A Bus Tracking Information System using Consumer Grade GPS: A Case Study

Shamsul Kamal Ahmad Khalid, Nurul Syahadah Mohd Salleh, Noor Azah Samsudin
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat Johor, Malaysia.
shamsulk@uthm.edu.my

***Abstract*—A real-time tracking of public bus movement may help passengers plan their travel and reduce stress due to uncertainty. Several bus tracking information systems have been proposed, but costly to implement and also take longer time to setup. In this paper, a bus tracking information system using standard mobile Internet connection and consumer grade GPS is proposed, called UBAIS (University Bus Arrival Information System). The system will continuously update the estimated arrival time based on the actual location of the buses in the route leading to the next bus stop. With graphical information provided to the passenger via a mobile phone, passengers will be able to view the actual location of the buses in the route. A prototype for Universiti Tun Hussein Onn Malaysia's bus service has been implemented – consisting of buses equipped with Android devices that will continuously send current GPS locations of the buses to a centralized server and read by clients (fixed or mobile) with various update intervals. Several testing has been performed on the current prototype and it demonstrates good accuracy and increased satisfaction among passengers.**

***Index Terms*—Bus Tracking Information System; GPS Tracking System; Smartphone Applications.**

I. INTRODUCTION

Public transport (such as a bus service) commonly subjected to various real time problems that inhibit them from conforming to a scheduled arrival and departure time. Consequently, passengers may have to wait longer than expected, or miss a bus due to full occupancy. They may experience stress due to these non-conformance [1][2]. To further understand these circumstances, an investigation has been conducted on a bus service at Universiti Tun Hussein Onn, Johor Malaysia (UTHM).

Sikun Jaya Holding is one of UTHM contractors that offer transit bus services to ferry students from main campus to Residential College of Taman Universiti (KKTU) and vice versa. Sikun Jaya provides eight transit buses that stop frequently at a bus stop every 10 minutes. A bus will normally take around 25 minutes for one complete loop from KKTU-UTHM-KKTU. Students are informed about a fixed bus schedule movement pasted on notice board at each block in Residential College of Taman Universiti (KKTU), and also at all residential colleges inside the main campus.

The following problems have been observed:

i. Transit buses commonly arrive late from its scheduled time or earlier than expected.
ii. Sometimes more than one transit buses arrive simultaneously. When many buses arrive simultaneously at one bus stop, the next bus will have a wide gap of departure time. For example, it might exceed 30 minutes after the previous bus departure. This is sometimes referred to as bus bunching [3].
iii. Difficulty for students to arrange their own schedule to wait at the bus stop.

A solution to reduce the above problem is to provide real time information to the passengers so that they are aware of the bus movement.

The rest of this paper is organized as follows. Section II reviews existing bus tracking information systems and presents smart phone and GPS technology. The detailed descriptions of UBAIS implementation framework and methodology are presented in Section III. Section IV presents the results and discussion. Finally, conclusion and future work are presented in Section V.

II. LITERATURE REVIEW

A. Existing Bus Tracking Systems

Several solutions have been proposed to help user get more updated information about public bus services, such as CTA Bus Tracker [4], Waitless Bus Tracking System [5] and IIT Madras Bus Tracking System [6]. CTA bus tracking system is a commercial system developed for Chicago public transport. Both Waitless Bus Tracking System and IIT Madras Bus Tracking System are university campus bus services that have been developed for experimental/research purpose.

The CTA Bus Tracker allows different platforms to access bus location data. It is a complete bus tracking information system developed over twelve years duration and gets positive response from its customer. Although a very expensive investment, a research found that there has been an increase in the number of ridership since the installation of the CTA Tracker system [4]. The Waitless system consists of a device with several LED lights positioned on a map of bus routes to indicate the location of buses on campus. The Waitless bus tracking device is a standalone system that displays the real-time location(s) of the buses on Georgia Tech's campus. This system, designed to be deployed at various bus stops around campus, is comprised of a solar power source, a battery, a microprocessor, LEDs, and a wireless internet link. The wireless internet link will be used to poll a live XML feed from the NextBus server (via GTwireless) that contains GPS data of each bus's location. The data will then be parsed by a microprocessor and used to illuminate tri-color LEDs that will represent each bus's location [5]. The prototype cost around US7000 and if it were to be installed at all bus stations of the university, it will be estimated to cost around US40000. No further mobile application has been developed yet to replace the client side of the system. Meanwhile, IIT Madras Bus Tracking System [6] was developed using JAVA and supported web-based and android platform which is an advantage since user can view through a website or smart phone. It is quite similar to the CTA Tracker system.

