An Approach to Developing Internet of Things (IoT)－Based Services for Smart Museums

Rosen Ivanov[0000-0002-4768-6500]

Technical University - Gabrovo, 4 “H. Dimitar”, Gabrovo, Bulgaria
rs.soft.bg@gmail.com

Abstract. The aim of each museum is to present its exhibits in a way that enhances the visitor experience, satisfaction and knowledge. This objective is difficult to achieve, especially in museums with a very large number of exhibits and visitors. One possible solution is to use the modern Internet and mobile technologies in order to implement a service for personalized information delivery. The content that is delivered to visitors should be consistent with their preferences and free time. The paper presents an Internet of Things (IoT)－based solution for delivering personalized content to museum visitors, which includes: (1) Mobile app for visitors and curators; (2) Sensor network; and (3) Edge Computing Gateway. The main advantages of this solution are the following: platform independent service thanks to its full-stack JavaScript architecture; the service allows visitors to use it in multiple museums, not just in a particular museum; the service does not require permanent Internet access to operate; ability to deliver personalized content during the movement of visitors along the exhibits; relatively low price of the service thanks to the used hardware and open source software. Preliminary experiments in simulated and real environments show that proposed solution can be successfully used to create personalized content delivery services in smart museums.

Keywords: Museum Guide Systems, Visitors Profiling, Digital Heritage, IoT.

1 Introduction

The role of museum curators is very important because, based on their experience and knowledge they select which exhibits how to be presented to visitors (Ardito, 2018), (Antoniou, 2016). Their professional assessment can be improved if they have statistical information such as number of visitors who have viewed each exhibit and the time spent by them for this consideration. Partially, this information can be obtained by completing the questionnaire before visitors leave the museum. Practice shows that most of the visitors do not fill out such questionnaire or do not answer all questions. With the development of mobile and Internet technologies (Lee, 2015), this information can be easily obtained automatically (Chianese, 2014). If visitor location is accurately calculated, it is possible to deliver personalized content to each visitor for the exhibit that is in close proximity to it (Kosmopoulos, 2018).

Digital Presentation and Preservation of Cultural and Scientific Heritage. Conference Proceedings. Vol. 9, Sofia, Bulgaria: Institute of Mathematics and Informatics – BAS, 2019. ISSN: 1314-4006, eISSN: 2535-0366

A system for delivering personalized content should have the following main parts: (1) Visitor's localization module; and (2) Visitor's profiling module. Visitors should be able to locate both indoors and outdoors. Localization outdoors is not a problem because Global Positioning System (GPS) positioning of visitors is possible. The micro localization of visitors within the museum's rooms is a real problem. The most commonly used localization methods in this case are the following: proximity localization; lateration-based localization; fingerprinting-based localization; and dead-reckoning localization. Several technologies may be applied to indoor localization (Hossain, 2015). Parts of them require some changes in infrastructure and sensors to be deployed in the museum environment. Ultrasound-based localization is sensitive to temperature variance and multipath signals. Computer vision based localization algorithms are sensitive to the camera characteristics and lighting conditions. UWB-based systems have a higher deployment cost than other Radio Frequency (RF) techniques due to the need for high precise inter node time synchronization. Comparing with other signal sources RF-based localization is preferable because it is cost effective. To guarantee required localization accuracy many systems are hybrid – they combine several localization techniques into one more precise.

The main problem for personalized content delivery systems is the so-called cold start (Kosmopoulos, 2018). This is the moment when the app is launched, but a visitor's profile is missing, or the profile doesn't contain enough information to suggest personalized content. Profiling is a process that retrieves specific information for each visitor, which makes it possible to decide what content to be delivered (Antoniou, 2016). Most often this information is obtained implicitly by filling in a Web form or explicitly by an analysis of the time spent to viewing each exhibit. For example, the official mobile app for exploring the Louvre (Lourve, 2019) uses a form through which the visitor informs the system about: the time it can spend; preferred exhibits; and does the visitor have reduced mobility. Since the visitor may not enter this information, a personalized content delivery system should be able to build the profile of each visitor dynamically. This would be possible if the system is context-aware (Perera, 2014). Such systems can sense their physical environment, and adapt their functionality accordingly. In the specific case, to the context may be assigned: the location of the visitor; does the visitor moves or not; which exhibits are close to the visitor; museum opening hours; the visitor's affiliation to a particular interest group; the type of exhibits viewed by the visitor. The opening hours of the museum are necessary to estimate the number of exhibits to be offered for consideration. The activity of the visitor is analyzed to detect when she/he has stopped near an exhibit.

