

Learning Media Repository and Delivery System for Smart Classrooms using IoT and Mobile Technologies

<https://doi.org/10.3991/ijim.v13i02.9941>

Kobkiat Saraubon

King Mongkut's University of Technology North Bangkok, Thailand

callkob@gmail.com

Abstract—This paper presents a Learning Media Repository and Delivery System (LMRDS) for a smart classroom using the Internet of Things (IoT) and mobile technologies. It was designed to support active learning pedagogy. Teachers are able to broadcast learning media or course materials directly to the students' mobile devices; after that the students can interact with the media by drawing, editing or adding comments using their mobile device then broadcast it back to present or reflect their thinking. The system includes: 1) a server using a Raspberry Pi 3B+ and 2) mobile devices. The system supports full features involving two approaches in the form of an Internet model and a non-Internet model. The mobile applications were implemented using a cross-platform approach to support major mobile platforms including iOS and Android. The evaluation had three dimensions in terms of usability, functionality and security. The results revealed that all dimensions were evaluated highly. The teacher and students were highly satisfied with the system.

Keywords—Internet of Things (IoT), Smart classroom, Raspberry Pi

1 Introduction

Students need to develop twenty-first century skills to succeed in the information age [1]. The Partnership for 21st Century Skills lists three types of skills: learning skills, literacy skills and life skills. Students who lack or do not acquire will face stiff competition because the skills required in the workplace will continue to increase in tandem with economic development and new technologies in the global market. It is also reported that employers want workers who can think critically, solve problems creatively, innovate, collaborate and communicate [2]. These skills have always been important for students who will need to find and hold jobs in industry. To have these skills, students need to think deeply about issues, solve problems creatively, work in teams, communicate clearly in many media, learn ever-changing technologies and deal with a flood of information.

Research shows that learners learn best when they engage with the course materials and actively participate in learning activities such as analyzing, writing, reflecting and presenting [3] [4]. The traditional teaching model has forced learners to be passive in the classroom, listening to their teachers. Research however, suggests that learners

must do more than just listen; they must read, write and discuss to solve problems [4] [5]. Active Learning is a model of instruction that focuses on the activities and responsibility of learning. It promotes activities that involve learners in doing things and thinking about what information or tasks they are learning and doing. With this strategy, learners must be engaged in higher-order thinking such as analysis, synthesis and creation. Many effective learning models such as the Crystal-based Learning, 5E model [4] [6] among others encourage students to analyze and reflect on their thinking and creating solutions to meet twenty-first century skills. Presenting to classmates and the teacher is considered a critical step that allows reflection, facilitates comprehension and increases interest in the learning environment [7]. Especially for higher education, writing and presenting are practical and powerful teaching strategies [8].

In this study, we propose a Learning Media Repository and Delivery System (LMRDS) for a smart classroom using the Internet of Things (IoT) and mobile technologies. It is comprised of a media center with storage and android applications for students' mobile devices. The system allows the teacher to broadcast learning media directly to students' mobile devices. In addition, students are able to share or present their thinking by writing on the mobile screen with their fingers. The presentation will be broadcast to the classmates' mobile devices and the projector.

2 Literature Review

Interactive whiteboards (IWB), also called smart boards, have replaced the traditional blackboard in the smart classroom. It is a computer driven device that provides powerful multimedia presentation facilities and allows users (students and teachers) to access and manipulate electronic files by means of a projector (only for whiteboards) and the board's surface. The IWB serves a variety of functions including display, write and save media files. Studies related to the use of such boards in educational settings have shown that the technology can promote teacher-student interaction and student participation in the classroom [9] [10]. However, the device is quite expensive.

The IoT is a network comprising of various end-nodes, and technological and physical environments. The physical environment consists of human and nonhuman objects (referred to as things) that are linked together with a wireless network infrastructure that enables interaction and communication between the objects and the physical environment [11]. One of the very smart aspects of modern schools and classrooms is that the IoT improves the education itself and brings advanced value to the physical environment and structures [12]. The rapid development of the IoT and mobile technologies has attracted the attention of educational researchers. Many studies have investigated the use of IoT and mobile technologies as a complementary teaching strategy to acquire the highest learning quality and reduce constraints within the learning environment [3] [13] [14].

