

React Native and Android Mobile Apps for Smart Tourism Information Service to FITs

Hyun-Ji Cho* Jin-Yi Lee* Tae-Rang Park* and Jeong-Woo Jwa**

**Student, **Professor, Department of Telecommunication Eng., Jeju National University, Jeju, Korea*
e-mail : {whgus613, jin2414487, tr1991, lcr02}@jejunu.ac.kr

Abstract

We develop a smart tourism information system that provides smart tourism services to free independent tourists (FITs) through various content distribution channels such as mobile apps, Instagram, YouTube, and chatbots. The smart tourism information system provides location and storytelling-based tourism information, accommodation, restaurant information, and recommended travel products so that tourists can create a travel itinerary based on personalized situation awareness. The smart tourism information system also provides smart tourism services using commercial maps, navigation, and weather forecast APIs from the Korea Meteorological Administration to provide smart tour guide services to tourists who travel according to the travel itinerary. In this paper, we develop the React Native app that provides smart tourism services provided by the smart tourism information system. The smart tourism React Native app has implemented two methods: a method that directly connects to the smart tourism information system, and a method that provides services by interworking through the GraphQL Query Language developed by META (Facebook). The smart tourism React Native app implements OSMU (One Source Multi-use) by providing tourism information from mobile apps, photos from Instagram, and drone videos from YouTube as an integrated UI.

Keywords: *React Native App, Android App, Smart Tourism, Free Independent Travelers (FITs), One Source Multi-use (OSMU).*

1. Introduction

Smart tourism [1, 3] uses the Internet of Things, communication systems, big data, and artificial intelligence technologies to create a travel itinerary for tourists and provide a smart tour guide service while traveling. We developed a smart tourism Android app and the smart tourism information system that provides tourism information and recommended travel products to tourists, and tourists use this information to create personalized travel itinerary plans [4]. In the smart tourism information system, tourism information is implemented by classifying tourist attractions into plane (zone), line, and point (POI) types. Tourism information of plane and line type tourist destinations with waypoints is implemented to have hierarchical relationships. The smart tourism Android app displays tourism information according to the hierarchical relationships and the travel order of the waypoints. In addition, the line type tourist destination includes the pedestrian network composed of nodes and links built with the geographic information system (GIS) using

Manuscript Received: February. 26, 2022 / Revised: March. 3, 2022 / Accepted: March. 6, 2022

Corresponding Author: lcr02@jejunu.ac.kr

Tel: +81-64-754-3638, Fax: +81-64-754-3610

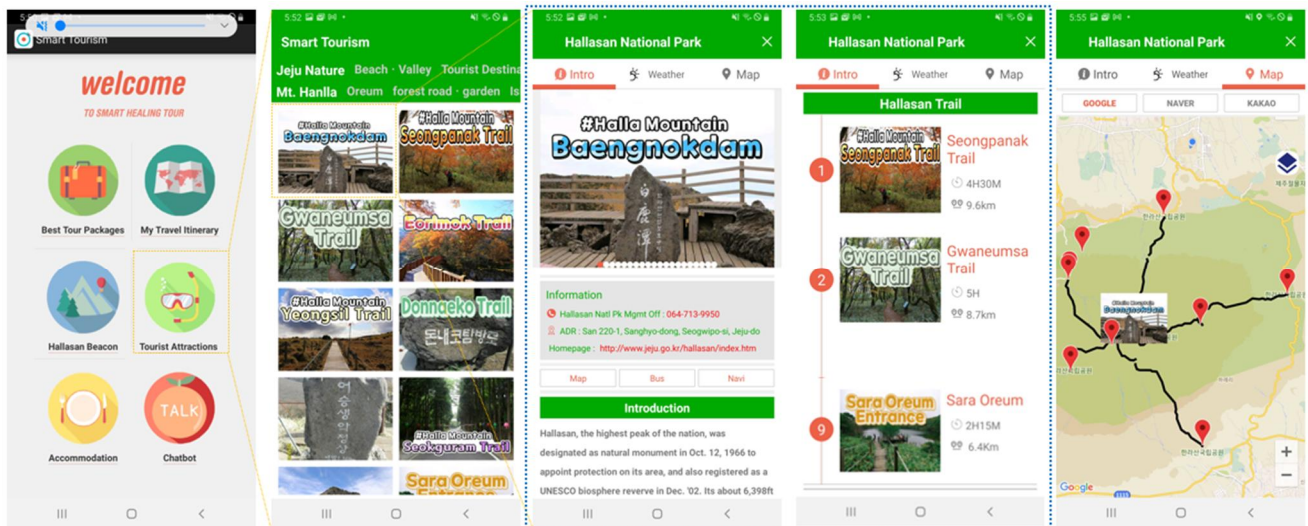
Professor, Department of Telecommunication Eng., Jeju National University, Korea

