

Article

A Theme Park Tourist Service System with a Personalized Recommendation Strategy

Feng-Chi Yu ^{1,*}, Pei-Chun Lee ², Pei-Hsuan Ku ³ and Sheng-Shih Wang ²

¹ Institute of Service Industrial and Management, Minghsin University of Science and Technology, Hsinchu 30401, Taiwan

² Department of Information Management, Minghsin University of Science and Technology, Hsinchu 30401, Taiwan; pilee@must.edu.tw (P.-C.L.); sswang@must.edu.tw (S.-S.W.)

³ Accton Technology Corporation, Hsinchu 30077, Taiwan; bagel_ku@accton.com

* Correspondence: geeyiu@must.edu.tw; Tel.: +886-03-559-3142 (ext. 2613)

Received: 31 August 2018; Accepted: 24 September 2018; Published: 27 September 2018



Abstract: In general, there exists numerous attractions installed in a theme park, and tourists in a theme park dynamically change their locations during a tour. Thus, a tourist may cope with the issues of selecting the attractions to visit while planning the tour route. This paper, based on the concept of location awareness, proposes a novel waiting time, called the personalized waiting time, to introduce a location-aware recommendation strategy. In addition, this paper presents an architecture of tourist service system using the proposed recommendation strategy to relieve the pressure on tourists and create the pleasant experience in their tours. The proposed location-based system consists of mobile app, ticket-reader, detecting/counting, and central subsystems, and the whole system was implemented in this study. We conducted numerous experiments and field testing results validated that the entire proposed system can correctly provide information, such as attraction introduction, recommended session time, estimated moving and waiting time, tour map, and the number of reservations. The system functions, including dynamical scheduling, attraction reservation, ticket verification, visitor detection, and visitor counting, also worked well.

Keywords: tourism; theme park; location awareness; recommendation system; personalization

1. Introduction

Recently, with the great popularity and pervasiveness of smart devices (e.g., smart phones or tablet PCs), mobile apps have gradually become a commonplace service in people's daily lives. In addition, with the rapid emerging of information and communication technologies (e.g., mobile and wireless communication, embedded technology, and sensing technology), more and more conventional industries introduce these technologies into their business operation. The tourism and leisure industry is no exception. Famous theme parks such as Walt Disney World, Universal Orlando Resort, and Ocean Park Hong Kong have all introduced information and communication technologies into their park services, which can facilitate visitors' satisfaction, loyalty, revisit intention, etc. Among these theme parks, Walt Disney World is undoubtedly a good example in creating fantastic user experiences based on various technologies such as RFID wristbands (Magic Bands), mobile apps, mobile payment, and even big data analysis [1], which altogether provide considerate services.

Although Walt Disney World and other theme parks such as Universal Orlando Resort have improved visitors' experiences with services such as FastPass+ or ExpressSM Pass, we observe that there is still room for improvement. One of the problems is that, in their mobile apps, when tourists need some attraction suggestions, there are a variety of decisions for tourists to make by themselves, which may cause pressure, or so-called choice overload, on tourists and negatively impact tourists'

experiences [2,3]. Moreover, the suggested attractions might include an attraction that the tourist does not want to visit because they are filtered from all attractions in the theme area, not according to the tourist's wish or favorite attraction list only. Another problem is that the suggestion provided by their mobile apps is a list of attractions, often quite lengthy, and the tourists have to browse through the list only to find one attraction they want to visit next.

In addition, the literature has investigated the issue of waiting in services and argued that the waiting time is actually important for services in many places such as restaurants and public transportation stations because it may influence the service experience of customers [4–7]. Although using many fantastic technologies, the problem of long lines in front of the most popular attractions is one of the core problems to be resolved in many theme parks. Walt Disney World, Orlando has tackled this problem with the FastPass+ mechanism, a sorted list of attractions' prompt wait times, and an interactive gaming mobile app for visitors to play during their waiting time in the line. The FastPass+ mechanism is basically a way to reserve an attraction in advance before a visitor goes to take the ride. The sorted indication of attractions' current wait times assists the visitors with their visiting decisions. The interactive gaming mobile app turns visitors' wait times to play times and might help relieve the bad feelings due to long waiting. Among the three solutions, we find that there is still one more thing which can be further improved, namely, the attractions' wait times. The attraction's wait time, called the general waiting time, provided by Walt Disney World is general to all visitors, no matter where the visitors are [8]. That is, the time does not take the tourist's location into account and does not supply visitors with their mobility time information to the attractions based on the visitors' location information. Unfortunately, the Walt Disney World's system only provides general attraction wait times regardless of where the visitors are. Universal Studio's system exhibits the same problem.

Many tourism recommendation systems are based on combinatorial optimization challenges. According to the level of complexity and requirements of users' recommended items, different recommendation strategies are applied in tourism recommender systems [9]. To query relatively simple items, such as restaurants or shops, content-based and collaborative filtering-based recommendation strategies are usually applied. Content-based strategies recommend services of tourism based on the similarity of user preferences and the descriptive information of services [10]. Collaborative filtering recommendation strategies are based on the opinions of users who share similar interests or locations [11–13]. In addition, knowledge-based and hybrid recommendation strategies are utilized to query more complex items, such as travel routes and plans [14,15]. Knowledge-based recommendation strategies offer items to users based on the knowledge about the relationship between users' requirement and a possible recommendation. The hybrid recommendation approach combining two or more recommendation strategies into one hybrid strategy was proposed in [16]. Moreover, a system based on the technologies of big data and time series analysis for smart tourism is designed in [17].

In recent years, social network analysis recommendation strategies based on spatial or attribute information obtained with the social networking tools have been proposed due to the dramatic growth of computer sciences and technologies [18,19]. Additionally, because mobile service information provides extra references in tourists' behavior, many recommendation strategies based on the new technologies, such as RFID and augmented reality (AR), are widely used in tourism and leisure industries [20,21]. In addition, the novel recommendation system based on artificial intelligence to provide the personalized favorite music is introduced in [22].

This study designed and implemented a theme park tourist service (TPTS) system with the proposed personalized recommendation algorithm to enhance the visitor experiences. The TPTS system comprises four subsystems: the mobile app subsystem, the ticket-scanning subsystem, the detecting/counting subsystem, and the central subsystem. The mobile app subsystem is a tool for tourists to take advantage of diverse system services such as query of park information and reservation of preferred attractions. The ticket-scanning subsystem scans the tourist's reservation (booking) ticket and sends the ticket information to the central subsystem for further verification.

The detecting/counting subsystem aims to detect and compute the queue length and accumulated visit count of each attraction. The central subsystem performs necessary computing and database management of the proposed system. The proposed TPTS system introduces an innovative function called the personalized dynamic scheduling to offer tourists the customized best plans mainly according to the location of tourists, the queue length, the capacity, and the duration of an attraction. A tourist can use the mobile app to establish his/her own favorite attraction list, choose his/her own preferred attraction priority, and obtain the personalized recommended attraction to visit next. In addition, we design location-based dynamic map and attraction reservation functions into the TPTS system.

The main contributions of this study are summarized as follows. The TPTS system eliminates choice overload because it leaves only three commonly considered strategies (i.e., closest attraction first, shortest wait-time attraction first, and hottest attraction first) for tourists to decide on their own. In addition, the TPTS system removes the tourist's pressure caused by searching for only one preferred attraction from lengthy suggested attraction list because it offers only one recommended attraction at a time, and the recommendation is solely derived from the tourist's favorite attraction list. In addition, the recommended attraction is always the timely best attraction according to the current conditions of the attractions in the theme park, not an out-of-date or out-of-time result in a pre-arranged recommended list. Overall, this paper introduces the concept of location-based wait times, designing and implementing this idea into the prototype of our proposed system. This concept is not meant to replace or outperform the current design of Walt Disney World's or Universal Studio's system, but to present a novel idea which can be further integrated into their systems and help escalate the tour experiences even more.

