An Implementation of Vehicle Management System on the Cloud Computing Platform

Hua Yi Lin[1], Meng-Yen Hsieh[2], Yu-Bin Chiu[1] and Jiann-Gwo Doong[1]
[1]Department of Information Management, China University of Technology, Taiwan
[2]Department of Computer Science and Information Engineering, Providence University, Taiwan
E-mail: calvan.lin@msa.hinet.net

***Abstract*—This study attempts to exploit high-speed computing capability of cloud computing and smart phones to achieve vehicle management system running on golf carts. Our research use smart phones or tablet as car machines that provide an application of location-based services on the mobile device with GPS (Global Position System), wireless camcorder, Google-map, visualization information, graphical presentation to provide personalized service. This study allows users instantly to access the information and master the movement of cars. Additionally, the location of golf carts, surrounding environment, and personnel information are transmitted through a wireless network to the monitor center. Monitor center provides vehicles real-time services, the fleet management and caddy caring services. In case, the proposed mechanism performs well, this study adopts it on the other situation.**

Keyword: cloud computing, vehicle management, mobile position.

1. INTRODUCTION

With the evolution of mobile communications, this study would like to integrate the smart phone, tablet PC, GPS, Google Map, and Google GAE cloud platform into a vehicular management system. Each vehicle carries an android-based smart phone as a car machine that keeps sending its coordinates and service requests to the monitor center via wireless. Administrators browse the web site to realize each vehicle's location. Moreover, the vehicle can use blue-tooth to receive the status of the car battery, current, voltage and temperature, and then deliver the valuable data to the monitor center. Thus, this study will provide a more complete vehicular management service.

Our proposed system includes vehicle tracking, information services, history record and monitoring mechanism. Each vehicle carries a smart phone embedded GPS, 3G/4G and 802.11n that connect the satellite and Google map to process its latitude and longitude location or address translation. Each vehicle exploits the application to deliver its register time, car number, position and address to the Google cloud platform. The Google cloud platform is responsible for receiving the information from the moving vehicles, and storing the information into the data store. Thus, the monitor center performs the management and control functions.

The rest of this study is structured as follows. Section 2 indicates the system architecture, including the cloud computing and development software. Section 3 describes the proposed vehicle management system. Section 4 demonstrates our application and monitor center. Conclusions and future research are finally drawn in Section 5.

2. SYSTEM ARCHITECTURE

The session describes the architecture of our proposed system, including the software, hardware and platform. In order to achieve the implementation of vehicle management on cloud computing platform, this study reserves partial functions for the future extension. The following presents our system components.

(1)Hardware: The client and monitor center hardware use a smart phone and a notebook respectively. The detailed specification is shown in table 1.

Table 1. System Hard ware

Hardware	Client	Monitor Center
CPU	1.5 GHz，Quad-core	Intel®Core TM i5-2467M (1.60GHz)
OS	Android™ 4.0 with HTC Sense™	Windows O.S.
Graph Card/Monitor	4.7 Inch super LCD 2	Intel HD Graphics 4000/ 11.6"(W/IPS) Touch panel
Storage	32 GB	120GB SSD
Memory	1 GB	4G
Monitor	4.7 inch super LCD 2	11.6"(W/IPS)/ Touch panel
Protocol	HSPA/WCDMA/ GSM/GPRS/EDGE	802.11/802.3

(2)Software development platform: The software development platform includes applications on Android (client), and Google application for monitor center (server).The following introduces App Inventor, Google app engine and Eclipse (Integrated Development Environment).

App Inventor:

App Inventor is a project of the Google laboratory that uses a modular stacking method to design an Android program. Programmer only uses the basic component (such as button, text label) to assemble the entire program. Recently, the newer version is upgraded to App Inventor 2. The development interface of App Inventor is based on browser. It supports Mac OS X, Linux, and Windows OS. Each programmer can use Google account to login and save his project on the server. Thus each programmer performs his project in anywhere. App Inventor includes three parts. The related module is depicted in figure 1.