GPS trajectory data collected in the field [5]. In the smart tourism information system, tourism information consists of geo-tagged storytelling, photos, VR images, and drone images. We also provide Instagram photos and YouTube video services using photos and video data of tourist destinations in the smart tourism information system.

We developed a server-based multilingual TTS system to provide a location-based smart tour guide service to traveling tourists using geo-tagging storytelling data and photo data [6]. The mobile app provides tourists with voice guidance received from the TTS server using the location information of GPS and beacons. Tourists can travel safely while receiving necessary tourist information through voice guidance according to their own travel itinerary. We developed a smart tourism chatbot to efficiently provide tourism information with a smart tourism app. In this paper, we develop the React Native app that provides tourism information to tourists in conjunction with a smart tourism information system. We create the integrated UI to provide Instagram photos and YouTube drone video services in the React Native app. We create thumbnails for the smart tourism app, Instagram, and YouTube using the same template in Adobe Premiere Pro. We also develop an advanced Android app for tourism information with hierarchical relationships and KML data on Google Maps.

2. Tourism Information with Hierarchical Relationships in the Smart Tourism App

The smart tourism service platform provides tourism information and recommended travel products to tourists preparing for a trip, and tourists can use this to create a personalized itinerary of travel products based on situational awareness. In this paper, we develop an Android app [7] for providing smart tourism service and a React Native app [8] for iOS. Figure 1 shows the screen of the smart tourism Android app.