The rest of this paper is organized as follows. Section 2 gives the basic idea of the proposed recommendation strategy and assumptions. Section 3 formulates the proposed personalized waiting time. Section 4 elaborates the design of the proposed theme park tourist service system. Section 5 presents the system implementation and the field testing results. Section 6 discusses two important issues, i.e., privacy and education, to be explored in the future. Section 7 provides concluding remarks.

2. Preliminaries

Figure 1 shows an example to illustrate the concept of the proposed personalized waiting time. Without the loss of generality, the next sessions of all attractions are assumed to start at the same time (i.e., all attractions have the same general waiting time). The current time and the next sessions of three attractions are 12:00 and 12:15, respectively. Thus, the general waiting time of three attractions is 15 min. Consider that there are two visitors who want to take Attraction A. They both find that the wait time is 15 min, and then decide to go to Attraction A right away. Assume that one of them (i.e., Visitor 1) should take 10 min to walk to it, while the other (Visitor 2) needs only 2 min to reach it. However, neither has the idea of how much time each of them needs to take to arrive at Attraction A if there is no mobility time information provided for them. They will just go to Attraction A (the same destination), with Visitor 1 finding that he/she can take the ride only after 5 min wait, while Visitor 2 realizing that she/he needs to wait for the ride for 13 min. Now, let us consider the feelings of the two visitors. In our opinion, people would like to keep moving to a certain destination rather than wait at a certain place, which is a common phenomenon that can be observed among car drivers or someone who is on a quest. Hence, we believe that Visitor 1's feeling would be better than Visitor 2's.

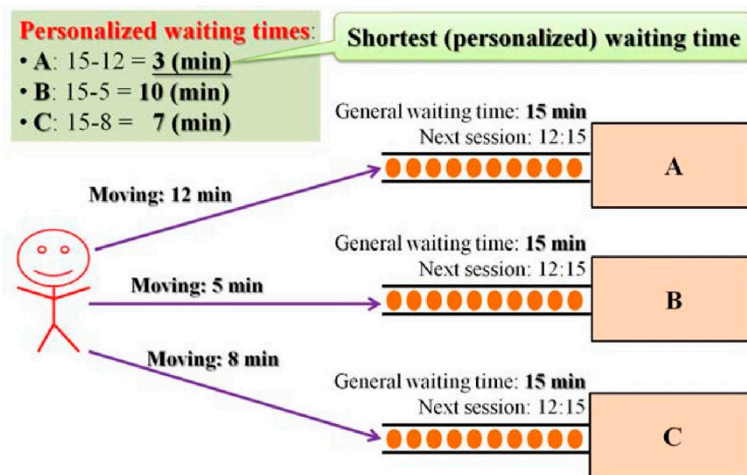


Figure 1. Basic concept of the proposed personalized waiting time.

The central idea of our study is the personalized waiting time, which is defined as the actual waiting time when the tourist arrives at an attraction and considered in the personalized dynamic scheduling function of the TPTS system. Recall that the waiting time considered in prior studies is the general waiting time, which is defined as the waiting time of the last tourist in the queue of length at an attraction. The general waiting time is computed based only on the queue length, the capacity, and the duration of an attraction, and totally irrelevant to any personal attribute of the tourist who is requesting any waiting-time related services. As shown in Figure 1, if considering the general waiting time only, the tourist has no preference among the three attractions and thus makes the final decision by himself/herself. The approaching times of the tourist moving towards Attractions A–C are, respectively, 12, 5, and 8 min. Thus, the actual waiting times when the tourist arrives at Attractions A–C are 3, 10, and 7 min, respectively. If the approaching times of the tourist moving towards these attractions are different, the tourist would feel or perceive that Attraction A costs the shortest waiting time among the three attractions. Therefore, the tourist is more likely to select Attraction A to visit next. This is how the personalized waiting time, taking the approaching time of the tourist into account, would improve the tourist’s perception of waiting as he/she arrives at the attraction.

This study considers the following assumptions to calculate the proposed personalized waiting time.

1. The inter-session time between operation sessions of the attraction is ignored.
2. The moving time of an attraction is derived as if the tourist starts to move to the attraction right after the tourist activates the personalized dynamic scheduling function.
3. The processing time at both the mobile app subsystem and the central subsystem as well as the network propagation delay between the two subsystems are ignored. In other words, the time when the tourist activates the personalized dynamic scheduling function at the mobile app subsystem and the time when the central subsystem receives the personalized dynamic scheduling request from the mobile app subsystem are considered as the same moment.

3. Personalized Waiting Time

This section provides the formulation of the proposed personalized waiting time. The notation used in the rest of the paper is summarized in Table A1.

As mentioned above, we argue that the personalized waiting time, taking the approaching time of the tourist into account, would improve the tourist’s perception of waiting when he/she arrives at the attraction. Specifically, the calculation of the personalized waiting time considers not only the queue length, the capacity, and the duration of an attraction, but also the moving time of the tourist

from his/her starting position to the attraction. Given n_i^Q , n_i^{Op} , and t_i^{Op} , the general waiting time of A_i , (t_i^{GW}) can be obtained from Equation (1).

$$t_i^{GW} = \left[\frac{n_i^Q}{n_i^{Op}} \right] \times t_i^{Op} \tag{1}$$

Definition 1 (Requesting Time). *The requesting time is defined as the time when the central subsystem receives the personalized dynamic scheduling request from the mobile app subsystem.*

Observed at the requesting time, this method considers the starting times of the current operation session and the next operation session. Specifically, the latter is defined as the immediate following session of the current operation session. Thus, we have $T_i^{Next} = T_i^{Curr} + t_i^{Op}$

To calculate the personalized waiting time, we have to determine the ideal session for the tourist. The ideal session of A_i is defined as the session which starts later than T^{Req} and can allow one visitor getting into A_i immediately after the $\left[\frac{n_i^Q}{n_i^{Op}} \right]$ th batch of visitors (in the queue of length $n_i^Q \geq 0$ observed at T^{Req}) is served. In other words, if a tourist who activates the personalized dynamic scheduling function arrives before and right at the start of the ideal session, he/she can get into A_i at the ideal session along with the last $n_i^Q - \left[\frac{n_i^Q}{n_i^{Op}} \right] \times n_i^{Op}$ visitors in the queue if $n_i^Q \geq 0$ observed at T^{Req} .

Definition 2 (Ideal Session Time). *The ideal session time is defined as the starting time of the ideal session.*

Theorem 1. $T_i^{Ideal} = T_i^{Next} + \left[\frac{n_i^Q}{n_i^{Op}} \right] \times t_i^{Op}$.

Proof. If all visitors in the queue plus one more visitor can be served in one single session (i.e., $n_i^Q + 1 \leq n_i^{Op}$), then the ideal session time is actually the starting time of the next operation session (i.e., $T_i^{Ideal} = T_i^{Next}$). On the other hand, if the visitors in the queue plus one more visitor cannot all be served in one single session (i.e., $n_i^Q + 1 > n_i^{Op}$), then ideal session time will be the starting time of a session later than the next session observed at T^{Req} and determined according to the queue length, capacity, and duration of A_i . As a result, we obtain $T_i^{Ideal} = T_i^{Next} + \left[\frac{n_i^Q}{n_i^{Op}} \right] \times t_i^{Op}$. □

Definition 3 (Moving Time). *The moving time is defined as the duration of the tourist moving from his/her location where he/she activates the personalized dynamic scheduling function to the location of an attraction.*

Definition 4 (Arrival Time). *The arrival time is defined as the time when the tourist arrives at an attraction.*

Given T^{Req} , we derive

$$T_i^{Arr} = T^{Req} + t_i^{Move} \tag{2}$$

Definition 5 (Recommendation Session Time). *The personalized waiting time is defined as the actual waiting time of the tourist when he/she arrives at an attraction.*

Theorem 2. If $T_i^{Arr} \leq T_i^{Ideal}$, then $t_i^{Wait} = (T_i^{Curr} - T^{Req}) + \left(1 + \left[\frac{n_i^Q}{n_i^{Op}} \right] \right) \times t_i^{Op} - t_i^{Move}$; otherwise, $t_i^{Wait} = (T_i^{Curr} - T^{Req}) + \left[\frac{T^{Req} - T_i^{Curr} + t_i^{Move}}{t_i^{Op}} \right] \times t_i^{Op} - t_i^{Move}$.