The description of the exhibits can be realized through various media, such as text, audio, images, video, virtual reality data and augmented reality data. It is desirable this description to be in accordance with one of the standards for the description of exhibits and collections. Each standard describes the exhibits through a specific number of metadata, for example: name; type; material; technique; owner; physical parameters; history; location; text description, images; videos; etc. The most commonly used standards for the description of exhibits are as follows:

- SPECTRUM. This standard details how to manage collections and what to do with artefacts at each stage of their lifecycle in a collection. Spectrum is for

museums of any size and any collection type. In last version (5.0) there are nine primary information groups.

- Dublin Core. This standard can be used to describe digital resources (video, images, web pages, etc.), as well as physical resources such as books, CDs and DVDs. The original set of 15 metadata terms, known as the Dublin Core Metadata Element Set (DCMES) are used to describe objects.
- Object ID. Object Identification (Object ID) is an international standard used for describing cultural objects, facilitating the identification of collections of archaeological, cultural and artistic objects in case of loss or theft. The Object ID standard defines nine categories of information to describe an object.

The main objective of this paper is the description of the design and preliminary tests of a service for personalized content delivery in smart museums. The architecture of the proposed service allows solving of problems analyzed above: (1) Precise visitor's localization using sensor network of Bluetooth Low Energy (BLE) beacons; (2) Supports implicit and explicit building of visitors' profiles; (3) The service is not limited to working in a particular museum; and (4) The service does not require permanent Internet access to operate.

The remainder of this paper is organized as follows. In Section 2, we describe the overall service architecture. Section 3 discuses validation of the proposed service and preliminary test, and finally Section 4 concludes the paper and presents some ideas for future work.

2 The Proposed Service Architecture

The IoT implementation of the service is used in order to monitor in real time the interaction of each visitor with the museum's exhibits. For this purpose the necessary sensor network is built within the museum. It includes BLE beacons, Near-field Communication (NFC) tags and Quick Response (QR) tags. The use of NFC and QR tags for exhibit identification is optional in order to reduce the cost of the service. When using BLE beacons, an automatic content delivery is implemented. In this case the service can be accessible to people with visual and motor disabilities.

Fig. 1 shows the overall structure of the proposed service architecture. It is composed, as described below, of four parts: (1) Edge Computing Gateway; (2) Master database in the cloud; (3) Mobile app for visitors and curators; and (4) Sensor network.

