IOP Conference Series: Materials Science and Engineering

PAPER • **OPEN ACCESS**

Smart campus attempt using green transportation mobile-based tracking app

To cite this article: M Rachmaniah *et al* 2020 *IOP Conf. Ser.: Mater. Sci. Eng.* **874** 012020

Smart campus attempt using green transportation mobile-based tracking app

M Rachmaniah[1*], AI Suroso[2*], M Syukur[3], I Hermadi[4], RW Putra[4], and MRG Tirani[4]

[1] Computer Science Department, IPB University, Bogor, West Java, Indonesia
[2] School of Business, IPB University, Bogor, West Java, Indonesia
[3] Agronomy and Horticulture Department, Faculty of Agriculture, IPB University, Bogor, West Java, Indonesia
[4] Computer Science Department, Faculty of Mathematics and Natural Sciences, IPB University, Bogor, West Java, Indonesia

meuthiara@apps.ipb.ac.id; arifimamsuroso@apps.ipb.ac.id

Abstract. Green campus is the creation of a green ecologically friendly environment that leads to smart campus establishment. In this study, green campus status is revealed by applying cross-campus community collaboration under green transportation bus. The purpose of this study was to develop a real-time Android-based tracking system, using the V model approach, aimed at green transportation movements and accessibility. The global positioning system of green transportation is interconnected via a back-end server that utilizing web-socket to incorporate client and server communications. The results are passenger app that enabling the passenger to track green transportation movement, acquire green transportation meta-information, perceive campus routes, and send awaiting notification. Further results are driver app for green transportation drivers to choose which vehicle to drive, change driving status, recognize other active coaches, locate nearer passengers stops, and identify campus routes. The created mobile app has successfully experimented on Android smartphones such as Samsung Galaxy Mega J5, Lenovo A6000, Xiaomi Redmi 3, Asus 2 Zenfone Laser, and Samsung J7 Prime.

1. Introduction

Self and information services and environment platforms are the three main characteristics of a smart campus. The smart campus is an outcome of the rapid development of GIS and computer network technology, and the internet of things (IoT) [1]. A smart campus is a campus that can adapt to every new situation that occurs in daily operations [2] and of three campus pillars that are administration, students, and lecturers [3]. The smart campus comprises of six components: iLearning, iGovernance, iGreen, iHealth, iSocial, and iManagement [4]. The combination of green campus and smart campus brings out the concept of smart green campus. The involvement of all campus communities in employing the existing resources and technology effectively and environmentally is an essential point of this concept that has a significant impact on campus sustainability [5]. In 2007 a study at Tongji University successfully applied green campus employing green concepts, ecological technologies, energy, and resource development practices that became a model of cost-effective campus expenditure [6]. Another study on proactive environmental management creates a healthy learning environment, build environmental awareness, and reduce campus operational costs [7]. Sustainable green transport is at the heart of the green transportation concept [8]. In 2014, IPB University proclaimed the green campus concept embracing five main programs: Green Transportation, Green Movement, Green Space, Green Building, and Green Energy. In 2015 the gas-fueled green transportation buses became the main program in priority. A total of five buses at Dramaga campus serves five different routes to support the mobility of the campus community in the 267-ha area. Along the bus routes, there are 19 stops, with an average of 1,918 passengers per day. Each green transportation bus has a departure schedule, bus stop arrival, and a mobile phone. However, in its implementation, schedules and routes were often ignored by bus drivers and passengers who demanded a different course. Therefore, real-time campus bus locations on its routes or bus stops were the main focus of this study.

Many researchers conducted studies on Android-based vehicle tracking systems using a global positioning system (GPS). One study provided current and offline information on bus, bus number, and bus routes, including estimate time needed to reach the destination [9]. Another study provided campus bus locations, bus details, driver details, stop location, and bus routes for students using Google Maps [10]. However, both studies demanding its user to login and entered the bus license plate. Further research gave an enhanced level of reliability in the navigation solutions of GPS-denied environments [11]. Another study sent their data in the form of latitude and longitude coordinates through short messages on the user mobile wherein the coordinates will be further plotted in the Android app automatically [12]. GPS satellites continuously transmit digital radio signals that carry satellite location data and time toward the corresponding recipient. The GPS composes of three segments, user segment (cell phones and personal digital assistants-PDAs), space segment (collection of satellites), and control segment [13]. The atomic clock that has punctuality of one-millionth of a second equipped the GPS satellites. Based on this information, the receiving station knows how long it takes to send signals to the recipient on earth. The longer time to get to the receiver, the bigger the position of the satellite from the receiving station [14].