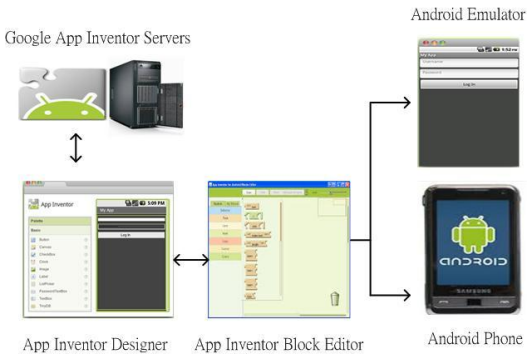

Fig.1. Operational modules from App Inventor

Google GAE cloud computing platform:

GAE supports a Platform as a Service (PaaS). Users focus on the development of webpage, and do not need to concern the server condition, network bandwidth, database, and resources. GAE is responsible for optimizing its working environment. The figure 2 depicts the detailed infrastructure, and there are front-ends, application servers, data-store, memory-cache and Google API. The following describes the function of each part.

Frontends:

Frontends is a load balancer, and it is responsible for receiving the requests from browsers. Subsequently, Frontends allocate the request to application server or static file server according the priority and the URL of clients. In case, there is no any request matching the URL of client, and then frontends response a error message HTTP 404 “Not Found”. The manager setups the service number of frontends named frontend instances. When the more the numbers of services that can offer the more the external request, and obtain faster responses.

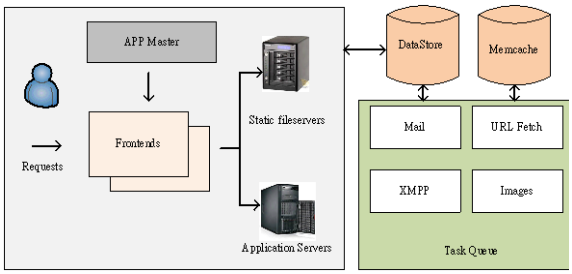

Fig. 2. GAE platform

Application Servers：

The application servers receive the requests from the frontend. Programmers exploit it to access the cloud data-store. Google adopts the big table to store data, or temporarily saves them into memory cache for improving the performance of accessing. Additionally, applications can use the application of GAE, such as URL fetch, task queue, mail, XMPP and images to reduce programming time.

Static file Servers:

As browser requests static resources, such as CSS, pictures, Java script. Frontend will obtain them from the static file server, and the application server do not need to process for improving the read and process performance. The static file is configured in appengine-web.xml.

Data-store：

GAE data-store provides JDO (Java Data Objects) and JPA (Java Persistence API) two types of entity store standards. Programmer exploits these API to access Google cloud database named data-store. Data-store is an object oriented database, and it is different from the relational database. The entity data is stored into a class, and each entity has a unique key for the identification.

Our proposed vehicle management system compromises android applications and the monitor center in GAE platform. Each golf cart in golf course carries a smart phone equipped with GPS, wireless, and 3G/4G for connecting the satellite. After receiving the location, the smart phone sends its coordinates to the GAE. The client applications compose four parts as follows:

Golf cart registers：The main function is to realize when the golf cart registers at the golf course. The monitor center records the related data into the center, such as battery status, maintenance time, driver and caddie.

Receiving massage: The main function is to pass and receive messages between a golf cart and the monitor center.

Navigation and Maps: The main function is to display the location and navigation for a golf cart.

Mood graffiti: The main function is to provide users with a simple camera, image processing and photo storages.

GAE cloud platform: The main function is to provide monitor center receiving a smart phone with its coordinates, address and valuable information, and stores them into the Google data-store. In additional, the monitor center instantly displays the real-time location of a golf cart, surrounding weather and traffic conditions on the monitor.

3. IMPLEMENTATION OF THE SYSTEM

Figure 3 presents our portal of the vehicle management system. The corresponding inventor blocks are shown in figure 4. When all blocks are done, users can down load and install the APK program into his smart phone. Subsequently, users exploit activity starter to open web page and start Google map.

Fig. 3. Vehicle management system portal

Fig. 4. The corresponding inventor blocks

Initially, when the golf cart registers, the portal shows up the login screen, as shown in figure 5. The main screen displays the golf cart's number, coordinate, 3G/4G signal, every 60 seconds recording its coordinate, and the button of displaying history coordinates. When the golf cart moves, the system real-time updates and synchronizes data on cloud platform to achieve monitoring capabilities. The following describes the procedure of our application.

Fig. 5. Golf cart registers

Fig. 6. The stored coordinates in cloud data-store

In this study, we obtain the current coordinate via location sensor, and then this study adopts tiny-webdb to record and share the car trace information. We build up the database on the cloud in advance, and store the golf cart's coordinate every 60 seconds. The users or monitor center can view the information on http://myappcom32.appspot.com to ensure the correctness of the writing data.