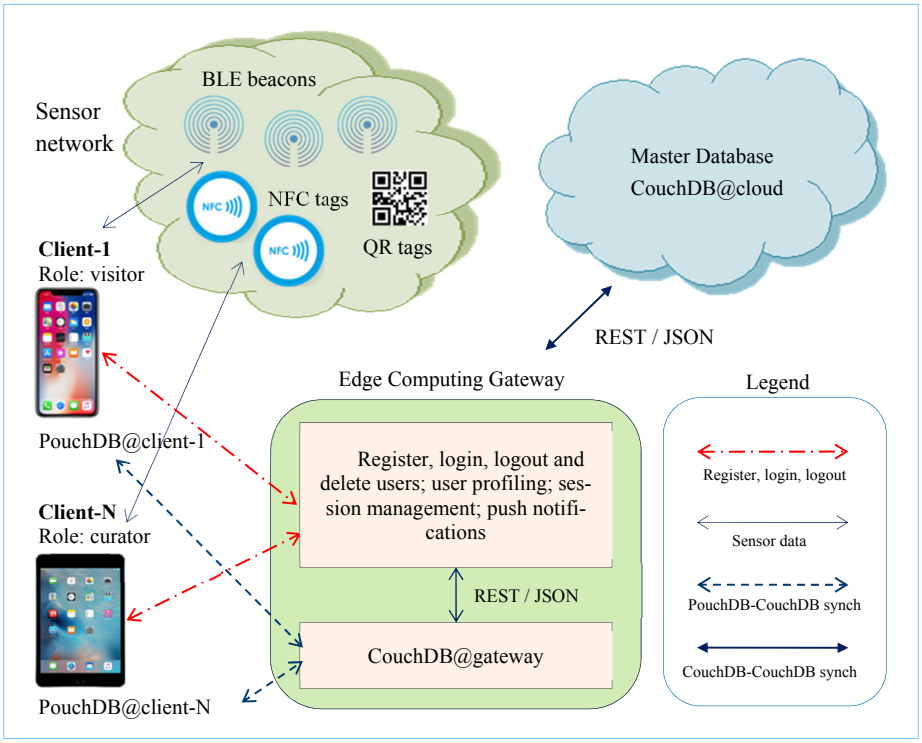

Fig. 1. General overview of the service architecture

2.1 Edge Computing Gateway

We use decentralized content processing at the edge of the museum's local network. Edge computing pushes data and computing power away from centralized point (cloud) to locations closer to the clients of the service. In our case edge computing gateway does not need connection with any cloud, but we use database backup at cloud to ensure that museum's data is highly available and to enable database management outside the local network of the museum. The services that are running on the edge computing gateway are WebSocket server and Database server. The CouchDB is used for database server. This server is configured to disable anonymous access to databases and to create a separate database automatically for each new user. The WebSocket server is implemented using Node.js. This server supports: (1) Database users' management (registration, login, logout, delete); (2) Sessions management; (3) Visitors profiling; and (4) Push notifications. All clients communicate with WebSocket server using REST requests. In order to reduce the cost of the service, the edge computing gateway is implemented by a single-board computer.

2.2 Master Database in the Cloud

As mentioned before, the service is fully functional and without a connection to the cloud. However, if a particular museum wishes to reliably store data for its visitors and exhibits, it is possible to backup the databases into the cloud. The service allows its clients to receive information about the exhibits of many museums using one mobile application. For this purpose, the cloud database stores information for all visitors. This allows sharing of visitors data between museum gateways and more quickly and accurately build their profiles. The CouchDB is used for the database server in the cloud. Synchronization of information between the gateway and cloud databases is implemented using filtered replications.

2.3 Mobile App for Visitors and Curators

Each client of the service gets access to the service functionality using a mobile application. This app can be downloaded automatically by scanning a QR code or touching an NFC tag (with a NFC-enabled mobile device) that should be placed at the entrance to the museum. Mobile devices with operating system Android or iOS are supported. According to StatCounter (GlobalStats, 2019), mobile operating system market share worldwide for May 2018 - May 2019 for these operating systems is 98.01%. As the service delivers personalized content, it must be able to identify each of its clients. Some of the existing applications use the MAC address of the mobile device as the unique identifier (Chianese, 2014), (Pierdicca, 2019). This solution is only possible for Android mobile devices. For iOS devices, this information is inaccessible – they return a UUID code that is temporarily unique. This requires each visitor to register in order to use the service. The registration information is a valid e-mail address and password. The user management service at edge computing gateway checks in the cloud database whether the client has already registered when visiting another museums. If the visitor was registered, information about its profiling is obtained. Registration is limited to creating a new user on the CouchDB@gateway server. If the registration is successful, WebSocket server returns token to access gateway services. After registration, the visitor is asked to enter information related to his/her profile (implicit profiling). This information may not be entered or partially entered. In this case, the service delivers personalized content when sufficient information is received during the visitor's movement in the museum. A local database is used to enable the mobile app to operate when there is no network connectivity with the gateway. Local database is created by running the PouchDB server. The information in the local databases is automatically updated from the gateway databases by using live replications.