Proof. In the proposed system, $t_i^{Wait} = T_i^{Recomm} - T_i^{Arr}$ according to the definition of the recommend session time. Because T_i^{Arr} and T_i^{Ideal} directly influence T_i^{Recomm} , we consider the following two cases to derive T_i^{Recomm} , and further derive t_i^{Wait} . □

Case 1. The arrival time is earlier than or equal to the ideal session time (i.e., $T_i^{Arr} \leq T_i^{Ideal}$).

According to the above discussion, we infer that the tourist (who activates the personalized dynamic scheduling request at T^{Req}) can get into A_i immediately after the $\left\lceil \frac{n_i^Q}{n_i^{Op}} \right\rceil$ th batch of visitors in the queue is served for all $n_i^Q \geq 0$. Thus, $T_i^{Recomm} = T_i^{Ideal}$. Because $T_i^{Ideal} = T_i^{Next} + \left\lceil \frac{n_i^Q}{n_i^{Op}} \right\rceil \times t_i^{Op}$, T_i^{Recomm} can be derived from

$$T_i^{Recomm} = T_i^{Next} + \left\lceil \frac{n_i^Q}{n_i^{Op}} \right\rceil \times t_i^{Op} \tag{3}$$

According to Equations (2) and (3), t_i^{Wait} can be rewritten as $t_i^{Wait} = \left(T_i^{Next} + \left\lceil \frac{n_i^Q}{n_i^{Op}} \right\rceil \times t_i^{Op} \right) - (T^{Req} + t_i^{Move})$. Because $T_i^{Next} = T_i^{Curr} + t_i^{Move}$, we obtain

$$\begin{aligned} t_i^{Wait} &= (T_i^{Curr} + t_i^{Op}) + \left\lceil \frac{n_i^Q}{n_i^{Op}} \right\rceil \times t_i^{Op} - (T^{Req} + t_i^{Move}) \\ &= (T_i^{Curr} - T^{Req}) + \left(1 + \left\lceil \frac{n_i^Q}{n_i^{Op}} \right\rceil \right) \times t_i^{Op} - t_i^{Move} \end{aligned} \tag{4}$$

Case 2. The tourist's arrival time is later than the ideal session time (i.e., $T_i^{Arr} > T_i^{Ideal}$).

In this case, the tourist will miss the ideal session and has to wait only for a short period within the duration of A_i after he/she arrives, because all the visitors waiting in line observed at T^{Req} have been served before or at T_i^{Ideal} . Therefore, we have

$$T_i^{Recomm} = T_i^{Next} + \left\lceil \frac{T_i^{Arr} - T_i^{Next}}{t_i^{Op}} \right\rceil \times t_i^{Op} \tag{5}$$

According to Equations (2) and (5), t_i^{Wait} can be rewritten as $t_i^{Wait} = \left(T_i^{Next} + \left\lceil \frac{(T^{Req} + t_i^{Move}) - T_i^{Next}}{t_i^{Op}} \right\rceil \times t_i^{Op} \right) - (T^{Req} + t_i^{Move})$. Because $T_i^{Next} = T_i^{Curr} + t_i^{Move}$, we obtain

$$\begin{aligned} t_i^{Wait} &= (T_i^{Curr} + t_i^{Op}) + \left\lceil \frac{(T^{Req} + t_i^{Move}) - (T_i^{Curr} + t_i^{Op})}{t_i^{Op}} \right\rceil \times t_i^{Op} - (T^{Req} + t_i^{Move}) \\ &= (T_i^{Curr} - T^{Req}) + \left\lceil \frac{T^{Req} - T_i^{Curr} + t_i^{Move}}{t_i^{Op}} \right\rceil \times t_i^{Op} - t_i^{Move} \end{aligned} \tag{6}$$

4. TPTS System Design

Figure 2 shows the system architecture of the proposed TPTS system. The TPTS system consists of four subsystems: central subsystem, mobile app subsystem, ticket-scanning subsystem, and detecting/counting subsystem. Each subsystem includes many modules to perform the corresponding functions. The mobile app subsystem is a mobile app for visitors to offer the tourist services. The subsystem provides tourists with many interfaces to obtain the park information, personalized tour suggestion, booking record, and so on.

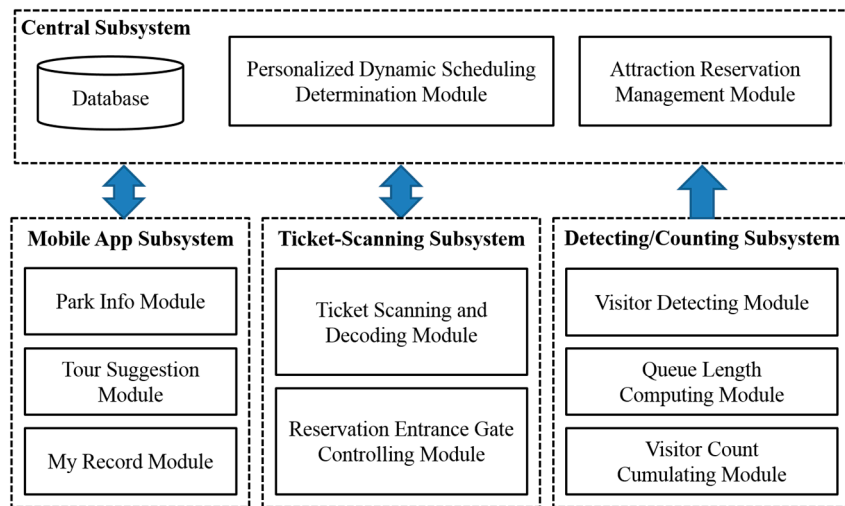


Figure 2. The TPTS system architecture.

4.1. Mobile App Subsystem

The mobile app subsystem (app) provides tourists with an integrated interface to take advantage of various tourist services. The visitors need only to download and install the app into their smartphones or tablet PCs and everything is on the go.

4.1.1. Park Info Module

This module provides the tourist with an interface to inquire general information about the theme park, such as news and tour guide. The tour guide includes the ticket prices, traffic, and opening hours of the park. It also provides the tourist with information about each attraction in the park, where the attractions are categorized by which theme area they reside. Furthermore, the search function is provided for the tourist to do a quick keyword search. In addition, the tourist can add attractions to his/her personal favorite or wish list (My Play List). This list can be used by the personalized dynamic scheduling function in the Tour Suggestion module.

4.1.2. Tour Suggestion Module

This module performs the central functions of the mobile app subsystem and provides the tourist with an interface to take advantage of these functions, including Personalized Dynamic Scheduling, Location-based dynamic map, and Attraction Reservation.

For ease of reserving attractions, we add the attraction reservation function embedded in the personalized dynamic scheduling function for the tourist to reserve the recommended attraction when the tourist obtains the recommended offered by the personalized dynamic scheduling function. Note that the service can also be independent of the personalized dynamic scheduling function according to the consideration of the industry of theme parks.