Figure 6 indicates the selected car number, and the application reads the history data from the Google data-store. In additional, the application exploits web viewer to call Google map functions to obtain the static map and the path of history, as shown in figures 7,8 and 9. In additional, the corresponding program block is as shown in figure 10.

Fig. 7. Selected car number

Fig. 8. The car trace coordinate

Fig. 9. Smart phone displays the history coordinates and map

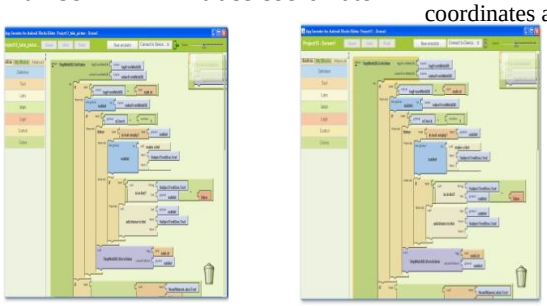

Figure 10. The corresponding program block

When the golf cart starts the vehicle management system, the smart phone keeps watching the information from monitor center. In case, any information is sent from monitor center, and the smart phone instantly displays on the screen. The driver or player is absent, after 10 seconds, the monitor center resends it to the caddy's smart phone, and ensures the message is arrival. This function exploits the following components, such as textbox_receiver, textbox_msgbody, button_send, texting, and notifier, as shown in figure 11.

Fig.11. Texting message in real time

In case, the driver needs help. The driver just presses the button (button_send), and then obtains the code and text from the texting component. This study also uses "texting.message.received" event to process the received message. Simultaneously, the screen popup the receiver: 5554 and the content "This is a book".

The implementation of the navigation and map on a test bed, we need the smart phone with GPS. When golf cart moves into the golf course, the smart phone obtains its coordinates and start recording its moving record. Initially, the application calls the location sensor to obtain he current position, and then use web viewer to start the Google map and mark the current coordinate, as shown in igure 12. Subsequently, the application saves its position nto Tiny_Web_DB in smart phone. After that, the pplication initials the activity starter and calculates the ath between beginning and destination place. Finally, the esults are displayed on the screen. The corresponding rogram block is as shown in figure 13.

Fig. 12. The navigation and map

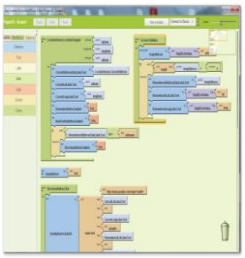

Fig. 13. The corresponding program block of the navigation and map

When users need to take a photo, the application provides graffiti mood function to take a photo, as shown in figure 13. Users select the red button and press the button to start the camera. The application also can play music, and process the image. Eventually, users save the photos into his smart phone at path: /storage/sdcard0/mydocuments/pictures/picture.png. In another word, users take a picture around the landscape as he needs. Figure 14 shows the function of graffiti mood.

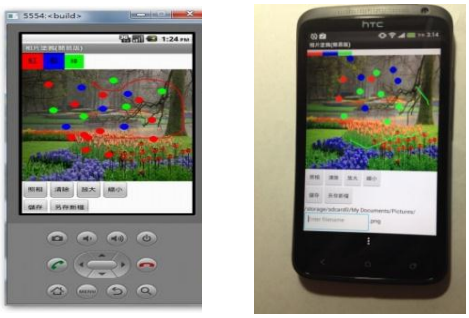

Fig. 14 The function of Graffiti mood

4. THE MONITOR CENTER OF VEHICLE MANAGEMENT SYSTEM

The monitor center is built via Eclipse IDE environment that provides Java and JSP programming on GAE cloud computing platform. In the proposed system, there are four parts agents, including Interface Service, Message Service, Display Service, and Record Service. The detailed infrastructure is as shown in figure 15. The main function of each agent is responsible for the following capability:

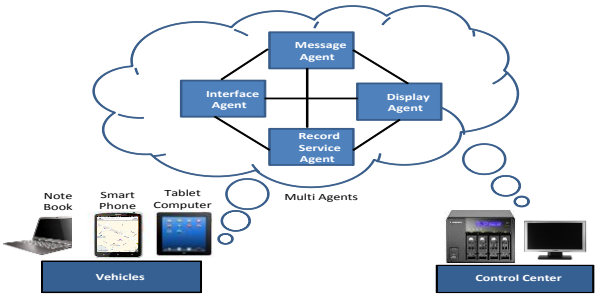

Fig. 15. Multiple agents in cloud computing platform

Interface agent：It takes charge of login and logout request. The moving golf cart uses smart phones to locate its coordinate, and sends its longitude and latitude back to monitor center via http request. After receiving the message, interface agent stores the received data into Google data-store. The data store adopts an object oriented database and maintains the table as an active list for golf carts, as shown in figure 16, and provides real time displaying service.

