Tangible and Personalized Smart Museum Application

Rosen Ivanov[0000-0002-4768-6500], VictoriaVelkova[0000-0002-2951-2005]

Technical University - Gabrovo, 4, H. Dimitar Str., Gabrovo, 5300, Bulgaria
rs.soft.bg@gmail.com, v.velkova@tugab.bg

Abstract. This paper presents the architecture of a Web app designed to deliver personalized content to museum visitors both indoors and outdoors. The app uses a tangible type of interface to obtain information about the museum's exhibits. For this purpose, each exhibit can be associated with an NFC tag, a Bluetooth Low Energy (BLE) beacon or a geofence in open-air museums. The service automatically profiles its users based on statistics about the preferred type of exhibits, media formats and time spent viewing each exhibit. Visitors can get the information they want about an exhibit even when the app is not running. Push notifications are used for this purpose. The necessary experiments have been conducted to prove the applicability of the service for real-time delivery of personalized content.

Keywords: Smart Museums, Tangible Interfaces, User Segmentation, User Profiling, Push Notifications.

1 Introduction

The main aim of smart museums is to transform static exhibitions into interactive ones by using modern technologies in the field of communication and Internet technologies ElDamshiry, & ElFouly,2022). The information about the available exhibits should be presented in such a way that every visitor is satisfied. This is possible if personalized content is delivered to visitors according to their interests and preferences. Visitor personalization is more successful the more accurate the visitor profiling. Visitor profiling can be implicit, explicit, or combined (Antoniou et al., 2016). Implicit profiling is mainly implemented by completing surveys. Explicit profiling is realized by analyzing visitor activity within the museum and time the visitor stays near the exhibits.

In addition to an enjoyable experience and satisfaction, most museums aim for their visitors to gain new knowledge (Vassilakis, 2017). For this to be possible, visitor engagement with the information presented about exhibits needs to be increased (Fan, 2022). To maintain high visitor engagement, it is very important how the user interface is designed and implemented and how information about the exhibits is visualized. Very good results are obtained when using tangible user interfaces (TUI). Tangible user interfaces are a form of human-computer interaction based on physical interaction with objects in the environment. The goal of this type of interfaces is to simulate the familiar way for humans to interact with objects, e.g., touch. The main advantage of tangible

Digital Presentation and Preservation of Cultural and Scientific Heritage. Conference Proceedings. Vol. 13, Sofia, Bulgaria: Institute of Mathematics and Informatics – BAS, 2023. ISSN: 1314-4006, eISSN: 2535-0366

interfaces is the reduction of cognitive load. Users do not have to be trained for long to use TUI successfully (Petrelli & O'Brien, 2018).

At the current stage, two technologies are mainly used to create a tangible type of user interface for mobile devices: 1) Near Field Communications (NFC) tags and Bluetooth Low Energy (BLE) beacons (Dragović et al., 2018). After associating an object with an NFC tag, it is easy to recognize the object by reading its identification code. With NFC technology, the distance at which communication is possible between the phone's NFC reader and the tag is less than 4cm. If you need to detect objects from a greater distance (several meters) you can use BLE beacons. The beacons broadcast a data packet during a set time interval. The structure of the packet depends on the beacon protocol. The packets may contain a variety of data, the most important of which are the beacon identifier (UUID), Minor and Major values, and transmitter power. When associating an object with a BLE beacon, we can recognize when we are approaching the object by analyzing the signal strength received from the beacon.

When the objects to be recognized are of large size, for example a building or a monument, beacons are not suitable. In this case, so-called geofences can be used (Shoji, 2021). The term "geofence" defines a virtual perimeter for a real geographical region. This region is most often described as a polygon or circle. The use of geofences to create location-based services for users is called geofencing.

A proven way to increase mobile user engagement is the use of push notifications. Push notifications allow services to send messages to their users even when the mobile app is not running. They can be used to deliver information and updates as well as encourage users to engage with the information delivered by the service. The engagement rate of push notifications is many times greater than the engagement rate of a normal user interface.

The main aim of this paper is to describe the design and preliminary tests of an application with tangible interface that delivers personalized content for museum exhibits. The app segments visitors according to their location and a profile that is dynamically built. A combination of requests/responses and push notifications are used to deliver content to visitors. The visitor receives structured content related to an exhibit, in volume and media formats according to their profile.