WebSocket first emerged in HTML5, allowing web server and web browser real-time communication. The server responds according to the request sent by the client. When a connection does establish, the initial HTTP-based communication changes into full-duplex communication [15]. A client is a subscriber when the client connects the server through a web-socket connection and receives data from the server while the link lasts. This connection is an ideal solution for real-time data exchange [16]. Google Maps is a server-side, which means that maps are stored on Google servers but use by users. The Google Maps API is one of the Google services that gives access to Google maps and modifies the map as needed on the client side [17]. After the map loads, interactions and modifications are happening. Adjustments can be the addition of markers, lines, or polygons.

This study developed real-time tracking apps for green transportation movement as an effort to demonstrate smart green campus concepts using an Android-based smartphone. The developed apps are passenger app for the campus community and driver app for green transportation driver. The apps development applies for mobile devices because around 63.1 million Indonesians are smartphone usage [18]. The apps development utilizes Global Positioning System (GPS) technology and Google Maps as a viewer of the bus routes. Also, WebSocket, especially SocketIO, facilitates a real-time communication system. The novelty in this study is that in addition to the passenger app, we also developed the driver app. Each mobile app has no requirement to login and key in the license plate because all passengers and driver's data are available in the database. Lastly, the tracking of every active green transportation is displayed on a single page, meaning passengers do not need to swap among pages.

2. Methods

The study records the year 2018 longitude and latitude data of the Dramaga campus (**Table 1**). The data include bus stops locations, real-time bus positions, bus routes, passengers' locations at specified bus stops, bus information, number of passengers at each bus stop, and map of campus. Real-time coordinates of longitude and latitude were gathered using Google Maps and GPS features of the Android smartphone. The green transportation buses and prospective passengers on the specified bus stop location coordinate data were further transmitted to the server and used by the driver and passenger apps accordingly. Bus stop latitude and longitude positions were measured using a smartphone's GPS and Google Maps. Dramaga campus has 15 actives out of 19 available bus stops at an area of 267 ha. The departure route starts consecutively from bus stop 1 - Graha Widya Wisuda to bus stop 15 – FEM/FEMA (**Figure 1**). The driver app will capture data on the number of passengers waiting at the bus stop location in real-time mode. There are seven green transportation buses maneuvers at the campus. Five buses are in operation at 06.00 - 18.00 hours and two buses are set aside for backups.

Table 1 Longitude and Latitude of Campus Bus Stop

Bus Stop Location	Longitude	Latitude
1. Graha Widya Wisuda	106.730421	-6.560520
2. Berlin	106.730901	-6.558936
3. FMIPA	106.731424	-6.557861
4. Asrama Putri	106.731583	-6.556086
5. Tanoto Forest building	106.730604	-6.555764
6. Menwa	106.729613	-6.555749
7. FAHUTAN	106.727786	-6.555795
8. International Dormitory	106.726983	-6.556112
9. Alhurriyyah Mosque	106.725555	-6.556588
10. GORLAM	106.724446	-6.556368
11. FPIK	106.723648	-6.556632
12. FKH	106.721843	-6.556785
13. Techno park	106.726804	-6.557457
14. LSI	106.726857	-6.558434
15. FEM/FEMA	106.728191	-6.559750

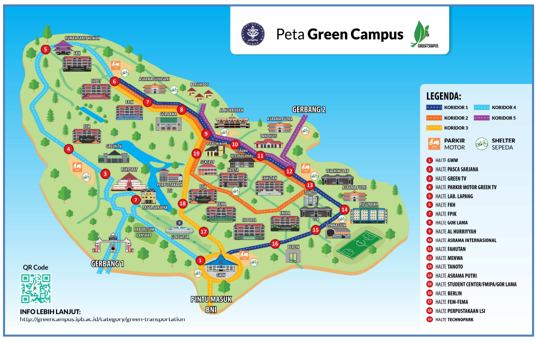


Figure 1 Campus map of green transportation route and its 19 bus stops.