All of the above systems relied on industrial grade GPS system installed on each bus and transmitted to a central server via a dedicated channel. The proposed system, however, investigates the possibility of a bus tracking system running over regular public wireless broadband connection using consumer grade GPS exist in many smartphones. It also aim at ease of installation, where a bus service provider can immediately install and launch a near real time bus tracking service using available phones and a web server. Table 1 shows the comparison of industrial grade GPS bus tracking systems with the consumer grade GPS bus tracking system.

Table 1
Comparison of existing systems and the proposed system

	CTA Bus Tracker	Waitless Bus Tracking System	IIT Madras Bus Tracking System	UBAIS using standard smartphone GPS
Language use	JAVA	Arduino	JAVA	JAVA
Platform	Web based SMS	Website LED Display at bus stop	Web based Android	Android[7]
Route selection	Yes	No	No	Yes
Bus stop selection	Yes	No	No	Yes
Costing	High	Medium	-	Low
Availability	Fixed	Fixed	Fixed	Fixed
City	Chicago	Georgia	Madras	Parit Raja

B. GPS

The Global Positioning System (GPS) is a satellite navigation system that provides location and time information in virtually all weather conditions [8][9]. GPS consist of 3 segments: the first segment is space segment where there are 28 satellite orbits earth and each satellite orbits 11,000 nautical miles above the earth. The second segment is the user segment consisting of receivers by which user hold and can be tracked by the GPS. In UBAIS, driver installs a mobile client program so that the bus can be tracked. The third segment is the control segment consisting of ground stations that make sure the satellites are working properly [10].

A GPS services provided in an Android smartphone consists of 5 components where every segment play important role to Android to work properly. The components built into the mobile GPS are: GPS Chip, GPS Driver, GL Engine, Android Location Services and User Applications [11].

III. Methodology

A. Framework

Figure 1 shows the implementation framework of UBAIS. User will access UBAIS on a smart phone with Wi-Fi connection or some data plan. Once a user login into the system, a map is shown with all the buses moving from one bus stop to another. The user will immediately know current locations of the buses. The locations are configurable to be updated at every 2-15 seconds interval. The main server periodically gets latest locations of the buses from the Android phones installed on the buses. Each of these phones runs a driver's application.

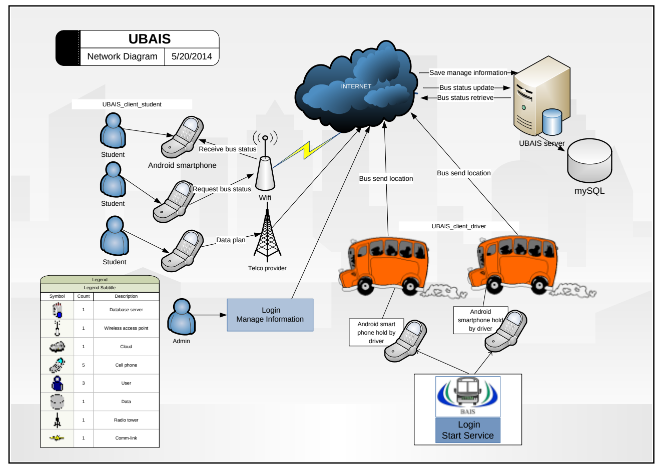

Figure 1: The UBAIS Framework

B. Analysis and Design

The requirements of UBAIS have been established via interviews and observation carried out with students and UTHM staffs. The design process and procedures will be illustrated in the following sections. To elaborate the main process of the system, Activity Diagram (AD) is used. In this AD, four main processes are discussed. The processes are login admin, driver, start and stop services. It also provides the functionality to add new drivers, new routes and new buses into the system. A bus movement view process is also created.