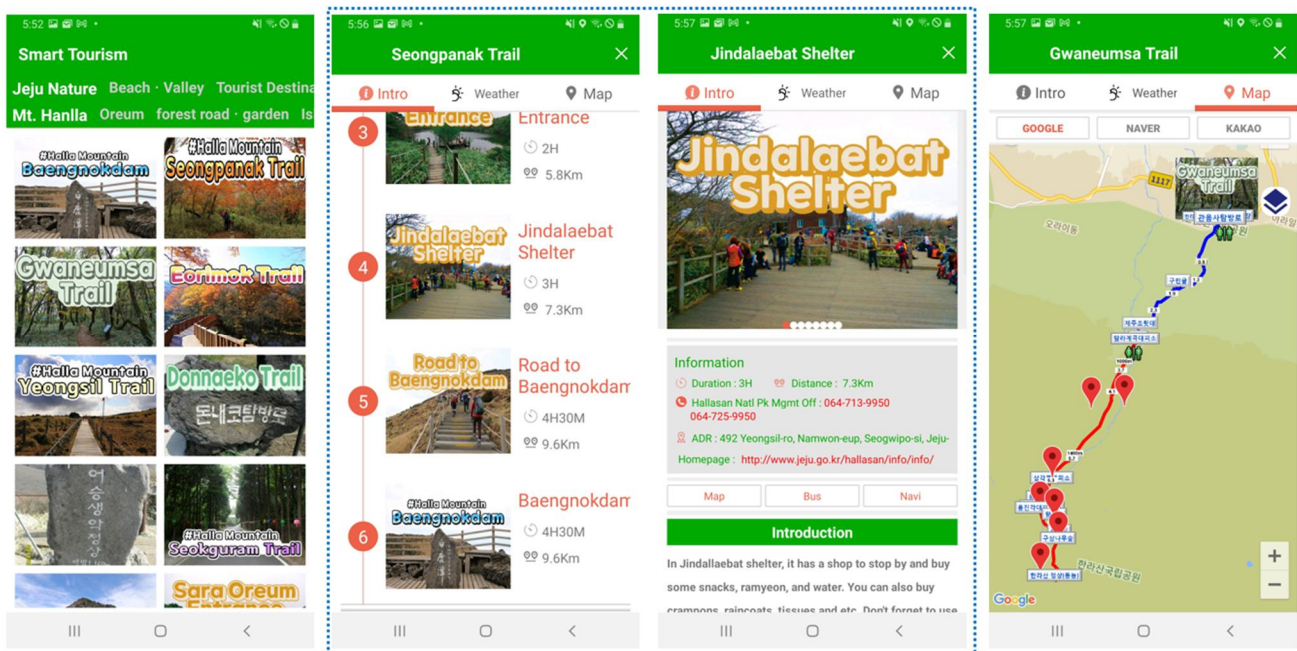


(a) main menu (b) Mt. Halla (c) Tourist information and Mt. Halla trails (d) KML data
Figure 1. Mt. Halla tourism information in the smart tourism Android app.

As shown in Fig. 1(a), the smart tourism app provides smart tourism services such as recommended travel products, my travel itinerary, and tourism information by linking with the smart tourism information system. We classify the tourism information of the smart tourism app into plane (zone), line, and dot (POI) type tourism information and implement the plane and line type tourism information to have a hierarchical structure. For example, Mt. Halla tourism information can be expressed by modeling the Mt. Halla trails such as Seongpanak Trail, Gwaneumsa Trail, Yeongsil Trail, Eorimok Trail, Donnaeko Trail, Eoseungsaengak Trail, and

Seokguram Trail as line type tourism information. We also model and express each trail as sub-line type tourism information. The Mt. Halla tourism information can be expressed by modeling as a hierarchical structure with line type tourism information (trails), and each line type tourism information has sub-line type tourism information. Figure 1(b) shows the Mt. Halla trails (line type tourism information). The Baengnokdam menu is a representative menu of Mt. Halla, and it shows information on Mt. Halla and tourism information on trails modeled as a line type. We can view tourist information for trails through the Baengnokdam menu as shown in Fig. 1(c). Through the Map menu of the Baengnokdam menu, you can view the trails of Mt. Halla on Google Maps, Naver Maps, and Kakao Maps. As Google Maps supports KML (Keyhole Markup Language) data [9], the trail route can be viewed on the map as shown in Figure 1(d). Tourists can easily select tourist destinations to travel by using tourism information having a hierarchical relationship and use this to easily create their own travel itinerary.

Figure 2(a) shows Seongpanak trail tourism information modeled as a line type of Mt. Halla. The Seongpanak Trail menu provides the distance and required time from the trail entrance to waypoints of the trail as shown in Fig. 2(b). Google Map of Donnaeko Trail Map menu provides information necessary for hiking as well as trail route as shown in Figure 2(c). In the trail route in Figure 2(c), blue indicates easy sections and red indicates difficult sections. Each waypoint of the trail is modeled and expressed as point (POI) type tourism information. Tourist information on each waypoint can be viewed through the waypoint menu of the trail as shown in Fig. 2(b). Tourist information on each waypoint provides basic information such as address, phone number, website, and URL, as well as weather information provided by the Korea Meteorological Administration [10], Google Maps [11], Naver Map [12], and KakaoMap [13] Views. The smart tourism app also provides a function in conjunction with Naver Navi for route guidance using buses and cars to tourist spots required for smart tour guides as shown in Fig 2(b). Tourists can travel easily and conveniently by using the bus and Navi functions according to their travel itinerary during their trip.



(a) Seongpanak Trail (b) Waypoint of the trail (c) KML data of the trail

Figure 2. Seongpanak trail tourism information modeled as a line type of Mt. Halla.

