International Journal of Business, Human and Social Sciences
ISSN: 2517-9411
Vol:9, No:10, 2015

SAP: A Smart Amusement Park System for Tourist Services

Pei-Chun Lee, Sheng-Shih Wang, Pei-Hsuan Ku

***Abstract*—**Many existing amusement parks have been operated with assistance of a variety of information and communications technologies to design friendly and efficient service systems for tourists. However, these systems leave various levels of decisions to tourists to make by themselves. This incurs pressure on tourists and thereby bringing negative experience in their tour. This paper proposes a smart amusement park system to offer each tourist the GPS-based customized plan without tourists making decisions by themselves. The proposed system consists of the mobile app subsystem, the central subsystem, and the detecting/counting subsystem. The mobile app subsystem interacts with the central subsystem. The central subsystem performs the necessary computing and database management of the proposed system. The detecting/counting subsystem aims to detect and compute the number of visitors to an attraction. Experimental results show that the proposed system can not only work well, but also provide an innovative business operating model for owners of amusement parks.

***Keywords*—**Amusement park, location-based service (LBS), mobile app, tourist service.

I. Introduction

According to the World Tourism Organization in a report of year 2000 edition, tourism has become the primary source of foreign exchange in many countries [1]. As a part of tourism industry, amusement parks have been playing an important role. Therefore, the development of amusement parks is one of the significant factors in the improvement of the hospitality and tourism industry worldwide. Several major issues regarding amusement parks include the visitors' satisfaction, loyalty and revisit intention. Information and communications technologies (ICT) can be good facilitation to these issues while the technologies have already become a major trend in the industry of amusement parks as observed.

Famous theme parks such as Walt Disney World [2], Universal Orlando Resort [3], Ocean Park Hong Kong [4], Janfusun Fancy World [5] and Leofoo Village Theme Park [6] in Taiwan have introduced ICT-based techniques into their park services, especially mobile apps [7]-[16] which get more and more popular in recent years. In addition to basic park and attraction information inquiry, these ICT-based park services have been designed to solve several existent problems which put tourists under pressure, such as tiresome long wait times for attractions. For example, Walt Disney World [17] and Universal Orlando Resort [18] have been carrying out FastPass+ or Express^{SM} Pass schemes for highly popular attractions and providing tourists with wait time of each attraction. Janfusun Fancy World [5], [13] presents a solution to show three colors representing different wait time periods of attractions.

The solutions mentioned in the examples are all useful yet leave various levels of decisions to tourists to make by themselves, which may bring on pressure on tourists and negatively impact tourists' experiences in the tour. What if the park service goes the extra mile to give tourists an even easier experience by automatically planning customized schedules for individual tourists? In response to this question, this paper proposes a smart amusement park (SAP) system to accommodate the tour pressure problem with an innovative solution called the GPS-Based dynamic scheduling scheme to offer tourists the customized best plans of tour (a recommended attraction with the recommended time to arrive and play) based on tourists' GPS coordinates without tourists making plans by themselves. Tourists can use the mobile app to establish their personal attraction list, choose their preferred attraction priority (closest first, shortest wait time first, or hottest first), and require the SAP system to figure out the best tour plans for them.

The rest of this paper is organized as follows. Section II gives an overview of the proposed system. Section III elaborates the proposed smart amusement park (SAP) system. Section IV presents the implementation and testing of the SAP system. Finally, Section V concludes this paper.

II. Overview

The SAP system consists of three subsystems: mobile app subsystem, central subsystem, and detecting/counting subsystem.

A. Mobile App Subsystem

This subsystem is the mobile app provided by the amusement park for tourists to download and install on their smartphones or tablets, and can be used by tourists to access SAP services. This subsystem may communicate with the central subsystem via Wi-Fi or 3G/4G communications systems through the Internet.