i. Use Case Diagram

Figure 2 shows the Use Case Diagram for UBAIS. There are three types of actors: user, admin, and driver. Each actor has its own function. In this case study, the users are UTHM students as they are the primary groups using the transit bus service. With user privilege, they can view maps of their favorite routes. The driver's application provides functions to start and stop services. For security purpose, the driver is required to login into UBAIS from his/her mobile application. Once a driver successfully login into UBAIS, the client program will immediately sends its locations to the UBAIS server on a regular interval. Administrator is the last actor. The administrator (also known as 'admin') is given two roles: login and manage information. An administrator is given a privilege to manage information related to UBAIS system including managing new drivers, updating information and displaying list of drivers. Beside drivers, an administrator can also manage busses and routes. Only registered drivers are able to login into UBAIS and only registered buses are traceable by the UBAIS.

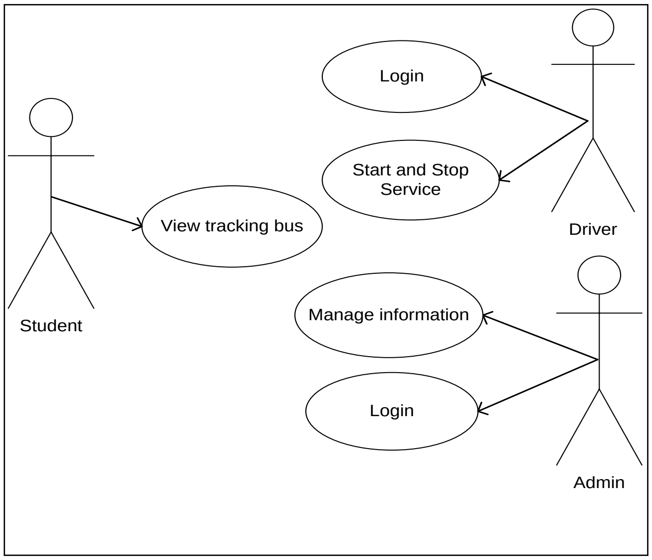

Figure 2: UBAIS use case diagram

ii. Sequence Diagram

Sequence diagram is used to define event sequences, closed to the desired outcomes. Figure 3 shows the start and stop services sequence diagram. Driver will **sent getservice()** to system, the system will display **get_service.form()** and retrieve service by **getservice_details()** and these details are displayed to the driver.

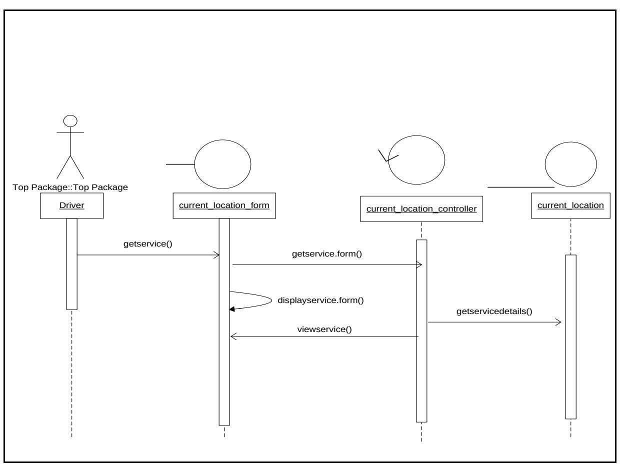

Figure 3: Driver start and stop service sequence diagram

Figure 4 shows a manage information. The system **getbusdetails()** from **form_display()** and **bus_controller()** will send information to the database based on the method chosen using either **addbusdetail()**, **updatebusdetail()**, **deletebusdetail** or **viewbusdetail()**.