3. Smart Tourism React Native App for OSMU Tourism Information Service

We are enhancing the competitiveness of smart tourism services by providing tourism information and recommended travel products by integrating various content distribution channels such as smart tourism, Instagram, YouTube, and chatbots with our React Native app. In this paper, we develop React Native app to provide smart tourism app on iPhone.

3.1 OSMU Tourism Information Service

OSMU (One Source Multi-use) means developing single medium into multiple types of media. Smart Tour application can provide photos of tourist information on Instagram and drone images of tourist destinations on YouTube. The smart tourism React Native app has added a UI to the existing Android app so that it can provide photos of tourist information on Instagram and drone images of tourist destinations on YouTube. Through this UI, user can contact various tourism contents served by other platforms. Smart tourism services provide tourism information and recommended travel products by integrating various content distribution channels. Smart tourism React Native app provides Instagram photos and YouTube drone video services while providing tourism information to individual tourists. Figure 3 shows the smart tour service system that provides smart tour services with Android app and React Native app. Smart tourism React Native app provides tourism information of major tourist destinations from the smart tourism information system. Figure 3(a) shows the main menu providing YouTube, smart tourism app, and Instagram service. Figures 3(b) and (c) show the main menu of the smart tourism app and Mt. Halla tourism information of tourist attractions.

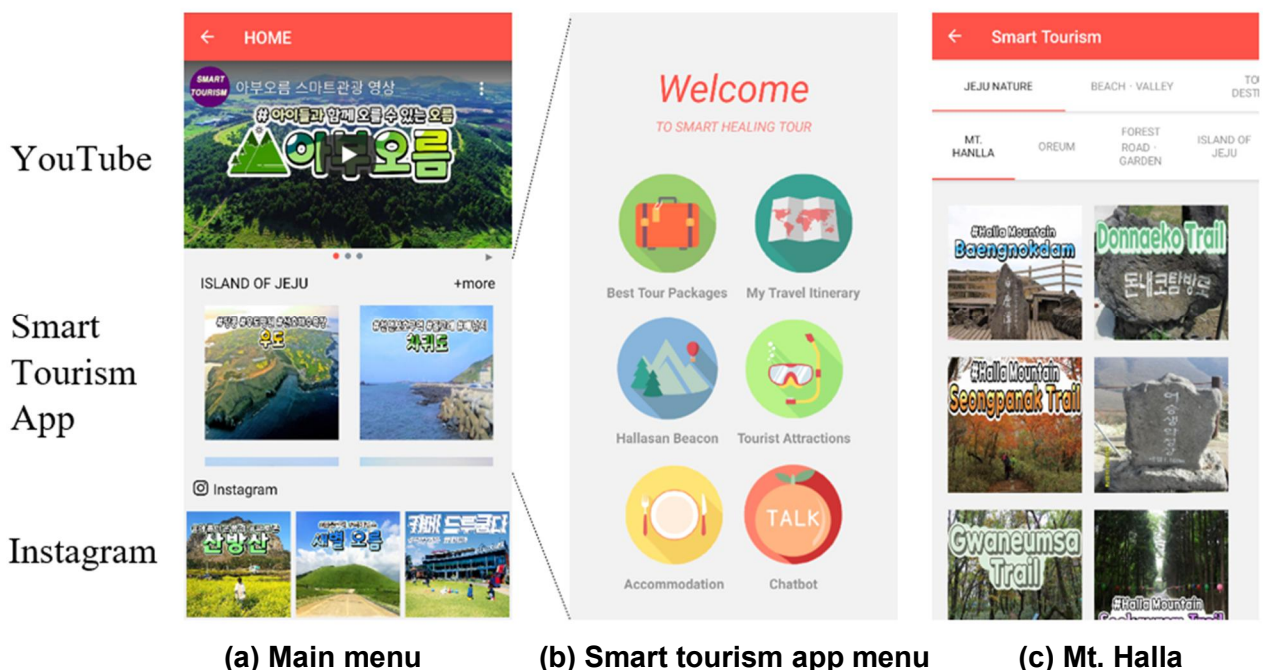


Figure 3. Smart tourism React Native app for OSMU tourism information service

3.2 React Native Operation Process

React Native is a framework for developing iOS and Android apps using the functions of Android and iOS native modules. In this paper, we develop a React Native app to run a smart tourism app on iOS. React-Native provides various components for UI implementation in app development. Developers use UI components to