- Personalized Dynamic Scheduling

This function provides the tourist with a customized recommendation of the next attraction to visit according to the tourist’s location, favorite or wish attraction list (My Play List), preferred attraction priority (strategy), without the tourist making plans or too many decisions by himself/herself. To reduce the choice overhead of tourists, the TPTS system offers only one recommended attraction at a time. In addition, the recommended attraction is always the prompt best next attraction to visit according the tourist’s strategy, not an out-of-date or out-of-time result in a recommended list figured out or arranged in advance. Furthermore, we use the personalized waiting times instead of the general waiting times of attractions to find a recommended attraction in order to improve the tourist’s perception of waiting.

There are three choices of preferable strategies as the optimal solution: Closest First, Shortest Waiting Time First, and Hottest First. The “Closest First” strategy determines the attraction that is the closest one to the tourist. The “Shortest Waiting Time First” strategy determines the attraction having the shortest personalized waiting time. The “Hottest First” strategy determines the attraction with the maximum number of visits. Once the best next attraction is obtained, the central subsystem determines the recommended session time, the estimated moving time, and the estimated personalized waiting time in terms of the attraction. Then, the mobile app displays this information when receiving the response from the central subsystem.

- Attraction Reservation

This function provides the tourist with attraction reservation or booking services. The tourist can reserve attractions in advance, and enjoy his/her ride without the painful waiting in a long line. This function provides tourists with the attraction reservation service after the personalized dynamic scheduling function offers the recommended next attraction. When a tourist activates this function, the mobile app subsystem will proceed with the following steps.

- Step 1.** Request the central subsystem to return a list of bookable sessions of this attraction, each session with a current bookable quota, and display this list to the tourist.
- Step 2.** Hint the tourist to select the booking amount, such as three persons, after the tourist chooses a bookable session.
- Step 3.** Request the central subsystem to reserve the designated attraction after the tourist selects the booking amount and confirms his/her booking choice.
- Step 4.** Display the reservation result when receiving the response from the central subsystem. If the result is successful, the mobile app subsystem will store a digital booking ticket for later verification when the tourist arrives at the reservation entrance of the attraction.

- Location-based Dynamic Map

This function provides the recommended route(s), direction(s), estimated distance(s), and moving time(s) from the location of the tourist to his/her specified attraction on an electronic map, where the map can be easily zoomed in and out. This function can be easily accommodated via widely used Google Maps API, just as Walt Disney World and Universal Studios, Orlando have already adopted in their systems. Each attraction can be regarded as a point of interest (POI), and the spatial and related attribute of POI data are recorded previously through the concept of the GIS. The spatial relation between the tourist and POI can be found through the location service of the mobile device. As the tourist moves, the route will dynamically adjust according the tourist’s new location. With the location-based dynamic map function, the tourist will no longer get stressful because of misrouting or difficulty when seeking for an attraction on a static map. This function can link to the web map service (WMS) independently, thus we can choose different WMS sources for customized map designing in different theme park. Moreover, The TPTS can be utilized easily and adopted to the specific characteristics of each theme park.

4.1.3. My Record Module

This module provides the tourist with an interface to access his/her favorite or wish attraction list, booking tickets, and history of selected attractions using the My Play List, My Booking Tickets, and My Play Record items, respectively. The My Play List shows the attractions selected by tourists themselves, and is the operation basis for the personalized dynamic scheduling function. Particularly, the personalized dynamic scheduling function finds the recommended attraction only from this list. We can select the attractions optionally or modify directly in this list. The My Booking Tickets shows all the booking tickets. When a tourist arrives at the reservation entrance, he/she can draw out the corresponding ticket from this list and show the ticket to the ticket-scanning subsystem for further

verification. The My Play Record displays a chronological history of attractions ever selected by the tourist.

4.2. Ticket-Scanning Subsystem

This subsystem aims at scanning the tourist's booking ticket, requesting the central subsystem to verify the validity of this booking ticket, and trigger the reservation entrance gate of the attraction. Recall that there are two modules in this subsystem, briefly discussed in the following subsections.

4.2.1. Ticket Scanning and Decoding Module

This module scans the tourist's digital booking ticket, such as a QR code shown on his/her smartphone screen, decodes the ticket information, and requests the central subsystem to verify this booking ticket according to the ticket information. Upon receipt of the response from the central subsystem about the validity of the ticket, this module hands over the proceeding task to the reservation entrance gate controlling module.

4.2.2. Reservation Entrance Gate Controlling Module

This module triggers the reservation entrance gate to open up for the tourist to pass if the tourist's ticket is valid according to the response from the central subsystem. If the ticket is not valid, this module can instead show an error message to inform the tourist. For testing and demonstration purpose, the implementation of the prototype can be either software- or hardware-based. For example, we can implement this module simply as a software which indicates whether a visitor is allowed to enter the reservation entrance gate based on the response of the central subsystem. On the other hand, we can also introduce micro-controllers (e.g., Arduino or Raspberry Pi) and actuators into the prototype implementation, in which they shall act like the reservation entrance gate, behaving according to the response from the central subsystem.

4.3. Detecting/Counting Subsystem

This subsystem is responsible for detecting tourist penetration through the entrance of an attraction, calculating the queue length and the number of visits to the attraction, and sending this information to the central subsystem at appropriate timings. Note that the queue length and the number of visits are two significant parameters to the personalized dynamic scheduling, where the queue length is for calculating the personalized waiting time, and the number of visits determines how popular ("hot") the attraction is. The three modules in this subsystem are described as follows.

4.3.1. Visitor Detecting Module

This module detects the tourist penetration through the entrance of an attraction, and notifies the Queue Length Computing module and the Visitor Count Cumulating module to calculate the queue length and the number of visits.

4.3.2. Queue Length Computing Module

This module computes the queue length of an attraction when a tourist passes through the entrance (notified by the Visitor Detecting module) or at every turn of attraction operation (i.e., when the attraction finishes a round of operation and tourists in the waiting line can move into the attraction to have a ride). This module also sends the value of the queue length to the central subsystem for database updating at appropriate timings. If we require a highly real-time value of queue length, we can have this module send the value of queue length every time this value is changed. Otherwise, we can have this module send this value less frequently.

4.3.3. Visitor Count Cumulating Module

This module accumulates the number of visits (visitor count) to an attraction when a tourist passes through the entrance of the attraction (notified by the Visitor Detecting Module). In addition, this module sends the visitor count to the central subsystem for database updating at appropriate timings. If we require a highly real-time visitor count, we can have this module send the count every time this value is changed. Otherwise, we can have this module send this value less frequently.

4.4. Central Subsystem

The central subsystem performs the kernel computing and database management of the TPTS system. With respect to the mobile app subsystem, the central subsystem accepts the park information inquiry, personalized dynamic scheduling, and attraction reservation requests from the mobile app subsystem. After processing these requests, the central subsystem returns corresponding responses to the mobile app subsystem. With respect to the ticket-scanning subsystem, the central subsystem accepts the ticket verification request from the ticket-scanning subsystem, verifies the validity of the ticket, and responds the result to the ticket-scanning subsystem. With respect to the detecting/counting subsystem, the central subsystem accepts the notifications of the values of the queue length and the visit count from the detecting/counting subsystem, and updates the database accordingly. Besides the database, this subsystem consists of the Personalized Dynamic Scheduling Determination module and the Attraction Reservation Management module.

4.4.1. Personalized Dynamic Scheduling Determination Module

This module provides kernel computing to the personalized dynamic scheduling function of the TPTS system. Specifically, this module is responsible for calculating the personalized waiting times and recommended session times of attractions, comparing and finding the attraction with the shortest waiting time among the attractions, and finding the most popular (“hottest”) attraction according to the visit counts of attractions recorded in the database.