The V-model development method is used to guide the development of passenger and driver mobile apps. Each step of the development stage followed by the interrelated test scenarios and its test attainment (**figure 2.a**). Here, the green transportation managers examined the test scenarios as an assurance that the system follows the development plan. The mobile apps development started with the analysis stage, where we collect qualitative data through interviewing the green transportation manager, intended to gather needs analysis of the green transportation stakeholders' requirements, and the creation of an acceptance test scenario. The results of this phase were to use case diagrams, use case descriptions, activity diagrams, and class diagrams. The design stage utilized mockups design that follows the class diagrams of the analysis stage. During the mockups design, we created unit testing, integration testing, and system testing scenarios. The implementation phase or coding comprised of two parts that are client-side and the server-side. Client-side coding was done using Android Studio and Java programming language. Server-side was done using the JavaScript programming language equipped with Express Js framework for the development of the back-end application programming interface (API).

Further, the unit testing, integration testing, system testing, and acceptance testing were carried out using developed test scenarios of the previous stages. Unit testing aimed at verifying the smallest unit of software design. Integration testing confirmed the relationship between one function to other functions to produce the desired output without any errors. System testing examined the server in the real environment using various types and versions of Android smartphones available at the current market. Finally, acceptance testing aimed to determine whether the application developed was following the original plan.

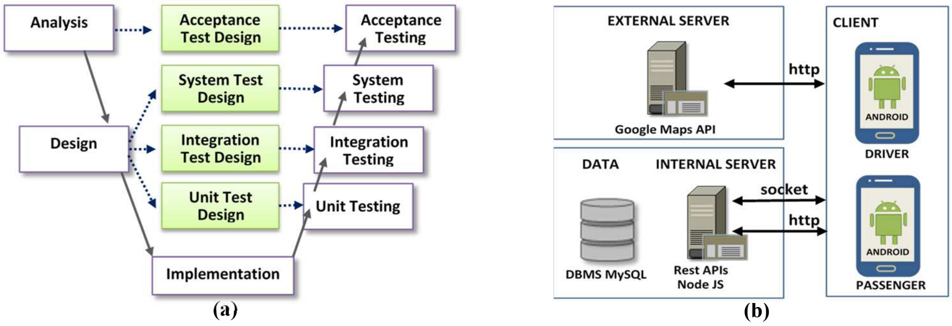


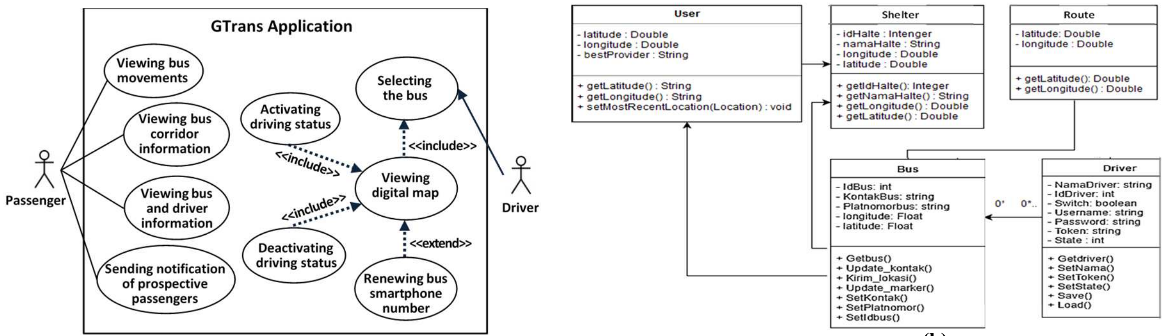
Figure 2 (a) The V-Model stages method and its related testing [19] and **(b)** Mobile apps architecture consisting of Google Maps API within the external server, data and internal server for Rest API and Node Js, and Android driver and passenger clients.

The application architecture consists of three parts, namely the client, server (internal and external), and data (**figure 2.b**). On the client-side, we built two separate apps, each for the passenger and the driver. The client-side was made using Android Studio with Java programming language. On data and internal server-side, we utilize MySQL DBMS, develop back-end applications using JS Node and JS Express framework, and employ the SocketIO library. The Google Maps API resides on the external server is used by the client to load maps and objects on maps such as bus stops to send and obtain bus and clients location. Communication between the client and the internal server consists of two arrangements of connection, i.e., through the socket and the REST API. The REST API is used to connect the client and the database contained in the server. The server sends the client request data to the REST API. The use of Sockets is a bridge for data exchange, especially location data in real-time between the clients and the server.