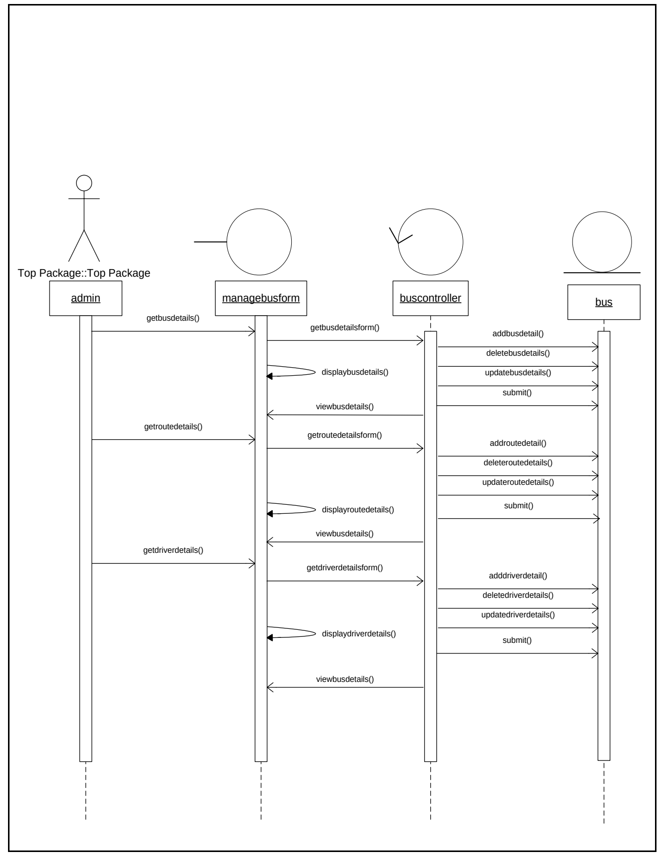

Figure 4: Admin manage information sequence diagram

iii. Class Diagram

For database support, a class diagram with five entities has been defined - admin, driver, route, bus and current location. As stated earlier, the admin manages routes, buses and drivers. The driver's phone will be responsible to update current locations of the buses.

C. The Estimated Time of Arrival (ETA)

The ETA is calculated based on the following equation:

$$T_b = T_a + \Delta t_{ab}$$

where:

T_b is the ETA
T_a is the current time at location *a*
Δt_{ab} is the estimated time needed to reach from *a* to *b*

The Δt_{ab} is calculated as follow:

$$\Delta t_{ab} = (d_b - d_a) / v_{ab}$$

where:

d_b is the road distance of bus stop *b* from bus stop 1
d_a is the road distance of bus stop *a* from bus stop 1
v_{ab} is the average velocity of buses from bus stop *a* to *b*

IV. Results And Discussion

UBAIS has been developed using Adobe Dreamweaver, PHP, Java script, mySQL database, ADT eclipse, a robust Apache web server and several android smart phones.

A. Show Map

Figure 5 shows the map interface to display bus locations from the user client side application. UTHM main campus map is overlaid with current map from Google map. Lines of bus routes are highlighted using blue line to make user easy to view the bus route. In this diagram, two buses with identifiable markers were shown on the map.

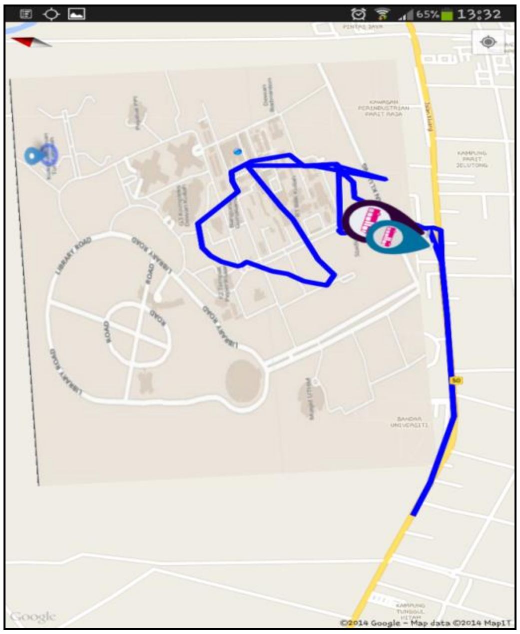

Figure 5: Display map with bus markers

Figure 6 shows the process to display map with bus markers. The map is initialized by reading and zooming to the longitude and latitude defined in the **mapCenter.**

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    try {
        // Loading map
        initilizeMap();