Under all three strategies, this module needs to determine the personalized waiting time and the recommended session time prior to performing the personalized dynamic scheduling. The two values are required to inform the tourist how long he/she may wait when arriving at the attraction and which session to take. Moreover, in the Shortest Waiting Time First strategy, the personalized waiting times of a group of attractions are required to determine which attraction has the shortest personalized waiting time. To conclude, the calculation of personalized waiting times is crucial to the personalized dynamic scheduling function.

Recall that the central subsystem in the TPTS system supports the personalized dynamic scheduling function including three strategies. When receiving a personalized dynamic scheduling request from the mobile app subsystem, the central subsystem determines which strategy the tourist chooses and proceeds accordingly. The steps that the central subsystem performs for three strategies are, respectively, as follows.

- Closest First strategy
 - Step 1.** Calculate the personalized waiting time and recommended session time of the closest attraction based on its moving time found in the received personalized dynamic scheduling request.
 - Step 2.** Send the personalized waiting time and recommended session time of the closest attraction to the mobile app subsystem.
- Shortest Waiting Time First strategy

- Step 1.** Calculate the personalized waiting time and recommended session time of each of attractions based on their moving times found in the received personalized dynamic scheduling request.
 - Step 2.** Find the attraction with the shortest personalized waiting time.
 - Step 3.** Send the attraction ID/name that has the shortest personalized waiting time as well as its personalized waiting time and recommended session time to the mobile app subsystem.
- Hottest Fist strategy
 - Step 1.** Find the most popular (hottest) attraction (the implemented prototyping TPTS system regards the attraction with the highest number of visits as the hottest attraction).
 - Step 2.** Calculate the personalized waiting time and recommended session time of the hottest attraction based on its moving time found in the received personalized dynamic scheduling request.
 - Step 3.** Send the attraction ID/name that is the most popular as well as its personalized waiting time and recommended session time to the mobile app subsystem.

The case study for the functionality of the scheduling determination module is presented in Section 5.2.2.

4.4.2. Attraction Reservation Management Module

This module provides kernel handling to attraction reservation and booking ticket verification. The former handles attraction reservation or booking requests from the mobile app subsystem, while the latter verifies the validity of booking tickets.

- Attraction Reservation Handling

When receiving a booking-attraction request of a designated attraction from the mobile app subsystem, the central subsystem will proceed with the following steps. Figure 3 illustrates the detailed message flow chart between the mobile app subsystem and the central subsystem regarding attraction reservation.

- Step 1.** Draw out all bookable sessions of this attraction and the current bookable quotas of these sessions from the database.
- Step 2.** Return the information drawn in Step 1 to the mobile app subsystem via a booking-attraction response.
- Step 3.** When receiving the booking-session-amount request from the mobile app subsystem, the central subsystem will insert a new booking record of the designated attraction into the database and update related field(s) in the database.
- Step 4.** Send a booking confirmation message to the mobile app subsystem.

- Booking Ticket Verification

When receiving a ticket verification request from the ticket-scanning subsystem, the central subsystem will proceed with the following steps. The detailed message flow chart between the ticket-scanning subsystem and the central subsystem is shown in Figure 4.

- Step 1.** Compare the data in the ticket verification request with the booking records in the database; if no corresponding record exists, the central subsystem will return a ticket verification response with answer “invalid” to the ticket-scanning subsystem and end the verification process; otherwise, go to Step 2.

- Step 2.** Check if the tourist arrives during the appointed period (e.g., within 15 min before the reserved session starts); if not, the central subsystem will return a ticket verification response with answer “invalid” to the ticket-scanning subsystem and end the verification process; otherwise, go to Step 3.
- Step 3.** Recognize this ticket as valid, update related fields in the database, and return a ticket verification response with answer “valid” to the ticket-scanning subsystem.

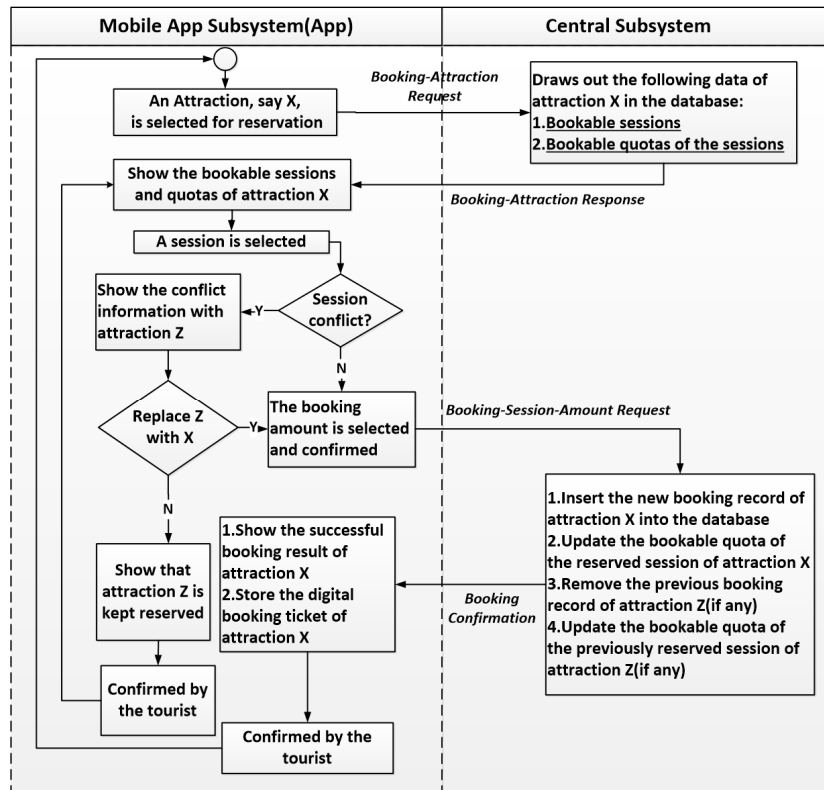


Figure 3. Message flow chart of attraction reservation.

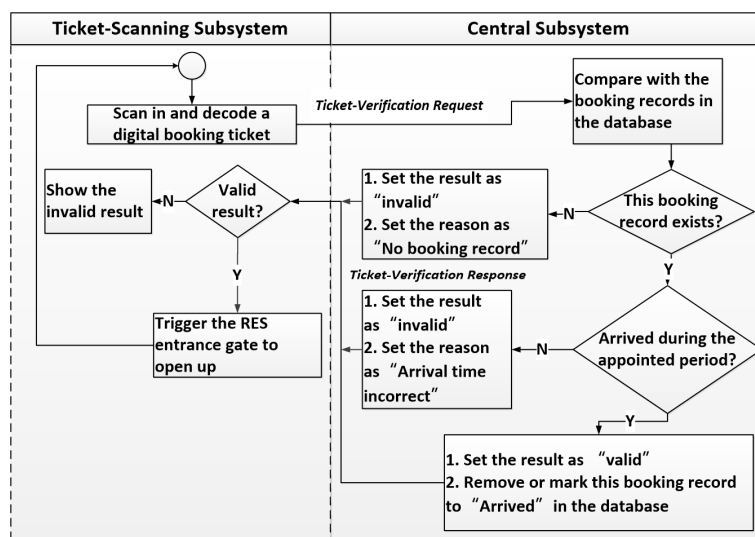


Figure 4. Message flow chart of verification of a booking ticket.

5. System Implementation and Field Testing

This section mentions the implementation issues of the proposed TPTS system, and then shows the experimental result.

5.1. Implementation Details

This section, respectively, presents the hardware and software components used to implement the four subsystems of the proposed TPTS system.