B. Central Subsystem

This subsystem is responsible for SAP system kernel computing and database management, and may be located in the control room of the amusement park. It communicates with the mobile app subsystem (app) and the detecting/counting

P. C. Lee and P. H. Ku are with the Department of Information Management, Minghsin University of Science and Technology, Xinfeng, Hsinchu 30401 Taiwan (e-mail: pjlee@must.edu.tw, p2686290@gmail.com).

S. S. Wang is with the Department of Information Management, Minghsin University of Science and Technology, Xinfeng, Hsinchu 30401 Taiwan (Corresponding author, phone: +886-3-5593142ext.3444; fax: +886-3-5595142; e-mail: sswang@must.edu.tw).

subsystem to process the requests or notifications from these two subsystems. The main function of this subsystem is dynamic scheduling by which the system determines the best plan of tour based on the GPS location of the tourists and the tourist's preferred priority.

C. Detecting/Counting Subsystem

This subsystem consists of sensors which can sense human penetration cooperated with computing elements supposed to be deployed at the entrances of the attractions in the amusement park. When a tourist enters an entrance, the sensor will detect this event and notify the computing element to accumulate the number of daily visits and the queue length. Also, this subsystem will re-calculate the queue length at every turn of attraction operation. The number of daily visits and the queue length will be sent to the central subsystem via Ethernet or Wi-Fi at appropriate timings to update the values of monthly visits and the queue length of the attractions in the database.

III. SAP: Smart Amusement Park System

This section describes the detailed design of the proposed smart amusement park system.

A. Mobile App Subsystem

The mobile app subsystem provides tourists with an interface to use SAP services. Tourists need only to download and install the app into their smartphones or tablets and everything is on the go. The architecture of the mobile app subsystem is shown in Fig. and will be depicted as follows.