        GoogleMap map = ((MapFragment) getFragmentManager()
                .findFragmentById(R.id.map)).getMap();

        mapCenter = new LatLng(1.8482290, 103.0715070);

        map.moveCamera(CameraUpdateFactory.newLatLngZoom(mapCenter, 25));

        setUpMap();

        busroute.setRoute();
        BUSLOCATION = 0;
```

Figure 6: Process to initialize the display map

B. Display Bus Detail

Figure 7 shows a user interface that includes current GPS locations of the buses, continuously receiving updates from the server. It shows longitude and latitude of both buses.

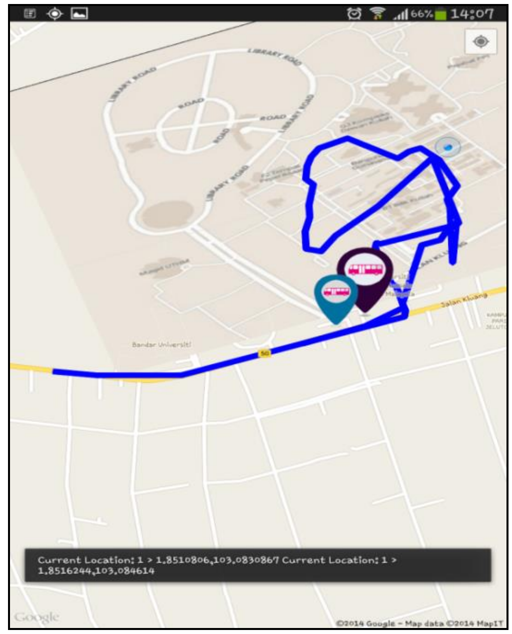

Figure 7: Display bus with actual GPS locations

Figure 8 shows the code snippet for updating the GPS location of the buses. After reading the current locations, the application will redraw the bus markers and update the textual information of the locations. It will loop the above process as long as the service is on.

```
MainActivity.BUSLOCATION += 1;
MainActivity.BUSLOCATION1 = MainActivity.BUSLOCATION + 1;

MainActivity.mBUS.setPosition(busroute.ROUTE.get(MainActivity.BUSLOCATION));

if (MainActivity.BUSLOCATION <1 ) MainActivity.BUSLOCATION = 1;