3. Results and Discussion

3.1 Passenger and Driver Apps Development Stages

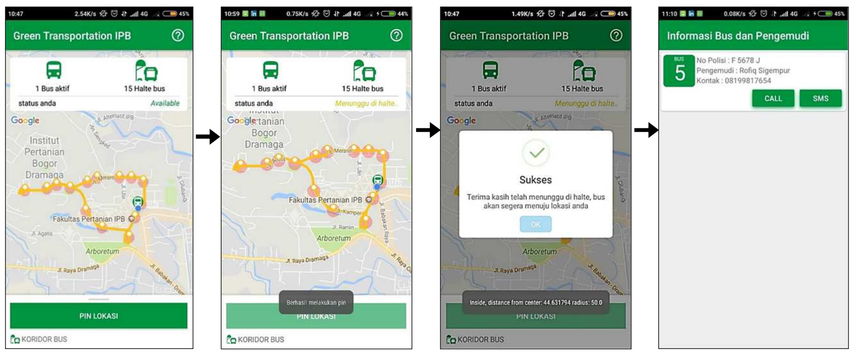
The number of drivers during daily duty is seven people; each driver drives any bus to the nearest bus pool entrance. Each green transportation bus comes with an Android-based smartphone, meaning that the smartphone attaches to the bus, not to the driver. In this study, using passenger app, the passenger views bus position, and maneuver views information of the bus and the driver, view bus route information, and send their waiting notification using personal identification number (PIN) location. Meanwhile, using the driver app, the drivers choose the bus at the nearest bus pool entrance, activate driving status, view the digital map of other active buses, view the number of passengers waiting at a specific bus stop, and renew bus smartphone number. The entire design illustrated in a use case diagram, as depicted in **figure 3.a**. Following that, we created class diagrams to explain how the system works (**figure 3.b**). Also, we conducted analyses of the workings or activities of the functions contained within the use case diagram by creating activity diagrams and examined acceptance test scenarios. Approved test scenarios for the passenger side applied for real-time bus monitoring and pin location notification sent by the passenger. To enable notification, passengers should be within 25 meters radius of a specific bus stop distance. Approved test scenarios for the driver side implemented during login, bus selection, status activation, bus contacts update, and logout activities.



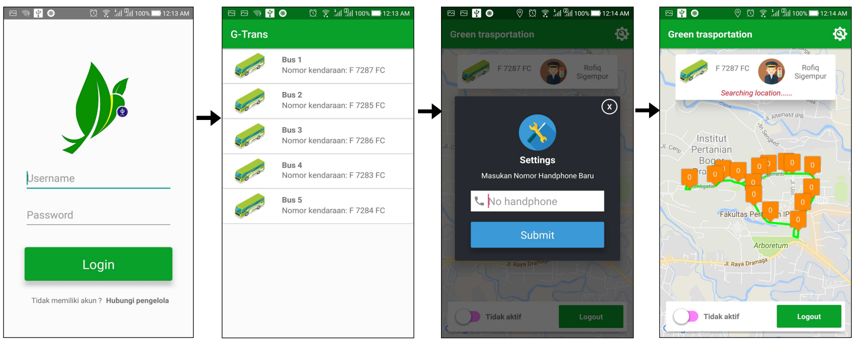
(a) (b)

Figure 3 (a) Use case diagram with passenger and driver actors and **(b)** Class diagram of green transportation tracking app.

The design stage presented in the form of mockups utilizing extensible markup language (XML) of the Android Studio, which gave an overview of the appearances and functions of the app. During the design stage, we also created test scenarios for unit tests, integration tests, and system tests of passenger and driver prototypes. Based on the analysis and stakeholder tests approval, passenger app consisted of the main page, the help page, the driver and bus information page, splash screen pages, and a pop-up message notification in the middle of the page regarding their PIN pointed to their current location (**figure 4.a**). Meanwhile, the driver app comprises of three activities: login activity, select bus activity, and maps activity (**figure 4.b**). Approved system testing scenario for both apps apply on a various Android operating system such as Jelly Bean, Android KitKat, Android Lollipop, Android Marshmallow, and Android Nougat.