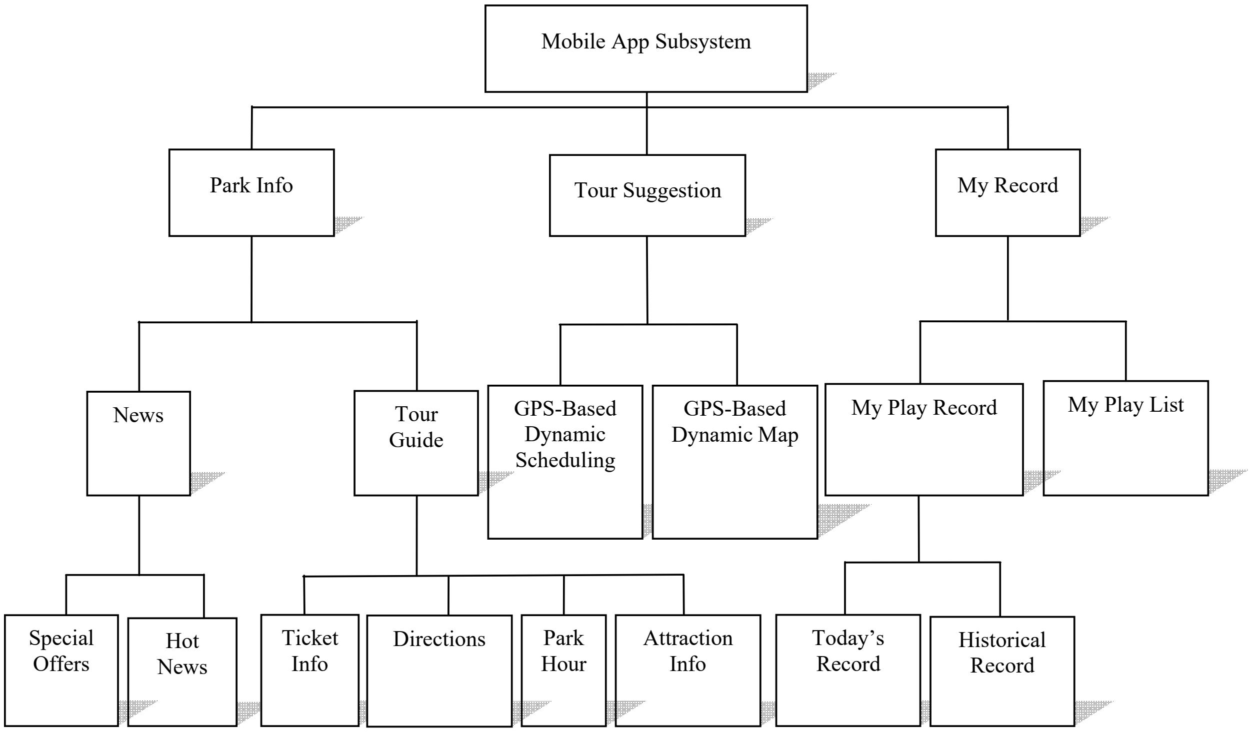


Fig. 1 Architecture of Mobile App Subsystem

1. Park Info

Park Info provides the information about the amusement park, such as News and Tour Guide. News offers tourists Special Offers and Hot News, and Tour Guide introduces Ticket Info, Directions (traffic), Park Hour and Attraction Info to tourists. Attraction Info provides tourists with detailed information about each attraction in the park, where attractions are categorized by which theme area they resides. Besides, the search function is provided for tourists acquainted with attraction names to do a quick keyword search. Tourists can select attractions in Attraction Info to add them into their personal My Play List. My Play List can later be used in Tour Suggestion function.

2. Tour Suggestion

As a location-based service, app can determine whether a tourist is in the park or not based on the GPS coordinates. When a tourist activates the Tour Suggestion function, app will first check if the tourist is in the park. If so, app will then provide the tourist with GPS-Based Dynamic Scheduling function and GPS-Based Dynamic Map function.

i. GPS-Based Dynamic Scheduling

This feature offers three choices of different preferable priorities such as Closest First, Shortest Wait Time First, and Hottest First. The tourist can choose one of the three priorities as his or her preferred scheduling priority to require SAP system to find the best (recommended) attraction in My Play List based on the priority, predict the best (recommended) time to arrive and play in the attraction, and provide all the recommended attraction information to the tourist as the best scheduling solution. Procedures of the three priorities are elaborated respectively as follows.

- Closest First

The SAP system regards the attraction with the shortest walking time as the closest attraction (the walking speed is considered as being constant). When a tourist activates the GPS-Based Dynamic Scheduling function with priority of Closest First, app will use Google Maps Directions API [19] to acquire the walking times of the tourist from his or her GPS coordinates to all the attractions' GPS coordinates in My Play List. Then app will find the attraction with the shortest walking time, request the central subsystem to calculate its recommended time to arrive and play, and display the result to the tourist.

- Shortest Wait Time First

After app obtains the walking times to the attractions in My Play List using Google Maps Directions API, it will send them to the central subsystem. Then the central subsystem calculates and finds the attraction with the shortest wait time and its recommended time to arrive and play based on the walking times received. Later on, the central subsystem sends back the result to app for displaying. Note that the wait time in this paper is not the general wait time of an attraction, but an adjusted wait time based on the tourist's GPS coordinates. In other words, the wait times of tourists may differ depending on how long they need to walk to an attraction.

- Hottest First

The SAP system regards the attraction with the highest Monthly Visits as the hottest attraction. Similarly, app will acquire the walking times using Google Maps Directions API and send them to the central subsystem. Then the central subsystem finds the hottest attraction in My Play List based on the Monthly Visits of each attraction. With the walking time of the hottest attraction, the central subsystem calculates the recommended time to arrive and play, and returns the result to app as the best solution for the tourist.

i. GPS-Based Dynamic Map

When a tourist requires the GPS-Based Dynamic Map function for a specific attraction, app will use Google Maps Directions API and Google-Play-Services library to show the route and direction from the tourist to the specified attraction on the electronic map of the amusement park, where the map can be easily zoomed in and out.

ii. My Record

There are two lists in My Record described as follows.