2.4 Sensor Network

It is suggested that the localization of visitors and objects to be implemented through a sensor network containing BLE beacons. They are wireless sensors that continuously advertising their location using a BLE radio transmitters (Sornalatha, 2017). Mobile

device, which supports BLE, can receive advertising data packets and determine visitor's proximity to the beacon analyzing Received Signal Strength Indicator (RSSI) value in the advertising packed. Standard Beacon APIs return proximity as immediate, near, and far. The proposed service uses its own beacons API and algorithm to process beacons' data (Ivanov, 2017). Thus it is possible to obtain the distance to each beacon in centimeters, as well as the visitor position with an error of less than 1 m. Beacons differ depending on what is identified with them, for example: a building, a room, a group of exhibits, and an exhibit. Depending on this, a specific beacon calibration is performed (transmitter power, RSSI at a distance of 1 meter from the beacon, and advertising interval). Beacons enable delivered content to be triggered automatically while visitors move through museum.

2.5 Database Design

Fig. 2 shows the databases that the service uses, and how synchronization of information between local, gateway, and cloud databases is realized. The databases that the service uses are the following:

- Database "metadata". This database contains metadata about the museum: ID code; UUID of the beacons; list of the types of exhibits; list of interest groups; list of types of disabilities with which the infrastructure of the museum is compliant; and museum opening and closing time.
- Database "museum-ID-visitor-ID" contains information for a particular user. Each visitor has its own database that only that visitor can read and write. This database contains information used for user profiling.
- Database "sensors" contains information for each sensor in a particular document.
- Database "object-mapping". Since it is possible the objects to be identified in three ways (beacon ID, NFC tag ID, and QR code), it is necessary to describe the relation between the sensor ID and corresponding object ID.
- Database "objects" describes each object with a separate document.
- Database "floorplan". This database contains floor plans for each building of museum in SVG format.
- Database "evacuation". This database contains information that is used when an evacuation plan is activated.

Museum exhibits are described by a set of metadata, the major part of which is defined in the Object ID and SPECTRUM standards: (1) The name of the exhibits; (2) A list of the keywords describing the exhibit; (3) History (where exhibit was found, when it was found, cultural period, etc.); (4) Current location (country, city, museum, etc.); (5) General information about the exhibits (type, technique used, is the exhibit a cultural value, is this an original exhibit, etc.); (6) Physical description (material, physical dimensions and weight); (7) Bibliography (list of references with information about the exhibit); (8) Documents (photos, videos and other file formats); (9) Related exhibits (list of IDs of similar exhibits); (10) Comments (Web addresses of social networks where are comments about the exhibit).