The Raspberry Pi is a low-cost (approximately 35USD), credit-card-size, single-board computer that can be plugged into a computer monitor or projector that uses a standard keyboard and mouse [15]. It contains a processor and connectors for external

devices including USB and HDMI. It also has the ability to interact with the outside world, and has been used in a wide array of digital maker projects from an end-node to a server. The Raspberry Pi has been used widely and effectively in IoT-based applications and prototyping solutions [16]. Saheb et al. [17] present the IoT-based Lecture Delivery System using the Raspberry Pi. The system was implemented to be a media composer. Teachers recorded an audio file in .WAV and a video file in MP4 using their smartphone or in .h264 format by a camera module connected to the Raspberry Pi separately. The video file was converted into MP4 format and then merged on the audio file into .MOV format using a program in the Raspberry Pi; it was then uploaded to the main server using FTP protocol.

Mobile devices are considered convenient learning tools that provide access to all the different learning materials available. Mobile application development approaches are categorized into three categories: 1) Native mobile application 2) Mobile web application and 3) Cross-platform mobile application, which is also known as a hybrid mobile application.

A native mobile application is developed in a specific programming language provided by the platform vendors such as Objective-C and Swift for iOS, or Java and Kotlin for Android operating systems. Native mobile applications provide fast performance, rich user experience and a high degree of reliability. However, it is time-consuming to develop because it is tied to one type of operating system, which forces the company or developer creating the app to make duplicate versions that work on other platforms. As a result, the development costs are higher than the other approaches.

A mobile web application is a web application developed for mobile devices. Within this approach, there are Responsive Web Applications (RWAs) [18] and Progressive Web Applications (PWAs). An RWA is designed to render the web content well on any screen platform and screen size using a custom CSS and HTML to resize, hide, shrink, enlarge or move the content. It is an online website hosted on a web server. The web application can be accessed through any mobile device using a built-in web-browser application. The downside is that the RWAs need a good Internet connection to load the web content and interact with the users because the RWA is not installed on the device. Another downside for the RWAs is that accessing the device hardware is limited. A PWA [19] is a progression from an RWA that combines the best of the web and native applications by leveraging the new features supported by modern browsers to deliver an app-like experience to users. These features include using service workers, web app manifests, push notifications, app shell architecture, offline support, smooth UIs with 60fps animations, access to the file system (some web browsers) and access to some device hardware. Service workers are event-driven workers that run in the background of an application, and which are separate from a web page. A service worker is a JavaScript that listens to events such as fetch, update, install, and perform tasks. It acts as a proxy between the network and application which is able to perform network requests and cache information in the background. The content is loaded onto the device for offline use. As a result, the PWAs are reliable user experiences that have the reach of the web, and which respond quickly to user interactions with smooth animations.

Cross-platform mobile application development employs non-native language such as HTML/CSS/JavaScript or C# to generate applications for multiple platforms. It is divided into three categories: Webview-based, Native Widget-based and Custom Widget-based applications. The Webview-based approach allows developers to create mobile applications using web technology such as HTML, CSS and JavaScript. The biggest issue of this approach is performance, which is far behind the performance of native applications. The Native Widget-based applications employ JavaScript and JSX (JavaScript syntax extension) by React Native to construct their application from components that are then mapped to the platform specific native widgets through a “bridge”. In order to create a high performance application, the number of components passing through the bridge should be low. The Custom Widget-based applications employ canvases to render a rich set of widgets for the whole UI. Table 1 illustrates the comparison among mobile application development approaches.

Table 1. Comparison table among the approaches

	Native	RWA	PWS	Cross-platform
<i>Performance</i>	<i>Excellent</i>	<i>Medium - Poor</i>	<i>Medium - High</i>	<i>Medium - High</i>
Web with HTTPS required	No	No	Yes	No
Technology stack	Different for each platform	Web (HTML5,CSS, JavaScript)	Web (HTML5,CSS, JavaScript)	Web and single codebase (C# .Net)
Hardware access	High	Low	Moderate	Moderate
Development cost	The highest if developed for multiple platforms	Low, similar to pure web costs	Moderate, similar to web costs, but extra skills are required for PWS	Moderate, similar to web costs, but extra skills are required

3 System Analysis and Design

The LMRDS is an IoT and mobile-based application dealing with three stakeholders including teacher/instructors and students. The design and development has been driven by the needs of major functionalities of mobile learning, multimedia learning, and the requirements of stakeholders that meet the principles of mobile learning, multimedia learning along with the learning management system [20] [21]. The requirements for the system were analyzed and concluded from sixty students and ten teachers through in-depth interviews under the requirement-engineering framework [22].