compose the screen of the app, and the events that occur in the app are handled with functions implemented in JavaScript as shown in Fig. 4. META (Facebook) is adding components that accompany event processing through continuous updates. React Native uses threads to run applications. Threads include UI Thread, JavaScript Thread, Native Thread, and Render Thread. The UI Thread is called the main thread and is used to handle Android and iOS UI view rendering. On Android, it is used for 'measure//layout/draw'. The JavaScript (JS) Thread is used to handle all the logic of the application and runs your application's JavaScript code, API calls, touch events, and more. Native Modules Thread is used when an application accesses the platform API, and each module has the thread. The Render Thread is used to draw the UI only in Android 5.0 Lollipop. The Render Thread generates the OpenGL commands needed to draw the UI.

The working process of React-Native is as follows. The developer composes the screen of the app with UI components and implements the function using the JavaScript language in the React Native system. The React-Native system loads all the codes in the JavaScript Bundle when the application is running and runs the UI thread. While the UI Thread is running, the parts that need layout calculation are sent to the Shadow Thread to process the calculation, and then sent back to the UI thread to implement the screen. JavaScript Thread and Native communicate through Native Bridge. The Native Bridge is responsible for communicating with the native layer using JavaScript code. The iOS app calls the Objective-C API to render the iOS component, and the Android app calls the JAVA API to render the Android component. Since the called code is executed on the JavaScript Thread, not the UI Thread, it can be executed asynchronously without affecting the user experience. The biggest difference between React and React Native is the existence of a Native Bridge. React does not use the Native Bridge, but instead uses the Virtual DOM, a layer that exists between the developer's content and the actual screen rendering. React creates an in-memory data-structure cache, computes the resulting difference, and renders only the minimal changes needed. In React Native, the event generated by the application user is transmitted to the JavaScript Thread in the form of JSON object through the Native Bridge. In the JavaScript Thread, the function corresponding to the event is called and processed, and then passed back to the UI Thread through the Native Bridge, and the UI Thread composes a new screen and delivers it to the Native. After performing this process, the application user is provided with a new screen for the event.

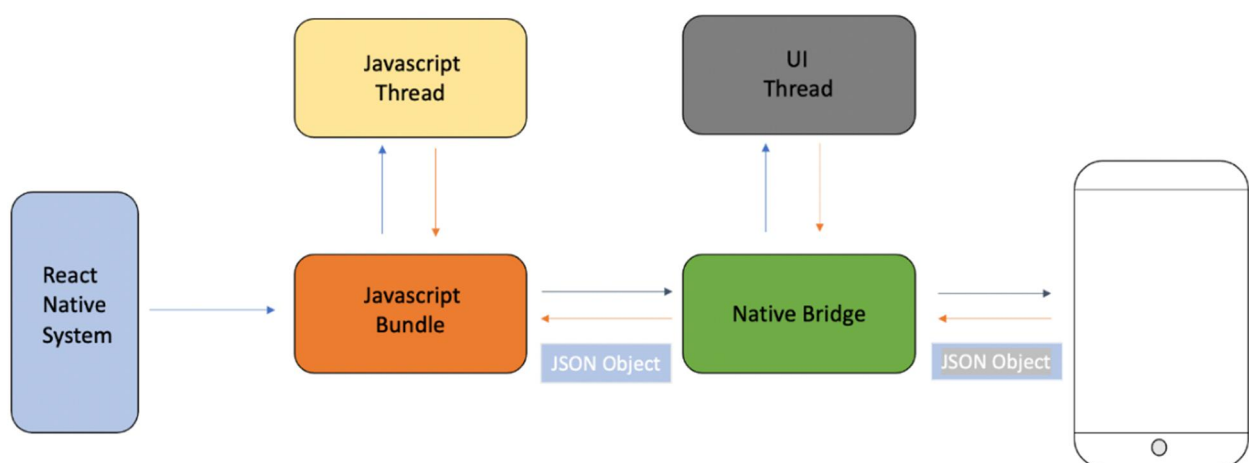


Figure 4. React Native operation process.