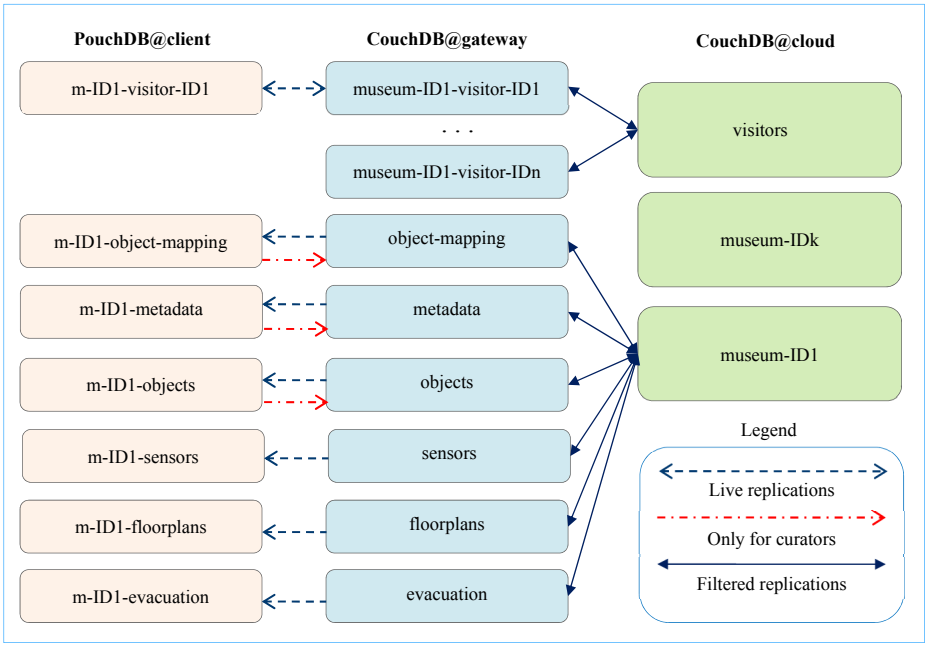

Fig. 2. Database design

3 Service Validation

The proposed service architecture is validated in simulated museum environment and real environment (university department). We have built a prototype of the proposed architecture which includes: (1) Mobile app; (2) Edge computing gateway; (3) Sensor network; and (4) Master database in the cloud.

3.1 Edge Computing Gateway

Edge Computing Gateway is implemented through single-board computer Raspberry Pi 3 (Raspbian Stretch OS), Node.js and CouchDB 2.2 server. The CouchDB server configuration includes: (1) Create accounts for an administrator and curators; (2) Enable SSL support; (3) Enable Cross-Origin Resource Sharing (CORS); (4) Enable couch-database-per-user scenario; (5) Prohibit anonymous access to databases; and (6) Configure the replications with CouchDB@cloud. When replication source is cloud database and target is gateway database only documents belonging to target database should be replicated. We use filter functions in design documents to realize filtered replications.

To test the service we built accounts for 1000 visitors (role=visitor) and 5 curators (role=curator) using simulated registration requests to the WebSocket server. For each new visitor in system database "_users", the CouchDB server automatically creates a database called "museum-ID1-visitor-ID2," where ID1 is the ID of the museum, and

ID2 is a hex encoded visitor name. The information required for the profiling of each visitor is stored in a database with name that matches the name of the database created for it. Next, we add 1000 documents into database "objects". Each document describes an object, for example museum, building, level, room or exhibit. Each exhibit related object is filled with computer generated data. The size of each document from this database is up to 2.8 MB, depending on the files that are attached to each of them. Finally, we add into database “sensors” 1020 documents, which describe 1000 virtual sensors (900 BLE, 50 NFC tags, and 50 QR codes) and 20 real BLE beacons.

3.2 Mobile App

A hybrid mobile application (HTML, CSS and JavaScript) has been developed using the PhoneGap framework (CLI 8.0.0). The user interface is built using a Framework7. The app can be installed on Android and iOS mobile devices. The JavaScript libraries pouchdb-7.0.0.min.js and pouchdb.find.js are used to implement the functionality of the local PouchDB server. This server is started with a SQLite adapter because it allows unlimited data storage compared to adapters IndexedDB and WebSQL. A snapshot of the mobile app running is shown in Fig. 3.

3.3 Sensors

The price for the implementation of the service depends mainly on the cost to building the sensor network. In order to obtain the functionalities of the service, it is necessary to associate a beacon with each exhibit. Some of the beacons are priced too high and others can only be used with the manufacturer's API. For this prototype, BLE beacons YJ-15XXX by Holyiot are used. They use Nordic chipset nRF51822, have small size (Ø26 mm) and can be easily programmed. At a price below $10, they are perfectly suited for implementing the service.

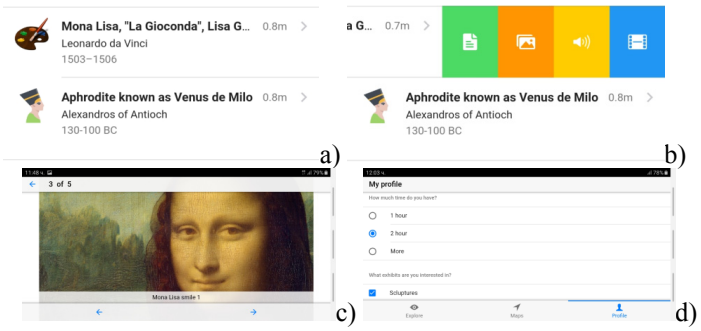

Fig. 3. Screenshots (real beacons, simulated data): a) Notification for nearby exhibit(s); b) Select preferred media type; c) Browse images; and d) Build user profile.