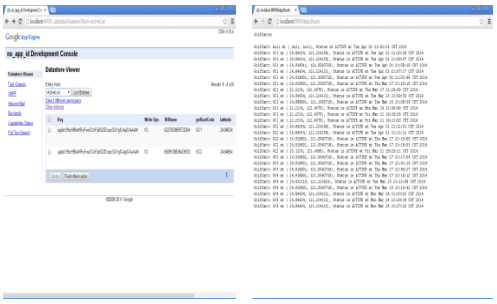

Fig. 16. Object oriented database in cloud computing platform

Record agent: When the interface agent and message agent receive the longitude and latitude, time or text message. It takes charge of accessing and modifying the GAE data-store.

Display service：This agent is responsible for displaying the active list from data store, and marking its place on the web page at monitor center. The markup message includes the weather, address, traffic situation, as shown in figure 17.

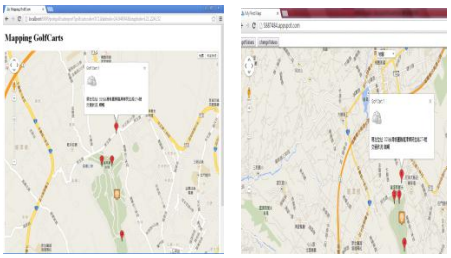

Fig. 17. The mark of a moving golf cart at the monitor center

Message service agent: It exploits http request/response methods to send and receive text message between golf carts and the monitor center. It stores the received message and store into the data-store for later analysis, or display message on monitor web page.

Our cloud-based vehicle management system owns the following operational procedures.
When the golf cart starts the vehicle management system, the system sends its current coordinate every 60 seconds to the interface agent and the interface agent send it to the record agent. Finally the record agent saves it into data-store. The monitor center connects the web site and watches each golf cart in the golf field to realize the route path of each golf cart. In additional, the monitor center can also real time displays the position, map, weather and traffic situation of each golf cart. The detailed information can be seen via the URL on 5887484.appspot.com.The red spot indicates the allocation of each golf cart, when user clicks on the spot, it shows up the current location of the golf cart,

surrounding weather and traffic situations. Additionally, users can select the Google's satellite map to see the real street map, as shown in figure 18.

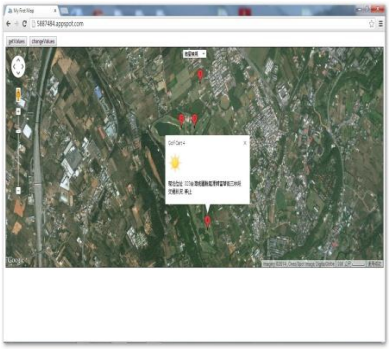

Fig. 18. Google's satellite map

When a user requests services, he uses the smart phone to send a request to the monitor center. As the interface agent receives the request, it delivers the request to the message agent. Subsequently, the message agent sends the user's coordinate and request to the record agent to store the information into data-store.

The implementation of the vehicle management system on GAE platform：

Here, we use Eclipse IDE to design the monitor center. Simply, the application executes receiving data from golf carts, storing data into the GAE data-store, and displaying the location of car. This study divides the program into three parts, including model, view and controller (MVC), as shown in figure 19.

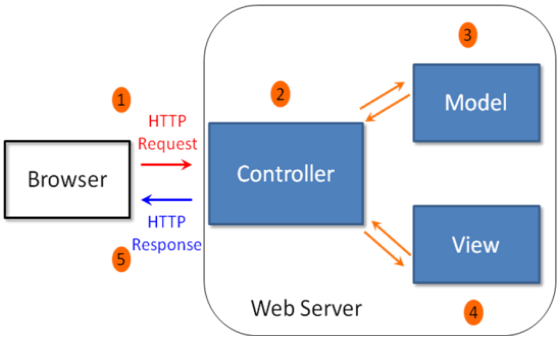

Fig. 19. MVC: model, view and controller

(1)Web page view:

JSP (Java Server Pages) is responsible for showing a page on the screen. Also, we can embedded the dynamic page into html, and dynamically shows data. GodfCartMap.jsp is the code for showing the golf cart location via map.