- My Play Record

In My Play Record, the tourist can check which attractions he or she has selected during the day in Today's Record. Moreover, the tourist can search for his or her selected attractions within months, years, or a specific duration in Historical Record.

- My Play List

This List can be used by tourists to require the GPS-Based Dynamic Scheduling function performed by SAP system. The attractions can be selected in Tour Guide/Attraction Info or modified directly in My Play List.

B. Central Subsystem

The central subsystem is responsible for system kernel computing and database management of SAP system. In addition to park information provision, this subsystem accepts requests from app, processes the requests accordingly, and replies app with corresponding results. Furthermore, this subsystem receives the statistics of daily visit counts and current queue lengths of attractions from the detecting/counting subsystem, and updates the fields of Monthly Visits and Queue Length in the Attraction table in the database.

GPS-Based Dynamic Scheduling is the main scheme proposed in this paper and its kernel algorithm is implemented in the *central subsystem*. GPS-Based Dynamic Scheduling scheme accepts the tourist's request from app and finds the best plan of tour based on tourist's GPS coordinates, My Play List, and preferred attraction priority (Closest First, Shortest Wait Time First, or Hottest First), aiming to offer a better customized tour experience and easier way for touring. The kernel algorithm of this scheme calculates the estimated wait time and the recommended arrival time of an attraction for a tourist who is somewhere in the park apart from the attraction. Note that the wait time is not the general wait time which is the same for all tourists, but it is an adjusted wait time based on tourists' GPS coordinates. To be specific, the calculation considers both the processing time of the current queue and the walking time of the tourist from his or her starting position to the attraction. The processing time of the current queue is related and affected by the remaining wait time of the current queue and the current queue length, as well as the capacity and duration of the attraction [20]. Therefore, for a tourist, we can calculate his or her estimated wait time t_i^{Wait} of a specific attraction i (where i =1, 2, …, n represents the ID of an attraction in a park with n attractions) as in Algorithm 1, where t_i^{Walk} is the walking time of the tourist from his or her starting position to the position of the attraction, t_i^{Remain} is the remaining wait time of the current queue, n_i^N is the current queue length, n_i^{Op} is the attraction capacity, and t_i^{Op} is the attraction duration. Note that the recommended arrival time *BestRnd* can also be calculated according to the start time *CurrRnd* of the current attraction operation in Algorithm 1.

Algorithm 1: Estimation of wait time and recommended time to arrive and play.

Input: $n_i^N, n_i^{Op}, t_i^{Walk}, t_i^{Remain}, t_i^{Op}$, *CurrRnd*

Output: t_i^{Wait}, *BestRnd*

begin

if $n_i^N < n_i^{Op}$ **then**

if $t_i^{Walk} \le t_i^{Remain}$ **then**

$$t_i^{Wait} = t_i^{Remain} - t_i^{Walk};$$

$$BestRnd = CurrRnd + t_i^{Op};$$

else

$$t_i^{Wait} = t_i^{Op};$$

$$BestRnd = CurrRnd + 2 \times t_i^{Op};$$

else

if $t_i^{Walk} \le t_i^{Remain} + \left\lfloor \frac{n_i^N}{n_i^{Op}} \right\rfloor \times t_i^{Op}$ **then**

$$t_i^{Wait} = t_i^{Remain} + \left\lfloor \frac{n_i^N}{n_i^{Op}} \right\rfloor \times t_i^{Op} - t_i^{Walk};$$

$$BestRnd = CurrRnd + \left(1 + \left\lfloor \frac{n_i^N}{n_i^{Op}} \right\rfloor\right) \times t_i^{Op};$$

else

if $t_i^{Walk} - \left(t_i^{Remain} + \left\lfloor \frac{n_i^N}{n_i^{Op}} \right\rfloor \times t_i^{Op}\right) \le t_i^{Op}$ **then**

$$t_i^{Wait} = t_i^{Walk} - \left(t_i^{Remain} + \left\lfloor \frac{n_i^N}{n_i^{Op}} \right\rfloor \times t_i^{Op}\right);$$