The following are the features of the proposed system that meet the requirements of the stakeholders. Fig. 1 illustrates the proposed system, which is comprised of a Raspberry Pi as the server and mobile devices (a projector and screen are optional).

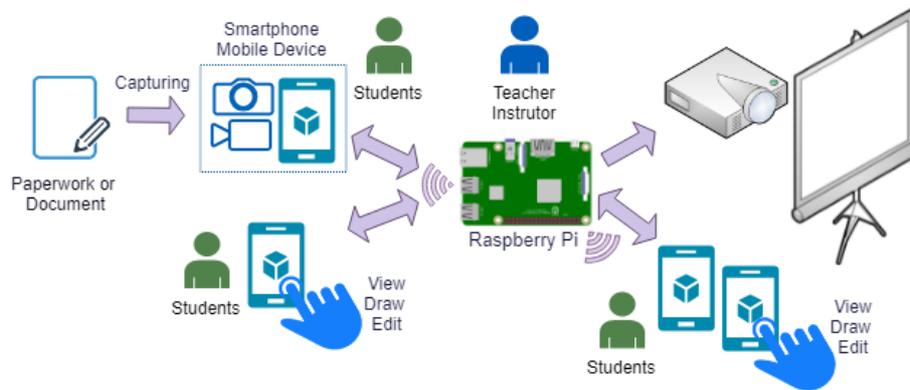


Fig. 1. Proposed system

- The teachers are able to broadcast the learning media/multimedia or course materials directly to the students' mobile devices;
- The learning media can be created and broadcast easily using the proposed application installed on a teacher's or students' mobile device;
- The system supports all device types (smartphones and tablets) and popular mobile platforms in the market (iOS and Android);
- The teachers are able to use the application to manipulate the files submitted by any student; for example, allowing the appropriate file to be broadcast over the class network, and to browse, view or remove files;
- The broadcast file is transferred to students' devices at a high speed (300-450 Mbps, 802.11n on the Raspberry Pi 3B and 1.3Gpbs, 802.11ac on the Raspberry Pi 3B+). The file will also be displayed on a projector connected to the system;
- The students are able to edit the shared file on their devices. They can add comments, edit or draw using their fingers then re-submit to the system to share and present to their classmates;
- The system is available in the Internet or non-Internet mode. A number of schools and institutions in developing countries, which do not provide Internet access, are able to use this system. This means that all learning media or course materials can be sent without connection to the Internet.

4 System Architecture

The system configuration is comprised of a Raspberry Pi server, clients (students' mobile devices), a projection screen and a conventional projector connected to the Raspberry Pi server using HDMI port or HDMI to VGA adapter. This section describes the component architecture as shown in Fig. 2.