In golf-cart-map, this study uses Google maps API v3 command to design the display map, golf carts and marked position. The path of a moving golf cart is stored into active-list in data-store. After executing GolfCartMap.jsp, the screen displays the following map, as shown in figure 20.

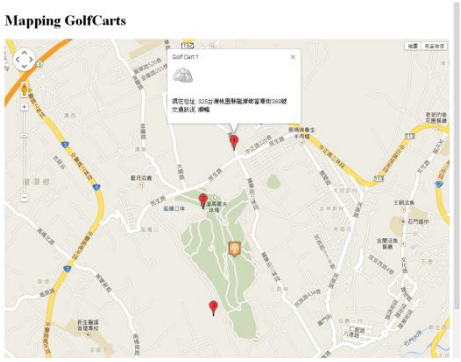

Fig 20. GolfCartMap.jsp displays the web page

(2)Controller:

Controller program is used to deal with the request from the browser, receive and process the input message, and integrate the other components to response the request. This study uses Java servlet to implement the control program, and there are three parts, including interface service, message service, recording service. The detailed presentation is as follows:

Interface service:

It deals with the login and logout service from smart phone, including Login-servlet, Post-GolfCart-Report-Servlet, and LogoutServlet. As receiving a login request, login-servlet checks the number of the golf cart, and responses welcome message. When the post-golf-cart-report-servlet receives a post request, it resolves the input of the number of golf cart, latitude, longitude, status and report-date-time-data, and then packets them into the golf-cart-report and active-list data object, eventually storing them into data-store. When receiving the logout request, log-out-servlet responses welcome, and delete the golf cart number from the active-list.

(3)Message service:

The message service is responsible for the request from smart phone. Google GAE provides with Mail, API, XMPP Services. This study uses message servlet to deal with the request and responses the acknowledgement.

Recording service:

This service stores golf-cart-report and active-list entities into data-store. Also, it responses the request from browser's list, and displays all golf carts' history. The record service includes PMF, DB-utils, and list-golf-cart-servlet. PMF is an interface of the JDO. DB-util is an application for accessing data-store, and the list-golf-cart-servlet responses the request of a list. After querying data-store, it displays all information. Additionally, the web.xml is configuration file in Java EE and servlet, and this study uses it to define the relation of servlet and URL in out program. Eventually, Eclipse creates it when building a new project automatically at war/web-inf/web.xml. The programmer needs to modify it for a running environment.

5.CONCLUSION AND FUTURE WORK

In this study, our proposed applications focus on dealing with the user requests and vehicle management in real time. Through the history of activity and record, we can analysis the behavior of vehicles. Thus, the proposed system provides effective and hospitality customized services. Moreover, the golf resources can be optimized at allocation and operation. To enhance the safety and reliability, this study will combine the golf cart with wireless sensor to detect the surrounding environment, tire pressure and golf cart maintenance. Thus the manager can save human power, financial and material resources. After revising the proposed system, we will adopt it on intelligent transportation, logistics, and storage management library. In the future, this study will enhance the functions to achieve the environmental protection, nature conservation, public medical and health caring.

REFERENCES

[1] S. Zhang, S. F. Zhang, X. B. Chen, & X. Z. Huo, "Cloud Computing Research and Development Trend", in Proceedings of the IEEE 2010 Second International Conference on Future Networks, pp. 93-97, 2010.

[2] N. Sultan, "Cloud computing for education: A new dawn", International Journal of Information Management, vol. 30, no. 2, pp. 109-116, 2010.

[3]http://mic.iii.org.tw/intelligence/reports/pop_Doc_promote.asp?docid=CDOC20091110009, 2011.

[4]http://www.digitimes.com.tw/tw/rpt/rpt_show.asp?cnlid=3&cat=MCN&v=IM0807&n=0, 2011.

[5]S. J. Barbeau, M. A. Labrador, P. L. Winters, R. Perez, and N. L. Georggi, "General Architecture in Support of Interactive Multimedia, Location-based Mobile Application", IEEE Communications Magazine, vol. 44, no.11, pp.156-163,2006.