$$BestRnd = CurrRnd + \left(1 + \left\lfloor \frac{n_i^N}{n_i^{Op}} \right\rfloor\right) \times t_i^{Op};$$

else

$$t_i^{Wait} = t_i^{Op};$$

$$BestRnd = CurrRnd + \left(1 + \left\lceil \frac{t_i^{Walk}}{t_i^{Op}} \right\rceil\right) \times t_i^{Op};$$

end

Based on Algorithm 1, the GPS-Based Dynamic Scheduling scheme is elaborated as follows. When the central subsystem receives a GPS-Based Dynamic Scheduling request from app, it determines which priority the request requires and proceeds accordingly:

1. Closest First

When this subsystem receives the request from app, it will calculate the estimated wait time and recommended arrival time based on the shortest walking time and send the result back to app.

2. Shortest Wait Time First

Similar to 1, after this subsystem obtains the tourist's walking times to each of the attractions in My Play List, it calculates the estimated wait times of each attraction based on Algorithm 1, chooses the attraction with the shortest wait time as the recommended attraction, and sends back the result including recommended arrival time to app as the best solution.

3. Hottest Fist

The SAP system regards the attraction with the highest Monthly Visits as the hottest attraction. After receiving tourist's request from app, this subsystem will retrieve the hottest attraction, calculate the estimated wait time and recommended arrival time using the walking time of the hottest attraction based on Algorithm 1, and return the result to app as the best solution.

As for the design of the database of the SAP system, this database consists of database tables such as News table, Attraction table, etc. The Attraction table stores all required attraction information of each attraction in the amusement park. The *Attraction ID* field is the primary key of the Attraction table that uniquely defines a particular attraction and relates to the other tables. The information in the fields of *Name*, *Photo*, *Limitation*, *Thrill Level*, and *Maintenance* are provided when a tourist looks up the Attraction Info. The *Longitude* and *Latitude* fields are required when SAP system performs the GPS-Based Dynamic Scheduling and GPS-Based Dynamic Map schemes. The values in the fields of *Queue Length*, *Duration*, *Capacity*, and the *Session Time* are used for calculation in Algorithm 1. The *Monthly Visits* and *Queue Length* fields are updated according to statistics sent by the detecting/counting subsystem. The value of the *Monthly Visits* field is reset and re-counted in the beginning of each month, and represents the popularity of the attraction.

C. Detecting/Counting Subsystem

The detecting/counting subsystem consists of sensors which can sense human penetration cooperated with computing elements deployed at the entrance of each of the attractions in the amusement park. The tourist penetration through an entrance will be detected by the sensor and then the sensor will notify the computing element to accumulate the daily visit counts and the queue length. Also, the computing element will re-compute the queue length at every operation turn. The daily visit counts and the queue length will be sent to the central subsystem at appropriate timings to update the Monthly Visits and Queue Length of the attraction in the database.

IV. System Implementation and Testing

A. Environment and Implementation Tools

The implementation environment of SAP system is shown in Fig. 1. The mobile app subsystem is developed in Android 4.3 platform using Eclipse integrated development environment (IDE) with Android SDK. The central subsystem is implemented using Visual Studio C#, hosted on an HP desktop PC running Windows. Microsoft SQL Server serves as the system database on the same desktop PC. The detecting/counting subsystem consists of a programmable Arduino UNO microcontroller board, an infrared sensor and a notebook, where the Arduino UNO board is connected with the notebook via a USB cable. As for the communication between the subsystems, the mobile app subsystem may communicate with the central subsystem via Wi-Fi or 3G/4G communications systems through the Internet, and the detecting/counting subsystem may communicate with the central subsystem via Ethernet or Wi-Fi systems.