(a) Passenger App



(b) Driver App

Figure 4 XML design for **(a)** use of passenger app at a bus stop, send PIN notification, receive pop up incoming bus message, receive bus and driver data; **(b)** Driver key in driver's username and password, select nearest entrance bus number, key in bus's smartphone number, and activate the bus to maneuver.

The implementation stage of this study was carried out for the driver and the passenger apps. Concerning coding for the passenger app, we provide a socket in the server to detect the connection requested by the client. Here we make client socket to be associated with the socket server. Hence, the socket on the client serves to establish the initialization of connection to the socket on the server. Following this, real-time location data exchange is obtained by initializing the event with the method called "lokasi" on the socket server or client (**figure 5.a**). Passenger app encompasses several functions, i.e., function for a passenger to receive bus location longitudinal data (**figure 5.b**), function to visualize bus movement on passenger map, and passenger notification waiting at a specific bus stop. Function on visualization bus movement utilize a particular marker bus appeared on the app/screen. However, if the marker bus is not established, then the marker bus is displayed. Also, passenger notification employed a socket on the server that received the data and added a counter to the number of prospective passengers based on the bus stop id.

Concerning coding for the driver app, we created the map function. This function comprises of three activities, i.e., route and bus stops information, update bus contact number, and change driver status. Route data and bus stops are visualized using Google Maps API. Here, we make a WebSocket connection to receive information on the number of passengers waiting at each bus stop (**figure 5.c**) and the location of other active buses to be further projected into a digital map. The passenger app employs REST API so that clients receive data from the database. This process is performed on the bus and driver information viewing function. The implementation utilizes the GET method by combining data from the driver, bus, and driver bus tables. The resulted data in the "status" attribute in the driver bus table is valued "1". The output results are data in JSON format with two objects, namely results and data (**figure 5.d**). Also, we make popup input forms to store new contact numbers and a switch button to change the status of the driver. When driver status is active, the app pops up active status and send driver location to the server via a WebSocket connection — location data sending occurs when the smartphone finds a new location. Location data send via a WebSocket connection is immediately displayed and not stored in the database, so that server performance is faster. When the driver's status is deactivated, then the status was inactive, and hence, the app stops sending the location.

```
public Socket mSocket;
{
    try {
        mSocket= IO.socket(SERVER)}
    catch(URISyntaxException e){
        Log.e("error","on create :
"+e.toString());}
}
a
@Override
protected void onCreate(Bundle
savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_maps);

    mSocket.connect();
    mSocket.on("lokasi",handlingLocationBus);
```

(a)

```
private Emitter.Listener handlingLocationBus = new
Emitter.Listener() {
    @Override
    public void call(final Object... args) {
        JSONObject data = (JSONObject)args[0];
        String id = new String();
        String longitude= new String();
        String latitude = new String();
        try {
            id= data.getString("id");
            longitude= data.getString("longitude");
            latitude = data.getString("latitude");
        } catch (JSONException e) {
            e.printStackTrace();
        }
        final String dataID = id;final String
dataLongitude = longitude;
        final String dataLatitude = latitude;
        runOnUiThread(new Runnable() {
            @Override
            public void run() {
      movingBus(dataID,dataLongitude,dataLatitude);
            } }); }};
```

(b)

```
private Emitter.Listener handling_halte_awal = new Emitter.Listener() {
    @Override
    public void call(final Object... args) {

        JSONArray array = (JSONArray) args[0];
        String paket_id = new String();
        int ix=0;
        for (; ix < array.length();ix++){
            try {
                final String count = ""+array.getInt(ix);
                final String ix_string = ""+ix;
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        update_halte(ix_string,count);

                    }
                });
            } catch (JSONException e) {
                e.printStackTrace();
            }
```

(c)

```
{
    "results": 200,
    "data": [
        {
            "kontak_bus": "081223991535",
            "id_driver": 1,
            "nama_driver": "Rofiq Sigempur",
            "id_bus": 1,
            "nomor_bus": 1,
            "platnomor_bus": "F 7287 FC"
        }
]}
```

(d)

Figure 5 (a) Client socket initialization to enable real time location data exchange; **(b)** passenger receive screen visualization of driver's location through the JavaScript Object Notation (JSON); **(c)** segment code to display the number of passengers at the bus stop; and **(d)** REST API output results in JSON format.