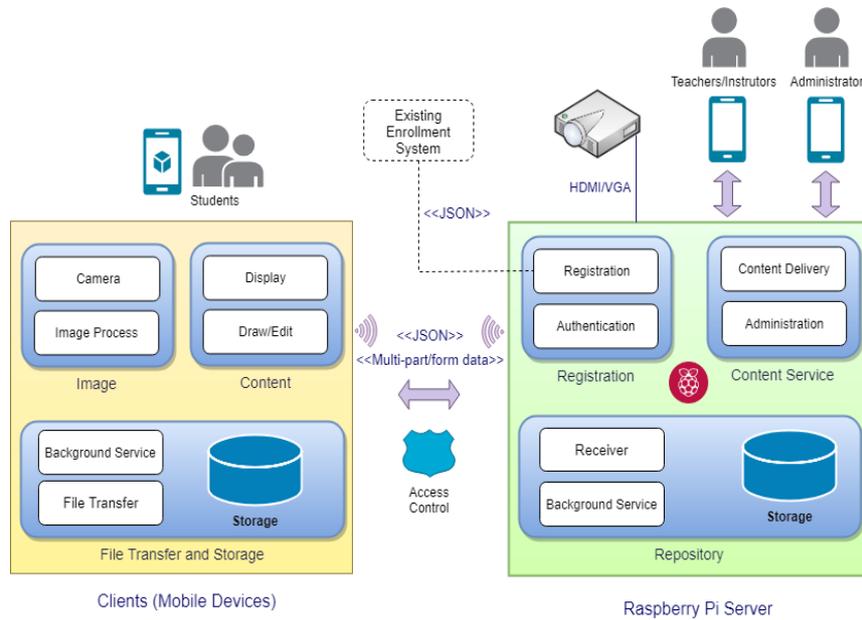


Fig. 2. System architecture

4.1 Server

The server system is comprised of three components as follows:

Registration

- The registration module allows students to register to access the system. There are two methods to register: signing up manually using their mobile devices or through the existing enrollment system. In the first method, students are not able to access the system until the accounts have been activated by the teacher or the administrator. In the second method, those students who have enrolled on the course do not need to register since the enrolled students' data from the institution's central server are automatically imported into the system without the teacher's activation.
- The authentication module: Allows students to sign into the system to access or use the services. Student IDs and passwords are used as the credentials for the authentication process. The credentials are sent to the server via a secure channel (SSL) to authenticate and identify a student's status. When the authentication process is finished, the connection is established then a secure token is sent to the client's mobile device. The token is stored in the mobile device under an application sandbox using a SharedPreferences mechanism (private mode) provided by the native SDK until the user signs out. The client authentication process is performed every time students use their mobile devices to access the server. All credentials are verified under a secure channel.

Content Service. The content delivery module delivers the learning media and course materials to the clients' devices (the student and teacher devices). To prevent junk media broadcast on the class network, the teacher must grant the file permissions. After that, the content is displayed on the clients' screens and the projection screen.

The administration module allows teachers/instructors to manage users' accounts (students and the teacher) and manipulate the learning media and course materials (add, update, remove and grant permissions to broadcast the files). The user interface for the module is based on responsive design creation that employs flexible layouts, flexible images and cascading style sheets. It is able to detect the client's screen size and orientation, and adjust the layout accordingly.

Repository

- The receiver module receives JSON data and digital content (meta-data and multi-part-formed data) captured by client's devices and then store it in the system's main storage.
- The background service module is a daemon service linked to the content delivery module, which allows students to download learning media or course materials from the Raspberry Pi server onto their devices. As a result, the users are able to open those contents when they are not connected to the server (at home or outside the classroom). This service improves the performance of the content delivery network and overcomes Internet traffic bottlenecks and heavy user traffic while delivering a variety of data and media downloads at a high speed by implementing Node.js service worker threads.
- The repository uses open standards to ensure that the content is accessible in that it can be searched and retrieved for later use. The learning media may be large, which consumes a lot of data transfer over the network and large amounts of storage disk space. To overcome this issue, a high-speed (Class 10) and high-capacity SD card is recommended for the Raspberry Pi server.

4.2 Clients

This section describes clients' devices.

Image

- The camera module allows students to capture their paperwork using the device's camera.
- The image process module provides automatic and manual image functions allowing the students to adjust and customize the captured images such as resizing, rotating, cropping, editing or adjusting the image contrast.

Content

- The display module displays content received from the server.
- The draw/edit module allows the client users (students or teachers) to edit, draw or add some comments.

File Transfer and Storage

- The background service module handles the basic functions called by the other components.
- The file transfer module handles data transmission between the clients and the servers (upload and download) using the HTTP.

5 Implementation

This section describes the implementation of the server and clients.

5.1 Raspberry Pi Server

To meet the requirements gathered in section three, the system is designed to support all features in both the Internet mode and non-Internet mode. In the Internet mode, the Raspberry Pi server is able to connect to the world through an Internet connection and gateway using Wi-Fi or Ethernet. The students are able to access the server via the Internet connection. In the non-Internet mode, the server itself is configured as an access point and a server allowing the clients' mobile devices to connect without an Internet connection.

The implementation employs client-server architecture for both modes. Node.js is installed for HTTP service including HTML/CSS/JS. MariaDB is used for relational database. Mosquito MQTT broker is used as the push notification service to trigger events on mobile devices such as reloading media, refreshing the UI and sending notifications. In this system, the Quality of Service (QoS) level 0 is exploited. This level is suitable for the trigger in the system since it provides the same guarantee as the underlying TCP protocol. The server/broker (Raspberry Pi) does not store any messages in the server.