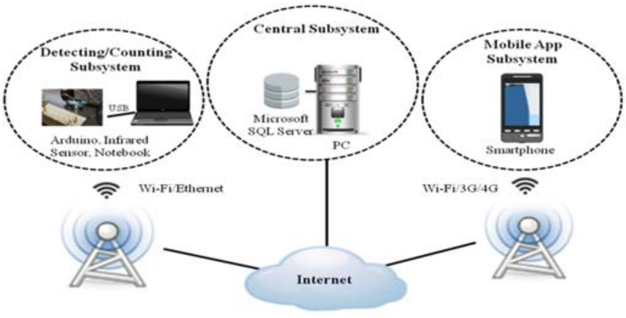


Fig. 1 SAP system implementation environment

B. Testing and Experimental Results

The testing field of the SAP system is shown in Fig. 2. There are six attractions located in the testing field. We will show some of our experimental results as follows. First, we test if the mobile app subsystem can access the system database in the central subsystem correctly. We start the experiment from the main menu of the mobile app subsystem. The successive screen captures are shown from Fig. 3 (a)-(d). Compared with the content in the database, we verify that the mobile app subsystem can access the database in the central subsystem and show the result correctly.

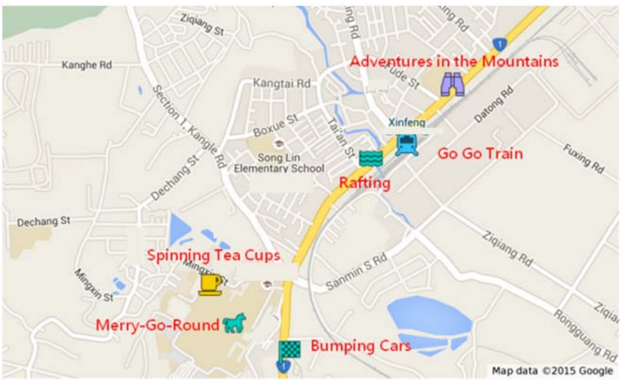
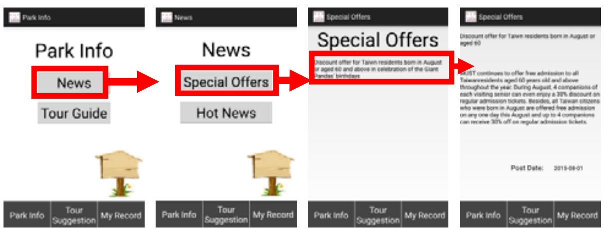


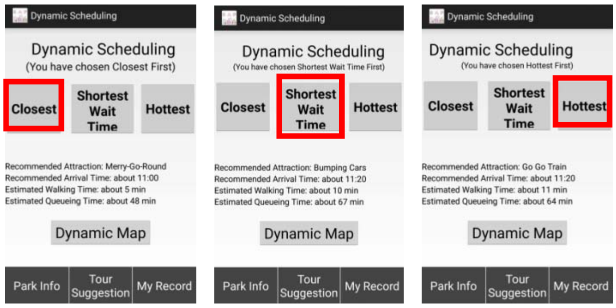
Fig. 2 The testing field of the SAP system



(a) (b) (c) (d)

Fig. 3 The experiment on Park Info

Second, the main feature of the SAP system, GPS-Based Dynamic Scheduling function, is verified as follows: Compared the results of the three priorities Closest First, Shortest Wait Time First, and Hottest First as shown respectively in Fig. 4 (a), (b), and (c) with the captured output on the central subsystem screen. We confirm that the SAP system calculates and displays the results correctly, and the communication between the mobile app subsystem and the central subsystem is correct as well. Consequently, we verify that the GPS-Based Dynamic Scheduling function does work well. In addition, we confirmed that the GPS-Based Dynamic Map derived from the proposed system is also correct.