3.2 Implementation Tests

During the unit test, we tested the output correctness of each function enclosed within the developed passenger and driver apps using the black box method. All test results successfully passed. On integration tests, we carried out with successful results using the black box method. Integration tests aimed to assess and ensure functions within the two developed apps to have interacted correctly. On system tests, we utilized requirement testing methods to ensure that the developed apps work well on the real environment using different Android-based operating systems. Here, the developed passenger and driver apps are successfully functioning on five different Android-based mobile devices and operating systems (**table 2**).

Table 2 Successful compatibility test using different mobile devices and operating systems

No	Mobile Device	Operating System	Screen Size (inch)	Resolution (pixels)	Test Result
1.	Samsung Galaxy Mega J5	Android Jelly Bean	6	720x1,280	Successful
2.	Lenovo A6000	Android KitKat	5	720x1,280	Successful
3.	Xiaomi Redmi 3	Android Lollipop	4,7	720x1,280	Successful
4.	Asus 2 Zenfone Laser	Android Marshmallow	5,5	720x1,280	Successful
5.	Samsung J7 Prime	Android Nougat	5,5	1,080x1,920	Successful

The passenger acceptance tests were carried out using alpha testing techniques. Here, stakeholders tested the apps by executing each test scenario agreed previously during test design. The stakeholder acted as a passenger and tested real-time monitoring function of the green transportation bus and notification of potential passengers sent using the location pin. Remind that passengers' location cannot exceed the 25-meter radius of a specific bus stop. Further, stakeholders acted as the bus driver logged in and chose a particular bus to drive, activated driving status, maneuvered to verify longitudinal location changed and sent to the passenger app, and updated the bus contacts and logged out. At the end of the test, stakeholders agreed and accepted the passenger and the driver apps developed.

3.3 Study Review

Tens of thousands of undergraduate and graduate students, professors, and visitors are present in the campus doing activities that spread over classrooms, in between class schedules, research and education labs, and offices over a reasonably large area. Some sites are within walking distance, but other far distant sites need green transportation, especially for moving class over a limited duration in-between class. Green transportation passengers are very dependent on the presence and timeliness of the bus arriving at the bus stop in order to get to the destination promptly. The passenger app is beneficial for green transportation users to find out the arrival of the bus at a bus stop. In this case, passengers can track the distance of the bus from where they are waiting at the bus stop. The driver app is beneficial for drivers to select any bus parked near the bus garage entrance. It is thus speeding up the work process. In the driver app, there is a notification of the number of passengers waiting at a bus stop. In this way, the bus driver is triggered to go to the target bus stop immediately. More and more passengers waiting at a bus stop will encourage bus drivers to maneuver to the specified bus stop instantaneously.

The limitation of this study is that it only covers a small part of the many elements contained in the smart campus. More elements for the smart campus are available; these are smart inventory, smart parking, automated street light, global systems for mobile communication-based alerts, air quality or noise level, light intensity monitoring system, garbage or waste overflow alert system, and garden automation water flows [20]. Nevertheless, the utilization of GPS technology with Google Maps is expected to provide significant implications in institute/college for the classy environment and less pollution as an endorsement of campus activities that are in line with the smart green campus approach. The concept of smart green campus links green transportation and the Internet of Things (IoT) of GPS features of the Android-based mobile devices. We believed that the majority of campus pupils have at least one mobile device. Green transportation, i.e., gas-fuelled buses, becomes essential to conserve the environment if we compare to gasoline-fuelled buses effects.

4. Conclusion

With the help of green transportation apps, the campus community can navigate the bus movement maneuvering surrounding campus bus stops within the 267-ha area. This navigation is supported by the back-end application that enables the server to receive/send real-time data from the client-side (passenger and driver). The campus community can send PINs location to notify their position to the nearest bus in operation. Multiple driver app features are available, which allow the driver to choose the available bus to ride, find out the number of passengers waiting at the bus stop, and sending its location to passengers. The successful use of green transportation apps on five different Android-based operating systems proven that the study help improves the utilization of campus transportation efficiently and hence supporting the green campus concept.