To support two modes (Internet and non-Internet), the server needs to be configured as an access point. Thus, a Host and DNS software are required. It is essential to configure the traffic routing. We employed Hostapd and DNSmasq for host and DNS, respectively. Hostapd enables the wireless interface of the Raspberry Pi into a Wi-Fi access point called a virtual router or virtual Wi-Fi. It has features including a RADIUS authentication server for MAC address-based access control, an IEEE 802.1X authenticator and dynamic WEP keying, RADIUS accounting, WPA/WPA2 (IEEE 802.11i/RSN) authenticator and dynamic TKIP/CCMP keying.

To use the 5 GHz band, the parameter in the configuration file (`/etc/hostapd/hostapd.conf`) must be `hw_mode=a` or `hw_mode=g` for 2.4 GHz band.

In the Internet model, the students can access to the server via an internet connection. The Raspberry Pi server must be connected to the Internet either through wire (Local Area Network) or wireless (Wi-Fi/Bluetooth). This can be accomplished using No-IP, which is a dynamic DNS service that maps the dynamic IP address of the network interface on the Raspberry Pi to the sub-domain of no-ip.com. This means

that the Raspberry Pi server is globally online without a public IP. Any clients' mobiles are remotely able to access via the Internet connection.

5.2 Clients

The clients are students' mobile devices. The major functions of the devices are to receive learning media and course materials, to capture, edit or draw, and to send to the server. To support the popular mobile platforms, iOS and Android, the cross-platform mobile application development paradigm is required. In this study, we employed React Native, which is a framework using JavaScript and React.

On the display page, students are able to add annotations using a finger (touch and draw on the screen) as shown in Fig. 3. To draw on a screen, a component that holds a canvas is required. This component listens and handles events such as touchdown, move and up. Event handlers must be implemented for each event in order to respond to the touch. In each event handler, the nativeEvent is used rather than the syntheticEvent, which is provided by React Native because some properties are needed that do not exist on syntheticEvent.

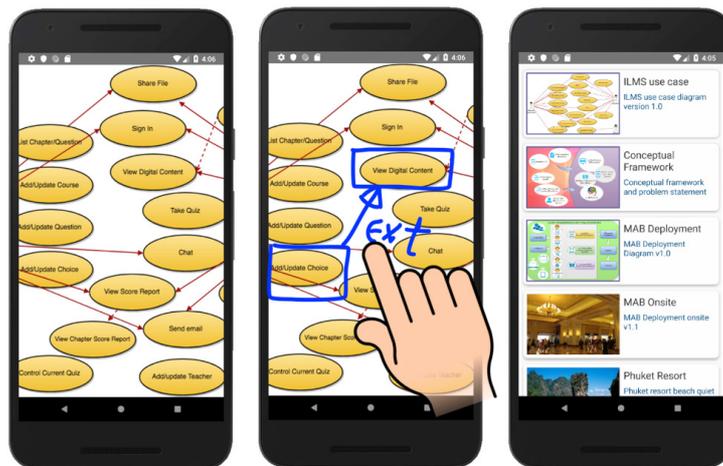


Fig. 3. View, draw/edit and broadcast to the system

The background service module includes base class, which handles the basic functions called by the other components and MQTT client service for sending notifications. The MQTT client service running on a separated thread is implemented to perform subscribe and publish messages to the MQTT broker. This service is able to perform long-running operations.

6 Scenario

Fig. 4 and 5 illustrate the usage scenario and activities respectively.

