobile, Far East University, Korea. His main research interests include mobile applications and network security.