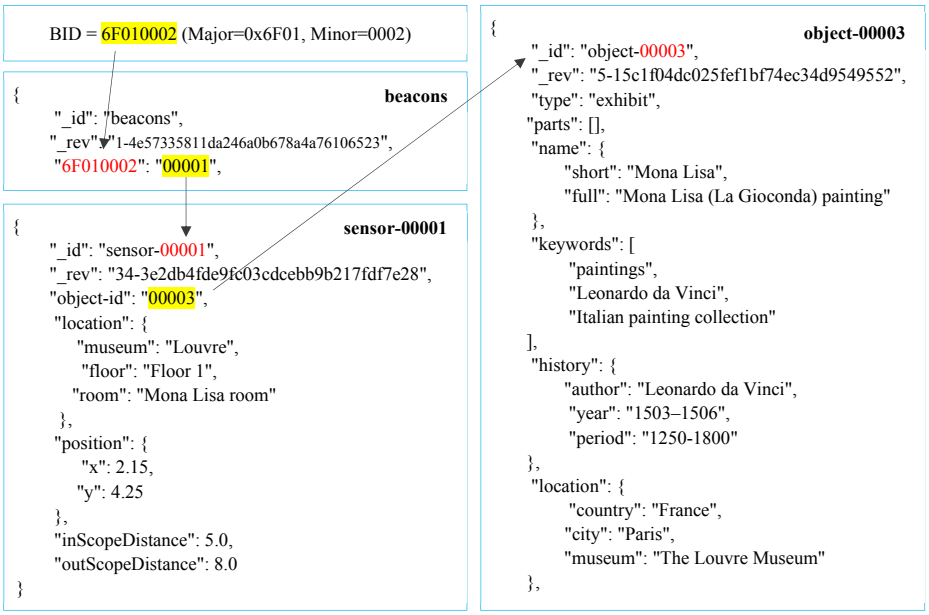

Fig. 4. Sequence of receiving the name of the document that describes an exhibit by a given beacon identification code (BID)

Each beacon is described by a JavaScript object. The most important fields of this object are following: RSSI value at distance 1 meter from the beacon (RSSI0); distance to the beacon under which the mobile app shows notification about it (*inScopeDistance*); distance to beacon after which mobile app stops processing data from beacon (*outScopeDistance*); text description; and an image. This information is obtained from "sensors" database. The object describing beacons provides the following methods: (1) RSSI data filtering using an Adaptive Kalman Filter (AKF); (2) Calculate the distance to the beacon; and (3) Checking whether the visitor is in predefined proximity to the beacon. All objects describing the beacons that are near to visitor are stored in the beacons pool (an associative map object). It allows adding a new beacon object, deleting a beacon object; as well as checking whether a particular beacon object is already in the beacons pool.

We use a callback function to process data from beacons. It is called every time when an advertising data packet from a beacon with a UUID associated with the museum is received. The values of Major, Minor and RSSI fields are extracted. A beacon identifier (BID) is obtained from values of the Major and Minor fields. Next, program code checks if the beacons pool contains a beacon object with that BID. If the object is missing, it is created and initialized. From document "beacons" (database "object-mapping"), the document ID that describes a sensor with this BID is obtained (see Fig. 4). From this document (sensor-00001) the object identifier with which the beacon is associated (object-id field) is obtained (object-00003). The RSSI data is then filtered us-

ing an AKF. Finally, the distances (di) between the visitor and i-th beacon in the beacons pool are calculated. If the visitor approaches a beacon and if *di<inScopeDistance*, the app displays a notification for that. If the beacon is used for localization, the information obtained is displayed in the location bar of the app. If the beacon is used to describe an exhibit the application shows the notification with the image and the text associated with that exhibit. If the visitor does not respond to the notification within 4 seconds, all personalized information about the exhibit is displayed on the screen. If the visitor moves away from the beacon and *di>outScopeDistance*, the beacon object is deleted from the beacons pool and mobile app hides the information about it.