3.3 Interlocking with smart tourism information system through a GraphQL Server

The smart tourism service platform provides tourism information and recommended travel products to tourists preparing for a trip, and tourists can use this to create a personalized itinerary of travel products based on context awareness. The smart tourism information system consists of the Spring Framework-based web server and the SQL database that stores tourism information. The smart tourism information system has location and storytelling-based tourism information, accommodation, and restaurant data so that tourists can create personalized travel itineraries. The smart tourism information system provides recommended travel products made by travel experts and local experts. Tourists can create their own travel itinerary using data selected from recommended travel products and tourism information provided by the smart tourism information system. GraphQL is a Middleware Server that processes and transmits unnecessary data to users when receiving tourism information data from a smart tourism information system. GraphQL sends and receives data directly and from the smart tourism information system and REST API. Smart tourism Reactive Native app can receive tourism information from smart tourism information system in two ways as shown in Fig. 5. The first is to receive tourism information from a smart tourism information system like an Android app, and the second is to link it through a GraphQL server. Although both methods are implemented in this paper, the smart tourism app service is currently being performed by directly connecting to the smart tourism information system.

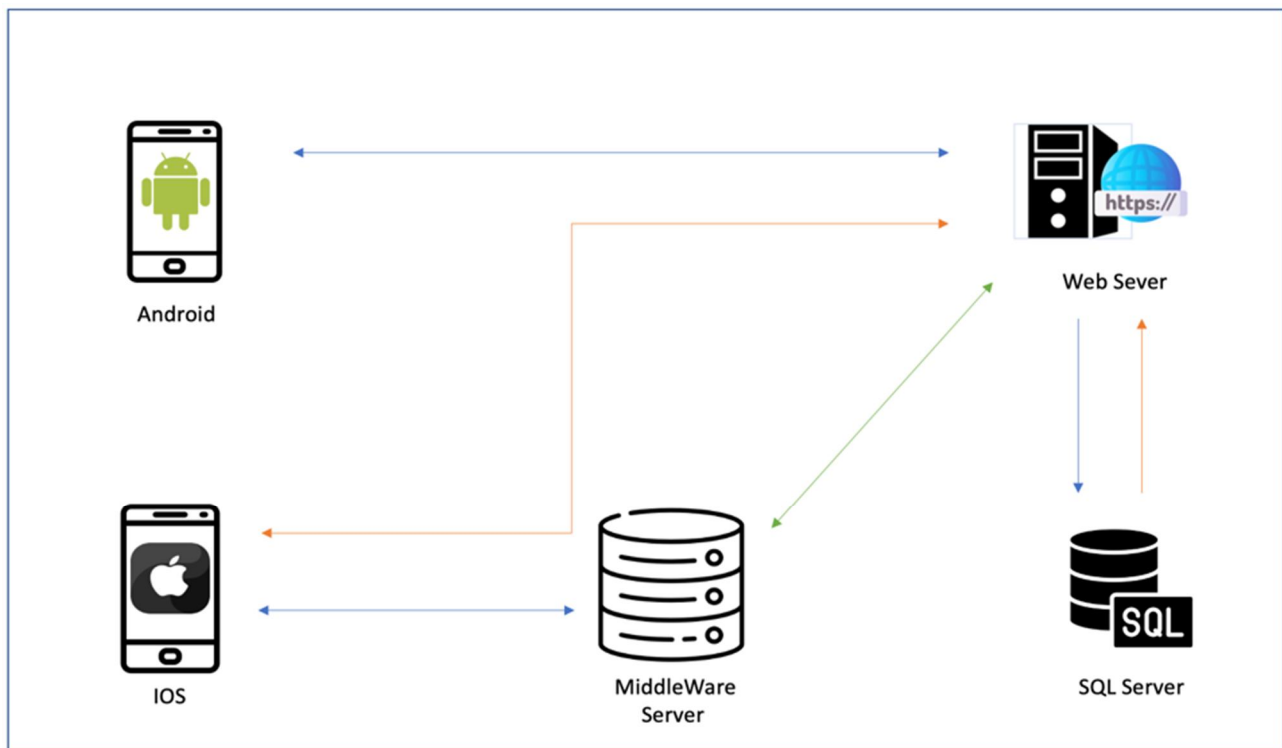


Figure 5. React Native app for providing smart tourism services.

4. Conclusions and Further Works

We have been developed a smart tourism service that provides recommended travel products and tourist information to individual tourists to create my own travel itinerary and provide tour guide services to tourists according to their own travel itinerary. In this paper, to efficiently provide tourism information to tourists, we propose a method to model tourism information of complex tourist destinations in a plane (zone) and line type

and to express tourism information in the hierarchical relationships. In the smart tourism app, tourism information is modeled as a plane and line type has the hierarchical relationships and expressed as KML data on Google Maps in the Map menu to enhance the user's convenience. Tourists can easily use tourism information on the map because KML data includes tourist information such as routes, POIs, and photos.