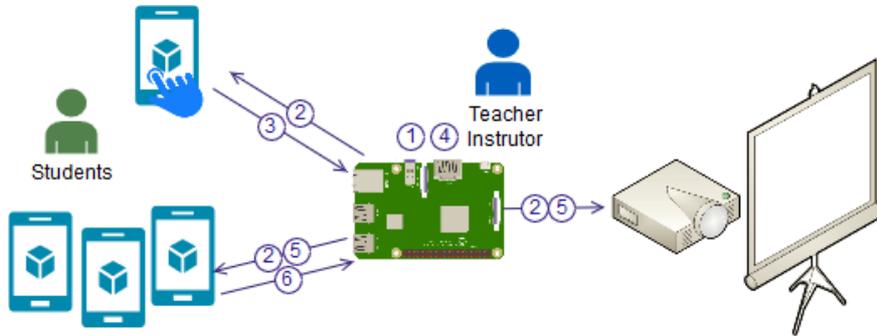


Fig. 4. Usage Scenario

- A teacher broadcasts learning media or course material to the class network through the server;
- It is pushed to all students' mobile devices and then displayed. The content is also displayed on a projection screen connected to the server;
- A student reflects his/her thinking by adding some comments or editing the content, which is then broadcast back to the server;
- The teacher opens the edited file then commits to broadcast it to the class network;
- The file is now displayed on the projection screen and pushed to all students' mobile devices;
- Other students share his/her paper work using a device's camera by capturing the work or selecting a file located in the device that is then submitted to the server. The teacher needs to commit to broadcast this file.

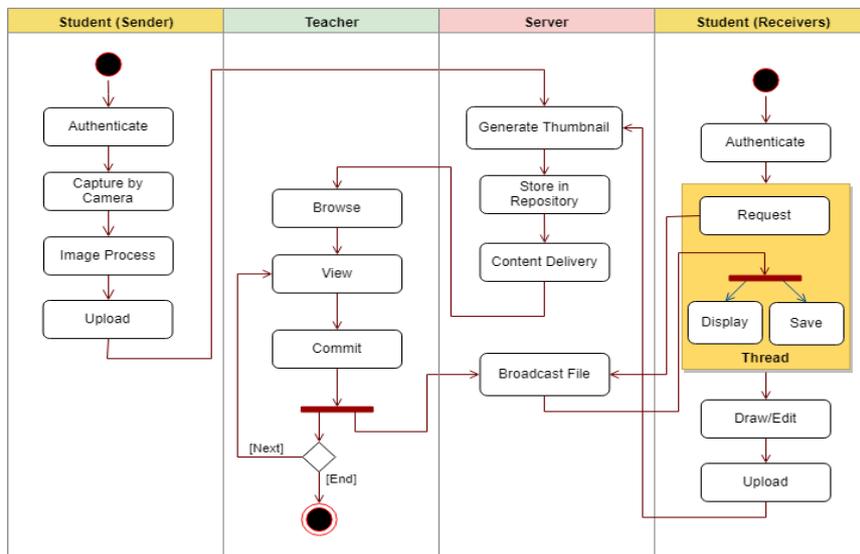


Fig. 4. Swimlane diagram

7 Results

The system was tested by the developer and then deployed in a classroom containing thirty-four students for six weeks. The evaluation was performed using a Likert-type scale questionnaire and interview. We divided the procedures into two parts. In the first part, we trained the students on how to use the system. In the second part, the teacher and students used the system. After six weeks, they evaluated the system. The evaluation had three dimensions in terms of usability, functionality and security [23]. The results revealed that all dimensions were evaluated highly. The teacher and students were highly satisfied with the system.

8 Conclusions

In this research, we developed and proposed an LMRDS for a smart classroom using IoT and mobile technologies to support active learning activities including analyzing, creating and presenting. The teacher could broadcast learning media or class materials directly to the students' mobile devices. After that, the students could interact with the media by drawing, editing or adding comments using their mobile device, and then broadcast it back to present and reflect their thinking. We separated stakeholders in terms of their concerns into teachers and students. The system included: 1) a server using a Raspberry Pi 3B+; and 2) mobile devices that belong to students. The system was designed to support full features involving two approaches in the form of an Internet model and a non-Internet model. In the Internet model, the Raspberry Pi server was able to connect to the world through an Internet connection (Wi-Fi or LAN). In the non-Internet model, the server itself acted as an access point, allowing the students' mobiles to connect directly to the server without an Internet connection. The mobile applications were implemented to support both mobile platforms including iOS and Android.