3.4 Discussion

Experiments conducted with virtually generated users, objects and sensors show that the proposed architecture is working and can be used to develop personalized content delivery services. The selected hardware for the Edge Computing Gateway implementation has enough computational power to serve competitive queries from 1000 visitors (delay of less than 1.6 seconds). When using a 32 GB MicroSD card, over 5000 objects can be described if an average of 5 MB is allocated to each. The beacon data processing algorithm works reliably even when simulating advertising data from 1000 beacons. In practice, the mobile app should be able to process data from no more than 50 beacons at the same time. Experiments in a real environment show that using our algorithm for RSSI data filtering and in-scope and out-scope intervals greatly reduces the impact of signal interference, even when there are groups of moving people in the rooms. The choice to use NoSQL document type databases is justified because they are very appropriate when describing data without a strict scheme and allow omitting certain fields and thus reducing the size of the documents. Document-type databases are appropriate when information in documents rarely changes. In a specific development, only the visitors' database documents are changed to profiling them. This is the reason why a separate database for each visitor is used.

4 Conclusions

The paper presents an IoT-based architecture of the service for personalized content delivery in smart museums. The prototype of the proposed architecture was build and we explain the configuration and implementation details for each part of the prototype. The proposed architecture is validated in simulated museum environment and a real environment (university department). In the future, the service will be tested in a real museum environment. In this case, the following more important issues should be addressed: (1) How to persuade visitors to install the mobile app on their phones and use the app during the whole time of their visit; (2) Reduce authentication time using OAuth2.0 service; (3) Develop the necessary application for beacons calibration; and (4) For museums with very large number of exhibits, the number of Edge Gateways should be increased. For this purpose, a module for redirecting user requests to Edge Gateways should be developed.

References

Antoniou, A. K.-T. (2016). Capturing the visitor profile for a personalized mobile museum experience: An indirect approach. *24th ACM Conference on User Modeling, Adaptation and Personalisation. 1618* (pp. 1-10). Halifax, Canada: ACM.

Ardito, C. B. (2018). From smart objects to smart experiences: An end-user development approach. *International Journal of Human Computer Studies, 114*, 51-68.

Chianese, A. (2014). Designing a smart museum: When cultural heritage joins IoT. *8th International Conference on Next Generation Mobile Applications, Services and Technologies* (pp. 300–306). IEEE.

GlobalStats. (2019). *Mobile Operating System Market Share Worldwide*. Retrieved from Mobile Operating System Market Share Worldwide: http://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-201805-201905

Hossain, A. M. (2015). A survey of calibration-free indoor positioning systems. *Computer Communications, 66*, 1-13.

Ivanov, R. (2017). An Algorithm for Micro-localization in Large Public Buildings. *18th International Conference on Computer Systems and Technologies* (pp. 119-126). ACM.

Kosmopoulos, D. (2018). A survey on developing personalized content services in museums. *Pervasive and Mobile Computing, 47*, 54-77.

Lee, I. (2015). The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons, 58(4)*, 431-440.

Lourve. (2019). *The Louvre App. The Louvre museum*. Retrieved from The Louvre App. The Louvre museum: https://www.louvre.fr/en/new-app

Perera, C. Z. (2014). Context aware computing for the internet of things: A survey. *IEEE Communications Surveys and Tutorials, 16(1)*, 414-454.

Pierdicca, R. M.-P. (2019). IoT and Engagement in the Ubiquitous Museum. *Sensors*, 1-21.

Sornalatha, K. (2017). IoT based smart museum using Bluetooth Low Energy. *3rd International Conference on Advances in Electrical and Electronics, Information, Communication and Bio-Informatics* (pp. 520-523). IEEE.

Received: June 12, 2019
Reviewed: June 30, 2019
Finally Accepted: July 15, 2019