5.1.1. Mobile App Subsystem

The mobile app subsystem is developed in Android platform using Eclipse integrated development environment (IDE) with Android SDK. For the proposed three strategies, we use Google Maps Directions API to acquire the moving times of the tourist from his/her GPS coordinates to all the attractions' GPS coordinates in My Play List. Because Google Maps Directions API considers the moving speed as being constant, the distances are directly proportional to the moving times.

5.1.2. Ticket-Scanning Subsystem

The ticket-scanning subsystem is implemented using Visual Studio C#, hosted on a notebook with an embedded webcam. The booking tickets are generated in the form of QR codes. Because the digital booking tickets are in the form of QR codes, the implemented program running on the laptop includes a QR code reader to scan and decode the tourist's QR code booking ticket. In the subsystem, the reservation entrance gate is emulated by a program which exhibits a virtual gate on the notebook screen. The program also includes a virtual gate to show on the laptop screen. When scanning in a QR code booking ticket, the program decodes the ticket information and sends the information to the central subsystem for ticket verification. Upon receiving the verification result from the central subsystem, the virtual gate would be shown as opening up if the result is valid, or shown as keeping closed if the result is invalid.

5.1.3. Detecting/Counting Subsystem

The detecting/counting subsystem consists of a programmable Arduino UNO microcontroller board, an infrared sensor and a notebook laptop. We used the Arduino UNO board and the infrared sensor to emulate the Visitor Detecting Module. Moreover, we used the notebook laptop to run processes emulating the Queue Length Computing Module and the Visitor Count Cumulating Module. To provide highly real-time values of the queue length and the visitor count, we had the notebook laptop send these values to the central subsystem every time the values changed.

5.1.4. Central Subsystem

The central subsystem was implemented using Visual Studio C#, hosted on a desktop PC running Windows. Microsoft SQL Server served as the system database on the same desktop PC.

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.

5.2. Testing Results

We conducted numerous experiments to test the major functions of the proposed TPTS system. Figure 5 shows the testing field of this study. The testing field is located around Minghsin University of Science and Technology. Suppose that seven attractions (denoted as red dots) and one tourist are located in the testing field. The coordinates of the attractions were previously acquired by GPS positioning. We selected three of them for testing and defined the condition of attraction, including the

number of sessions in one attraction, capacity of tourists, duration time in one session, and the number of visitors in the attraction. The location-based dynamic map was produced via the Google Maps API.



Figure 5. Field of the testing.

5.2.1. Attraction Information Displaying

To verify the operation of communication between the mobile app subsystem and the central subsystem, we selected an attraction, and then checked the content displayed on the screen. Figure 6 shows the testing result. Compared with the content in the database, we verified that the mobile app subsystem can access the database in the central subsystem and show the result correctly.



Figure 6. Experimental results of attraction information display: (a) attraction list; and (b) results.

5.2.2. Personalized Dynamic Scheduling

We considered four attractions, naemly Racing Cars, Spinning Tea Cups, Merry-Go-Round, and Mountain Adventures, to test the function of personalized dynamic scheduling, as shown in Figure 5. Assume that the location (i.e., GPS coordinates) of the tourist is (N 24.86284°, E 120.99045°). In addition, three attractions, i.e., Racing Cars, Spinning Tea Cups, and Merry-Go-Round, are assumed to be added into My Play List. The parameters of these attractions for testing are listed in Table 1.

Table 1. Attraction parameters in the experiments.

Attraction Name	GPS Coordinates (N, E)	Capacity (Visitors/Session)	Duration (Min/Session)	Popularity (Visits)
Racing Cars	(24.86332°, 120.98701°)	10	20	2323
Spinning Tree Cups	(24.86316°, 120.98852°)	10	20	12,120
Merry-Go-Round	(24.86325°, 120.99076°)	10	20	800

- Closest First strategy

Suppose that the personalized dynamic scheduling function with the Closest First strategy was activated at 12:07, and the queue lengths of three attractions were all 20 visitors. According to the result of Google Maps Directions API, we obtained the distances between our location and Racing Cars, Spinning Tea Cups, and Merry-Go-Round as 450 m, 220 m, and 55 m, respectively. Obviously, Merry-Go-Round is the closest attraction and should be recommended. In our experiments, the moving time of the tourist is 1 min because the walking time of tourists are assumed to be constant, and the queue length of the attraction is assumed to be 32 visitors. Thus, we obtained the waiting time and the recommended session time as 72 min and 13:20, respectively. As shown in Figure 7, the result verifies that the personalized dynamic scheduling function actually recommended the closest attraction (Merry-Go-Round in this experiment) when we selected the “Closest First” strategy. The result also confirms that the recommended session time, moving time, and personalized waiting time are all correctly calculated.

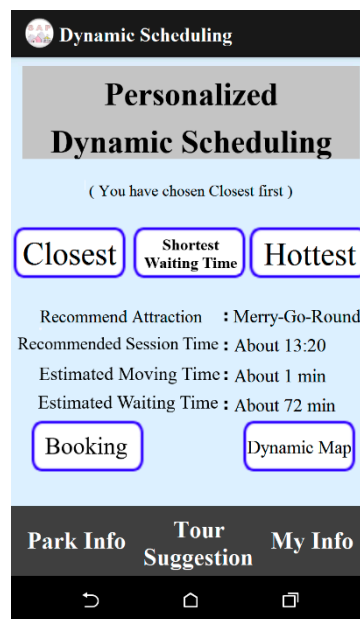


Figure 7. Experimental results of the Closest First strategy.

- Shortest Waiting Time First strategy

Suppose that we activated the personalized dynamic scheduling function with the Shortest Waiting Time First strategy at 12:10, and the queue lengths of three attractions were all 20 visitors. To determine the recommended next attraction, we calculated the recommended session time, moving time, and waiting time, as listed in Table 2. Note that this strategy prefers the attraction having the shortest personalized waiting time. Thus, Racing Car should be recommended because it had the shortest personalized waiting time (65 min).

Table 2. Calculation results for the Shortest Waiting Time First strategy.

Attraction Name	Next Session Time	Personalized Waiting Time (Min)	Recommended Session Time
Racing Cars	12:20	65	13:20
Spinning Tree Cups	12:20	67	13:20
Merry-Go-Round	12:20	69	13:20

Figure 8 illustrates the testing result, which verifies that the personalized dynamic scheduling function actually recommended the attraction with the shortest personalized waiting time (Racing Cars in this experiment) when we considered the “Shortest Waiting Time First” strategy. Moreover, the recommended session time, moving time, and personalized waiting time were all correctly determined.

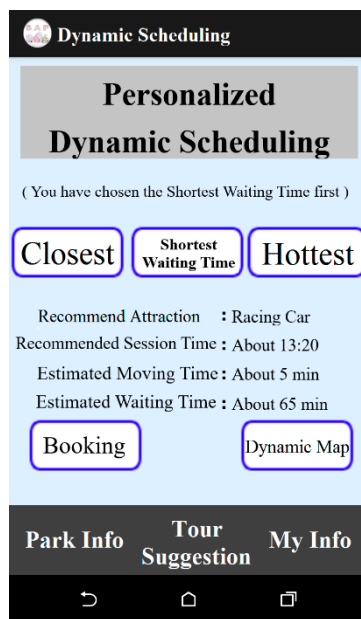


Figure 8. Experimental results of the Shortest Waiting Time First strategy.

- Hottest First strategy

Suppose that the personalized dynamic scheduling function with strategy “Hottest First” was activated at 12:15. The queue lengths of all attractions were assumed to be 20 visitors. According to the database in the central subsystem, we obtained that Spinning Tea Cups should be recommended because it had the largest visit count. Moreover, we determined the recommended session time, moving time, and waiting time as 13:20, 3 min, and 62 min, respectively. Figure 9 shows the testing result, which validates that the personalized dynamic scheduling function actually recommended the hottest attraction (Spinning Tea Cups in this experiment) when we selected the “Hottest First” strategy. The result also validates that the recommended session time, moving time, and personalized waiting time were all correctly calculated.