References

[1] Xue R., Wang J., Chen L., Using the IOT to construct a ubiquitous learning environment, Proceedings of 2011 Second International Conference on Mechanic Automation and Control Engineering. p. 7878-7880, Jul. 2011.
[2] Kwok L.F., A vision for the development of i-campus, Smart Learning Environments-a Springer Open Journal, 2(2):1-3.
[3] Kar A.K., Gupta M.P., How to Make a Smart Campus – Smart Campus Programme in IIT, Report, Indian Institute of Technology Delhi, 2015,
[4] Ng J W.P., Azarmi N., Leida M., Saffre F., Afzal A., and Yoo P.D., The intelligent campus (iCampus): End-to-end learning lifecycle of a knowledge ecosystem, Proc. 6th International Conference on Intelligent Environments, Kuala Lumpur Malaysia, p. 332–337, Dec. 2010.
[5] Aion N., Helmandollar L., Wang M. dan Ng J.W.P., Intelligent campus (iCampus) impact study, Proc. of the 2012 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops, Macau, China; p. 291–295, Dec. 2012.
[6] Tan H., Chen S., Shi Q., Wang L., Development of green campus in China. Journal of Cleaner Production, (64), p. 646-653. Feb. 2014
[7] Sahoo M.K., Mishra S.K., Green Campus – A Competitive Advantage and Sustainability for Management Institutions. DRIEM Business Review. 1(1):74-80, 2014
[8] Andriani D.M., Yuliastuti N., Assessment of transportation systems that lead to green transportation in the city of Surakarta (Penilaian sistem transportasi yang mengarah pada green transportation di Kota Surakarta), Journal of Regional and City Development (Jurnal Pembangunan Wilayah dan Kota, 9(2), p. 183-193, 2013.
[9] Jemilda G., Bala R.K., Johnson B., Linga G., Mobile application for college bus tracking, International Journal of Computer Science and Mobile Computing, 4(3), p. 500-507. Mar. 2015.
[10] Supriya S., Pooja S., Monika Z., Roshni J., Sonekar S.V., Real-time college bus tracking application for Android smartphone, International Journal of Engineering and Computer Science, 6(2), p. 20281-20284. 2015.
[11] Namdev D. and Pal Saroha G., Android Based Integrated Vehicle Navigation System, International Journal of Electronics and Communication Engineering and Technology (IJECET), 8(4), p. 1–6, July-August 2017.
[12] Saini J., Agarwal M., Gupta A., Manjula R., Android App Based Vehicle Tracking Using GPS And GSM, International Journal of Scientific & Technology Research, 6(09), p. 53-58, Sep. 2017.
[13] Moses K.B., Dada O.A., Akinwomni F.C., GSM Base Stations Location Monitoring using Geographical Information System, International Journal Computer Network and Information Security, (8), p. 39-45.
[14] Abidin H.Z., Determination of Position Using GPS and its Application (Penentuan Posisi Dengan Menggunakan GPS dan Aplikasinya). Jakarta(ID): PT Pradyan Paramita, 2007

[15] Skvorc D, M Horvat, S Srbljic. 2014. Performance Evaluation of WebSocket Protocol for Implementation of Full-Duplex Web Streams. Information and Communication Technology, Electronics and Microelectronics; 2014 May 26-30; Opatija, Croatia. p. 1003-1008.
[16] Wessels A., Mike P., Jahrain J., Syed R., Remote Data Visualization through WebSockets. 2011 Eighth International Conference on Information Technology: New Generations; p. 1050-1051, Apr. 2011.
[17] Sunyoto A., Overview: Google Maps API V3, Undergraduate dissertation, College of Information and Computer Management (Sekolah Tinggi Manajemen Informatika dan Komputer), Yogyakarta, 2010.
[18] Indonesian Internet Service Provider Association (Asosiasi Penyelenggara Jasa Internet Indonesia), 2016 Indonesian Internet User Profile (Profil Pengguna Internet Indonesia 2016).
[19] Dennis A., Wixom, Barbara H., System Analysis and Design, ed. 5, New Jersey (US): Wiley Inc. 2012.
[20] Shivaraj Kumar TH, Sriraksha TA, and Noor U Saba. An IoT Based Secured Smart e-Campus. International Journal of Humanities and Social Science Invention. Vol 6(3), p.88-93