9 References

- [1] P21. (2018). Framework for 21st Century Learning," [Online] Available: <http://www.p21.org/our-work/p21-framework>. [Accessed 2017].
- [2] Jobnetwork, (2017). Top 10 Employability Skills, [Online]. Available: <http://www.opportunityjobnetwork.com>. [Accessed 2018].
- [3] Saraubon, K. and Wanapiroon P. (2014). Investigation of Augmented Book using Multimedia Learning on Mobile Device, in The Fifth TCU International e-Learning Conference 2014.
- [4] Sinlarat, P. (2012). Principle and Teaching for Undergraduate Students, Chula Books.
- [5] Chickering, A. W. and Gamson, Z. F. (1987), Seven Principles for Good Practice in Undergraduate Education, vol. 9, pp. 3-7.
- [6] Duran, L. B. and Duran, E. (2014). The 5E Instructional Model, The Science Education Review, vol. 3, no. 2, pp. 49-58.
- [7] Juwah, C., et al. (2004). Enhancing student learning through effective formative feedback," York, The Higher Education Academy.

- [8] Bidabadi, N., Isfahani, A., Rouhollahi, A. and Khalili, R. (2016). Effective Teaching Methods in Higher Education: Requirements and Barriers. *Advances in Medical Education & Professionalism*, vol. 4, no. 4, pp. 170-178.
- [9] Digregorio, P. and Sobel-Lojeski, K. (2009). The Effects of Interactive Whiteboards (IWBs) on Student Performance and Learning, *Educational Technology Systems*, vol. 38, no. 3, pp. 255-312. <https://doi.org/10.2190/ET.38.3.b>
- [10] Gosain, K., (2006) Factors Influencing the use of Interactive Whiteboard. *International Journal of Advanced Research in Education & Technology*, vol. 3, no. 3.
- [11] IoT business. (2017). IoT Business Platform," [Online]. Available: <http://iotbusiness-platform.com/iot-thailand/>. [Accessed 2018].
- [12] V. Krotov. (2017). The Internet of Things and new business opportunities, *Business Horizons*, vol. 60, no. 6, pp. 831-841.
- [13] Saraubon, K. and Wanapiroon, P. (2015). Design of Crystal-based Learning on Mobile
- [14] Augmented Book, in *The Sixth TCU International e-Learning Conference 2015: Global Trends in Digital Learning*, Bangkok.
- [15] Chin, K.-Y. and Chen, Y.-L. (2013). A Mobile Learning Support System for Ubiquitous Learning, in *The 2nd International Conference on Integrated Information*.
- [16] Raspberrypi. (2017). Raspberry Pi. [Online]. Available: <https://www.raspberrypi.org>. [Accessed 2018].
- [17] S, M. and H, S. (2017). Survey of prototyping solutions utilizing Raspberry Pi, in *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija.
- [18] Saheb, S. Y. B., Mueed, G. A., Choudhary, H., and Ansari Z. (2017). IoT Based Lecture Delivery System Using Raspberry Pi, *International Journal of Advanced Research in Computer Engineering & Technology*, vol. 6, no. 3, p. 245247.
- [19] Refsnes, D. (2013). HTML Responsive Web Design, [Online]. Available: www.w3schools.com. [Accessed 2015].
- [20] ProgressiveWebApps. (2017). Progressive Web Apps, [Online]. Available: <https://developers.google.com/web/progressive-web-apps/>.
- [21] Saraubon, K., Nilsook, P. and Wannapiroon, P. (2016). System Design of Mobile Augmented Book, *International Journal of Interactive Mobile Technologies*, vol. 10, no. 1, pp. 52-59.
- [22] Cheong, C., Bruno, V. and Cheong, F. (2012). Designing a Mobile-app-based Collaborative Learning System, *Journal of Information Technology Education*, vol. 11.
- [23] Pressman, R. S. and Maxim, B. (2015). *Software Engineering: A Practitioner's Approach* 8th Edition, 8 ed., New York: McGraw-Hill.
- [24] Liu, H. H. (2009). *Software performance and scalability: a quantitative approach*, New Jersey: John Wiley & Sons, Inc.

10 Author

Kobkiat Saraubon, Ph.D. is with the Department of Computer and Information Science, King Mongkut's University of Technology North Bangkok (KMUTNB), Thailand. He is a lecturer and researcher experienced in the professional Internet of Things, native and cross-platform mobile application development. His research interests include mobile application and game development, mobile security and the IoT.

Article submitted 29 October 2018. Resubmitted 09 January 2019. Final acceptance 10 January 2019. Final version published as submitted by the authors.