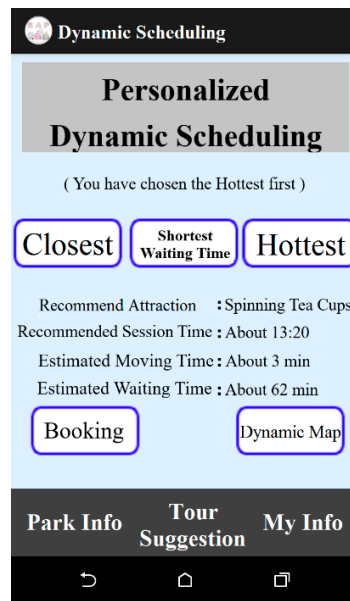


Figure 9. Experimental results of the Hottest First strategy.

The above experiments are not exhaustive because they only cover case $T_i^{Arr} \leq T_i^{Ideal}$. We further verified whether the personalized dynamic scheduling function correctly computes the estimated personalized waiting time and recommended session time under case $T_i^{Arr} > T_i^{Ideal}$. We added Racing Cars and Mountain Adventures into My Play List, and then activated the personalized dynamic scheduling function with the Hottest First strategy at 12:24. We determined the next session time as 12:40, the personalized waiting time as 6 min, and the recommended session time as 12:50. Figure 10 shows the result derived by the proposed TPTS system. Results validate that the personalized dynamic scheduling function correctly calculated the personalized waiting time and recommended session time.

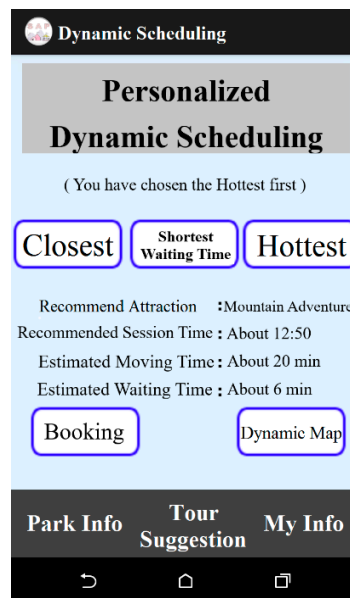


Figure 10. Experimental results for case $T_i^{Arr} > T_i^{Ideal}$.

5.2.3. Attraction Reservation Scheme

We tested the function of attraction reservation using the recommended attraction result shown in Figure 11a. This result was provided by the personalized dynamic scheduling function, and we started

to reserve this recommended attraction. The app showed the information including the available session and capacity for visitors, as shown in Figure 11b. Once selecting the preferable session, we provided the number of visitors to reserve, as shown in Figure 11c.

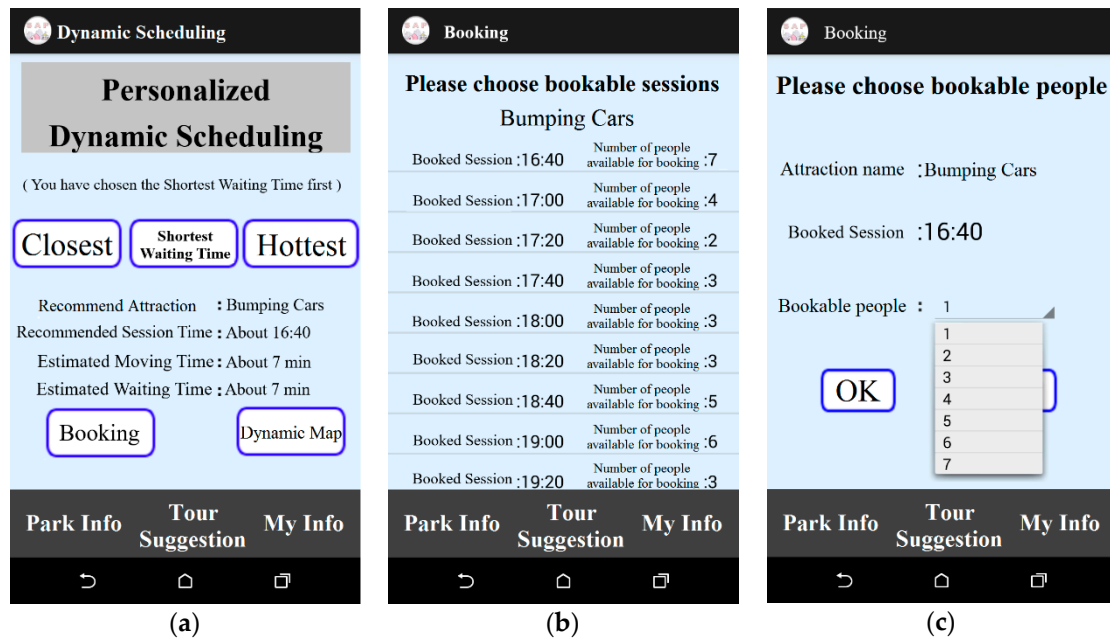


Figure 11. Testing steps of the attraction reservation function: (a) example result of the recommended attraction; (b) list of available sessions and capacities; and (c) selection of the number of visitors to book.

Figure 12a shows the result of attraction reservation if this booking is validated. In addition, our mobile app immediately generated a personalized booking ticket in the form of a QR code, as shown in Figure 12b. This experiment confirms that the attraction reservation function works correctly.



Figure 12. Testing result of the attraction reservation function: (a) result of the successful reservation of attraction; and (b) the booking ticket of the visitor.

6. Future Works

A noteworthy issue concerning privacy might arise due to the functionality of the attraction reservation. Regarding this functionality, a tourist ID shall be included in the process and stored in the database of the central subsystem. This tourist ID could be the cell phone number or some kind of identifier of a tourist who reserves an attraction, which is definitely part of the tourist's personal information. Therefore, the protection of the tourist ID is inevitably required to be observed. There are various kinds of security mechanisms which can be implemented in the proposed system to ensure the privacy of the tourist ID, such as the dual authentication method proposed by [23], and we leave this issue as part of our future work.

In addition, the proposed system can not only work for the amusement parks, but, with necessary modifications, also be a good assistant tool for education. In the context of current trend of flipped education [24,25], the proposed system can be built in a campus, where each classroom or campus facility is regarded as an attraction. Students have to group themselves into several teams and collaborate to tackle each problem stated at each attraction with all they have learned from multiple disciplines. The first team that conquers all the problems wins the battle. To name this example, it blends collaborative learning, problem-based learning, and game-based learning, which have been proposed to transform the current education system. Our system can play a role in this context. For example, each team can use a smartphone with our mobile app to search for the closest or shortest-waiting-time attraction and move there according to the location-based dynamic map. This can be a prospective future work and potentially profitable.

7. Conclusions

This paper has proposed a personalized recommendation strategy to develop a service system for theme parks to provide tourists with a more relaxing and easier tour experience. The proposed system, called the TPTS system, consists of a mobile app subsystem, a ticket-scanning subsystem, a detecting/counting subsystem, and a central subsystem. A specific novel function called the location-based dynamic scheduling function is designed to provide tourists with customized tour suggestions (recommended attraction with recommended session time) based on each tourist's favorite attractions, preferred attraction priority, and locations, without tourists making various tiresome decisions by themselves. Serving as a location-based service to improve the tourist's perception of waiting, the proposed personalized dynamic scheduling function considers not only the general wait time of current queue but also the estimated moving time of the tourist based on his or her location to yield a location-based wait time. In addition, for the completeness of the TPTS system, we also designed the attraction reservation and booking-ticket verification schemes as supplementary functions. Furthermore, the mobile app of TPTS system gives an integrated, easy-to-use interface for tourists who are familiar with smartphones or tablets nowadays. The design and implementation of the TPTS system is described in this paper, and the demonstrations show that the proposed TPTS system functions correctly.