String msg = "Current Location: " + MainActivity.BUSLOCATION + " > " +
        Double.toString(busroute.ROUTE.get(MainActivity.BUSLOCATION).latitude) + "," +
        Double.toString(busroute.ROUTE.get(MainActivity.BUSLOCATION).longitude);
```

Figure 8: Code to display bus detail information

C. Testing

Several testing have been conducted to measure the arrival accuracy and the optimum update cycle settings for updating the location of the bus. Accuracy is measured as follows:

$$a = |(\text{estimated time of arrival} - \text{actual time of arrival})|$$

Table2
Value of a

Value of *a* (secs)	Accuracy
<= 5	Very Accurate
15	Accurate
45	Less Accurate
>= 120	Not Accurate

The values and accuracy threshold in the table are adapted from acceptable network delay [12]. Figure 9 demonstrates the accuracy of ETAs (estimated time of arrivals) in UBAIS. Due to location updates are subjected to internet traffics, the accuracy reading appear to be inconsistence, however, it converges around accuracy of +/-15 seconds.

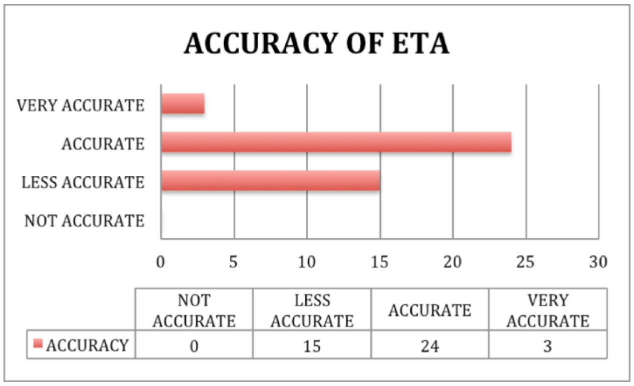

Figure 9: Accuracy of Estimated Time of Arrival (ETA)

Figure 10 shows the optimum frequency of bus location update cycles to be sent to the centralized server. It appears to give better accuracy with more regular updates. However, continuous updates may overwhelm a server and consume battery. Furthermore, with more buses' updates coming in, it may cause deadlocks to the database server. With the current configurations, it appears 4 seconds is the most optimum setting for update interval.

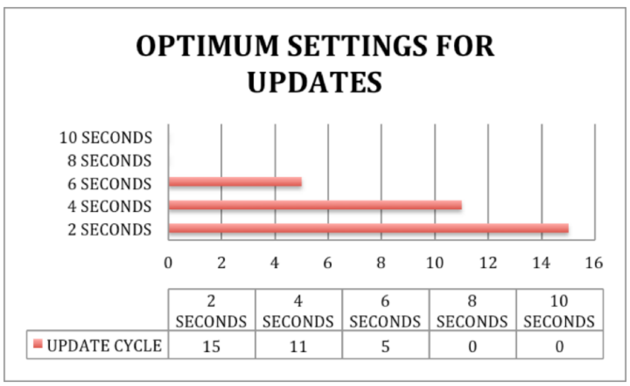

Figure 10: Optimum settings for update cycles

V. Conclusion

A bus tracking information system using consumer grade GPS smartphone connected via standard mobile network has been proposed. The system consists of three important components – firstly, smartphones installed on the buses that send locations via mobile broadband to a central server; secondly, the client android application that receive updates of their favorite bus locations in near real time; thirdly, the central location server that act as the middleman between the buses and the clients. From the experiments, the accuracy is relatively good, but there are rooms for improvement.

An improvement would be to re-design the framework to reduce update cycles yet achieve good accuracy. One way to do this is to apply push technology. The client application does not need to request location to get the latest location of the buses. Instead, it just sends an "alive" bit to the server. While the server regularly push updates to all the "alive" clients. This will ensure prompt delivery of information about the bus locations to the intended users. Dynamic update intervals may also improve the accuracy.

Acknowledgment

This work is supported by Universiti Tun Hussein Onn Malaysia. The grant number is RAGS R045.

References

[1] A. Minhans, S. Shahid, I. Ahmed, "An investigation into qualitative differences between bus users and operators for intercity travel in Malaysia,"*JurnalTeknologi*, vol. 70, no. 4, pp. 71-81, 2014.

[2] A. Minhans, S. Shahid, S. A. Hassan, "Assessment of bus service-quality using passengers' perceptions,"*Jurnal Teknologi*, vol. 73, no. 4, pp. 61-67, 2015.

[3] J. M. Pilachowski, *An approach to reducing bus bunching*, Dissertation, University of California, Berkeley, 2009.

[4] L. Tang, P. V.Thakuriah., "Ridership effects of real-time bus information system: A case study in the city of Chicago," *Transportation Research Part C: Emerging Technologies*, vol. 22, pp. 146-161, 2012.

[5] M. Brooks, C. Chidi, J. Mauldind, D. Nadeau.,"Waitless bus tracking device,"*Senior Design Project*, Georgia Institute of Technology, 2009.

[6] A. Koppineni., IIT Madras campus bus tracking**.** Retrieved November 10, 2015, from http://coeut.iitm.ac.in/webapp.html

[7] J. J. Drake, Z. Lanier, C. Mulliner, P. O. Fora, S. A. Ridley, G.Wicherski.,*Android Hacker's Handbook*. Wiley, Indianapolis, 2014.

[8] E. Hellman., *Android Programming: Pushing the Limits*. Pearson Technology Group, Indianapolis, 2013.

[9] G. Milette, A. Stroud, *Professional Android Sensor Programming*, Wiley, Indianapolis, 2012.

[10] M. Wolfson., *Android Developer Tools Essentials*.O'Reilly Media, New York, 2013.

[11] C. Delessio, L. Darcey, S. Conder., *Sams Teach Yourself: Android Application Development in 24 Hours*. Sams Publishing, 2015.

[12] J. Nielsen., Powers of 10: time scales in user experience. Retrieved February 20, 2016, from https://www.nngroup.com/articles/powers-of-10-time-scales-in-ux/