We also develop iOS React Native App for smart tourism service in conjunction with the smart tourism information system. Smart Tourism React Native App provides tourism information services through Instagram and YouTube as well as smart tourism apps. We provide OSMU tourism information service as a smart tourism React Native app to activate the smart tourism service. In the future, we plan to promote advanced development of restaurant recommendation and management functions to advance my travel schedule creation function.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.2021R1A2C1093283).

References

- [1] Gretzel, U., Sigala, M., Xiang, Z. and Koo, C, "Smart Tourism: Foundations and Developments", *Electronic Market*, pp.179-188, 2015.
DOI:10.1007/s12525-015-0196-8
- [2] Sameera A Abdul-Kader and JC Woods. "Survey on chatbot design techniques in speech conversation systems", *International Journal of Advanced Computer Science and Applications*, 6(7), pp.72-80, 2015.
DOI: 10.14569/IJACSA.2015.060712
- [3] F.Clarizia, F. Colace, M. Lombardi, F. Pascale, "A Context Aware Recommender System for Digital Storytelling", *IEEE 32nd International Conference on Advanced Information Networking and Applications*, pp.542-549, 2018.
DOI: 10.1109/AINA.2018.00085
- [4] JeongWoo Jwa, "Development of Personalized Travel Products for Smart Tour Guidance Services", *International Journal of Engineering & Technology*, 7 (3.33) 58-61, 2018.
DOI: 10.14419/ijet.v7i3.33.18524
- [5] Jeong-Woo Jwa, "Pedestrian Network Models for Mobile Smart Tour Guide Services," *International Journal of Internet, Broadcasting and Communication*, vol.8, no.1, pp.73-78. 2016.
DOI: 10.7236/IJIBC.2016.8.1.27
- [6] KiBeom Kang, JeongWoo Jwa, SangDon Earl Park, "Smart Audio Tour Guide System using TTS", *International Journal of Applied Engineering Research*, pp.9846-9852, 2017.
https://www.ripublication.com/ijaer17/ijaerv12n20_81.pdf
- [7] Android, <https://www.android.com/>
- [8] React, <https://reactjs.org/>, <https://reactnative.dev/>
- [9] Google KML, https://developers.google.com/kml/documentation/kml_tut
- [10] Korea Meteorological Administration, <https://www.kma.go.kr/kma/>
- [11] Google Maps for Android, <https://developers.google.com/maps/documentation/android-sdk/start?hl=ko>
- [12] Naver Map Android SDK, <https://navermaps.github.io/android-map-sdk/guide-ko/1.html>
- [13] KakaoMap Android SDK, <https://apis.map.kakao.com/android/guide/>