(a) (b) (c)

Fig. 4 The experiment on GPS-Based Dynamic Scheduling

V. Conclusion

In this paper, we propose a smart amusement park (SAP) system, which offers ICT-based services to tourists for a more relaxing and easier tour experience. A specific novel feature called GPS-Based Dynamic Scheduling is designed to provide tourists with customized tour suggestions (recommended attraction with recommended arrival time) based on each tourist's favorite attractions, preferred attraction priority, and GPS coordinates without tourists making various tiresome decisions by themselves. Serving as a location-based service, the GPS-Based Dynamic Scheduling scheme considers not only the general wait time of current queue but also the estimated walking time of the tourist based on GPS coordinates to yield a location-based wait time. Furthermore, the mobile app of SAP system gives an integrated, easy-to-use interface for tourists who are familiar with smartphones or tablets nowadays. The design and implementation of the SAP system are described in this paper, and the experimental results show that the proposed SAP system functions well.

Acknowledgment

This work is supported by the Ministry of Science and Technology of the Republic of China under Grant MOST 103-2221-E-159-006.

References

[1] Website of Tourism Bureau, M.O.T.C. Republic of China (Taiwan). Section 1.1 (Online). Available: http://admin.taiwan.net.tw/upload/contentfile/auser/b/wpage/chp1/1_1.1.htm
[2] Website of Walt Disney World Resort (Online). Available: https://disneyworld.disney.go.com/
[3] Website of Universal Orlando Resort (Online). Available: https://www.universalorlando.com/Home.aspx
[4] Website of Ocean Park Hong Kong (Online). Available: http://www.oceanpark.com.hk/html/en/home/
[5] Website of Janfusun Fancy World (Online). Available: http://fancyworld.janfusun.com.tw/en/

[6] Website of Leofoo Village Theme Park (Online). Available: http://www1.leofoo.com.tw/village/en/Info.aspx
[7] Mobile app with title "My Disney Experience" on Google Play, (Online). Available: https://play.google.com/store/apps/details?id=com.disney.mdx.wdw.google
[8] Mobile app with title "My Disney Experience - Walt Disney World" on App Store, (Online). Available: https://itunes.apple.com/app/id547436543
[9] Mobile app with title "Universal Orlando® Resort App" on Google Play, (Online). Available: https://play.google.com/store/apps/details?id=com.universalstudios.orlandoresort
[10] Mobile app with title "The Official Universal Orlando® Resort App" on App Store, (Online). Available: https://itunes.apple.com/tw/app/official-universal-orlando/id878217080
[11] Mobile app with title "Ocean Park Hong Kong" on Google Play, (Online). Available: https://play.google.com/store/apps/details?id=com.oceanpark.mobileapp
[12] Mobile app with title "Ocean Park Hong Kong" on App Store, (Online). Available: https://itunes.apple.com/hk/app/ocean-park-hong-kong/id554122091
[13] Mobile app with title "Janfusun Fancyworld" on Google Play, (Online). Available: https://play.google.com/store/apps/details?id=tw.com.rainmaker.jfs
[14] Mobile app with title "Janfusun Fancyworld" on App Store, (Online). Available: https://itunes.apple.com/tw/app/jian-hu-shan-shi-jie/id680988811?mt=8
[15] Mobile app with title "Leofoo Village with Fun" on Google Play, (Online). Available: https://play.google.com/store/apps/details?id=com.uni.leofoo
[16] Mobile app with title "Leofoo Village with Fun" on App Store, (Online). Available: https://itunes.apple.com/tw/app/liu-fu-cun-hao-hao-wan/id862930941?mt=8
[17] Webpage with title "With FastPass+, The Forecast Is More Magic," (Online). Available: https://disneyworld.disney.go.com/plan/my-disney-experience/fastpass-plus/
[18] Webpage with title "Universal Express^{SM} Passes," (Online). Available: https://www.universalorlando.com/Theme-Park-Tickets/Universal-Express/Express-Passes.aspx
[19] Webpage with title "The Google Maps Directions API," (Online). Available: https://developers.google.com/maps/documentation/directions/
[20] V. Vukadinovic, F. Dreier, and S. Mangold, "A simple framework to simulate the mobility and activity of theme park visitors," in *Proc. 2011 Winter Simulation Conf.*, pp.3253-3265.