Author Contributions: S.-S.W. proposed the research topic and gave the conceptualization. F.-C.Y. introduced and formulated the central concept. P.-C.L. and F.-C.Y. jointly designed the overall architecture and algorithms. P.-H.K. and P.-C.L. jointly designed the experiments and performed the validation of testing results. F.-C.Y. and S.-S.W. jointly wrote the article.

Funding: This research was funded by Research funded by Ministry of Science and Technology, Taiwan (MOST 104-2622-E-159-008-CC3).

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. Notations used in the paper.

Notation	Description
M	The number of attractions
A_i	The i th attraction ($i = 1, 2, \dots, M_i$)
n_i^Q	The queue length of A_i
n_i^{Op}	The capacity of A_i
t_i^{Op}	The duration of A_i
t_i^{GW}	The general waiting time of A_i
T^{Req}	The requesting time
T_i^{Curr}	The starting time of the current operation session of A_i
T_i^{Next}	The starting time of the next operation session of A_i
T_i^{Ideal}	The ideal session time of A_i
t_i^{Move}	The moving time of A_i
T_i^{Arr}	The arrival time of A_i
T_i^{Recomm}	The recommendation session time of A_i
t_i^{Wait}	The personalized waiting time of A_i

References

- Marr, B. Walt disney parks and resorts: How big data is transforming our family holidays. In *Big Data in Practice*; Marr, B., Ed.; John Wiley and Sons, Ltd.: Hoboken, NJ, USA, 2016; Chapter 33; pp. 211–215.
- Chernev, A.; Böckenholt, U.; Goodman, J. Choice overload: A conceptual review and meta-analysis. *J. Consum. Psychol.* **2005**, *25*, 333–358. [[CrossRef](#)]
- Scheibehenne, B.; Greifeneder, R.; Tdd, P.M. Can there ever be too many options? A meta-analytic review of choice overload. *J. Consum. Res.* **2010**, *37*, 409–425. [[CrossRef](#)]
- Davis, M.M.; Vollmann, T.E. A framework for relating waiting time and customer satisfaction in a service operation. *J. Serv. Mark.* **1990**, *4*, 61–69. [[CrossRef](#)]
- Ryan, G.; Hernandez-Maskivker, G.; Valverde, M.; Pamies-Pallise, M. Challenging conventional wisdom: Positive waiting. *Tour. Manag.* **2018**, *64*, 64–72. [[CrossRef](#)]
- Bielen, F.; Demoulin, N. Waiting time influence on the satisfaction-loyalty relationship in services. *Manag. Serv. Qual.* **2007**, *17*, 174–193. [[CrossRef](#)]
- Li, W.L. Impact of waiting time on tourists satisfaction in a theme park: An empirical investigation. In Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management, Macao, China, 7–10 December 2010.
- Hale, G.B.; Stafford, D.A.; Schwalb, A.; Craven, T.; Schweizer, K.W. Management of the Flow of Persons in Relation to Centers of Crowd Concentration via Wireless Control. U.S. Patent 7,532,941, 12 May 2009.
- Lu, J.; Wu, D.; Mao, M.; Wang, W.; Zhang, G. Recommender system application developments: A survey. *Decis. Support Syst.* **2009**, *74*, 12–32. [[CrossRef](#)]
- Yao, L.; Sheng, Q.Z.; Ngu, A.H.; Yu, J.; Segev, A. Unified collaborative and content-based web service recommendation. *IEEE Trans. Serv. Comput.* **2015**, *8*, 453–466. [[CrossRef](#)]
- Loh, S.; Lorenzi, F.; Saldaña, R.; Lichnow, D. A tourism recommender system based on collaboration and text analysis. *Inf. Technol. Tour.* **2003**, *6*, 157–165. [[CrossRef](#)]
- Ravi, L.; Vairavasundaram, S. A collaborative location based travel recommendation system through enhanced rating prediction for the group of users. *Comput. Intell. Neurosci.* **2016**, *2016*, 1291358. [[CrossRef](#)] [[PubMed](#)]
- Malle, B.; Giuliani, N.; Kieseberg, P.; Holzinger, A. The more the merrier-federated learning from local sphere recommendations. In Proceedings of the International Cross-Domain Conference, Reggio, Italy, 29 August–1 September 2017; pp. 367–374.
- Console, L.; Torre, I.; Lombardi, I.; Gioria, S.; Surano, V. Personalized and adaptive services on board a car: An application for tourist information. *J. Intell. Inf. Syst.* **2003**, *21*, 249–284. [[CrossRef](#)]
- Gandomi, A.; Haider, M. Beyond the hype: Big data concepts, methods, and analytics. *Int. J. Inf. Manag.* **2015**, *35*, 137–144. [[CrossRef](#)]

16. Al-Hassan, M.; Lu, H.; Lu, J. A semantic enhanced hybrid recommendation approach: A case study of e-Government tourism service recommendation system. *Decis. Support Syst.* **2015**, *72*, 97–109. [[CrossRef](#)]
17. Chen, W.C.; Chen, W.H.; Yang, S.Y. A big data and time series analysis technology-based multi-agent system for smart tourism. *Appl. Sci.* **2018**, *8*, 947. [[CrossRef](#)]
18. Jiang, S.; Qian, X.; Mei, T.; Fu, Y. Personalized travel sequence recommendation on multi-source big social media. *IEEE Trans. Big Data* **2016**, *2*, 43–56. [[CrossRef](#)]
19. Sun, X.; Huang, Z.; Peng, X.; Chen, Y.; Liu, Y. Building a model-based personalized recommendation approach for tourist attractions from geotagged social media data. *Int. J. Digit. Earth* **2018**, 1–18. [[CrossRef](#)]
20. Tsai, C.Y.; Chung, S.H. A personalized route recommendation service for theme parks using RFID information and tourist behavior. *Decis. Support Syst.* **2012**, *52*, 514–527. [[CrossRef](#)]
21. Jung, T.; Chung, N.; Leue, M.C. The determinants of recommendations to use augmented reality technologies: The case of a Korean theme park. *Tour. Manag.* **2015**, *49*, 75–86. [[CrossRef](#)]
22. Abdul, A.; Chen, J.; Liao, H.Y.; Chang, S.H. An emotion-aware personalized music recommendation system using a convolutional neural networks approach. *Appl. Sci.* **2018**, *8*, 1103. [[CrossRef](#)]
23. Holzinger, K.; Holzinger, A.; Safran, C.; Koiner-Erath, G.; Weippl, E. Use of wiki systems in archaeology: Privacy, security and data protection as key problems. In Proceedings of the International Conference on e-Business, Athens, Greece, 26–28 July 2010.
24. Holzinger, K.; Lehner, M.; Fassold, M.; Holzinger, A. Archaeological scavenger hunt on mobile devices: From e-education to e-Business: A triple adaptive mobile application for supporting experts, tourists and children. In Proceedings of the International Conference on e-Business, Seville, Spain, 18–21 July 2011.
25. Holzinger, K.; Koiner-Erath, G.; Kosec, P.; Fassold, M.; Holzinger, A. ArchaeoApp rome edition (aare): Making invisible sites visible—E-business aspects of historic knowledge discovery via mobile devices. In Proceedings of the International Conference on Data Communication Networking, e-Business and Optical Communication Systems, Rome, Italy, 23–17 July 2012.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).