The remainder of this paper is organized as follows. In Section 2, we describe the overall app architecture. Section 3 discuses validation of the proposed service, and finally Section 4 concludes the paper.

2 The Proposed App Architecture

The application has a client-server architecture. The business logic is implemented as a Node.js Web app and hosted in Microsoft's Azure cloud. For this purpose, the Azure App Service is used, which provides a scalable and self-patching Web hosting service using the Linux OS. The clients of the service are museum visitors. The service is designed to deliver personalized content to both indoor and outdoor museum visitors. Fig. 1 shows the architecture of the proposed app.

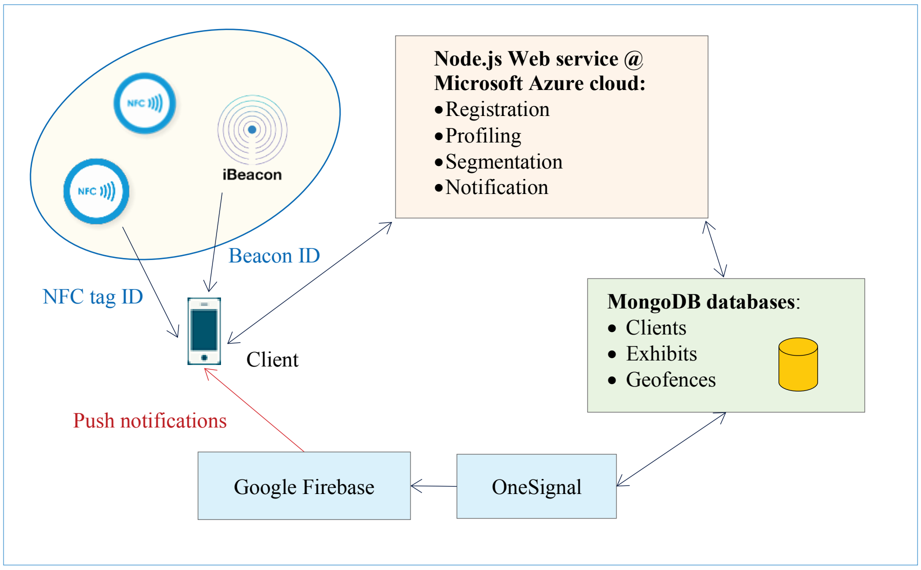

Fig. 1. General overview of the app architecture

2.1 Mobile App

Clients of the service must have a mobile device with the Android OS version 9+ app installed. It is foreseen that this app can be installed automatically after scanning an NFC tag placed in a suitable location, e.g., at the museum entrances. The main tasks of the mobile application are:

- Scan for available BLE beacons and NFC tags.
- Obtain visitor GPS positions at open-air museums.
- Recognition of visitor activity.
- Communication with Web app @ Azure cloud.

To implement the required functionality, the mobile application must have access to the following sensors and modules embedded in the mobile device:

- Accelerometer, compass, and gyroscope (optional).
- GPS receiver.
- NFC reader.
- Bluetooth Low Energy (BLE).

The accelerometer, compass and gyroscope are used to detect the activity of the clients. This algorithm classifies the activity by assigning each client to one of the following classes: still, walking, and running. The activity type recognition is used by the service which gets the GPS position of the clients. This service, to extend the battery life of the mobile device, reads and sends to the Web app the client's position only if a significant movement of the client relative to its previous position is detected. To make this possible, we use foreground service in Android version 9 to 11. Since in Android version 12+ it is forbidden to start foreground services from the background another approach must be used. Google recommends using WorkManager, but in this case the programmer has no control of the sensor scan time. For this reason, a service of class JobIntentService is used, which is periodically activated by onUpdate method of class AppWidgetProvider.

2.2 Web Service

The Web app @ Azure cloud is implemented as a Node.js service that communicates with clients via REST requests that are processed by an Express server. Communication between clients and the Web app is through secure HTTPS communication channel. The Web app is responsible for the following more important functionality:

- Clients' registration.
- Clients profiling and segmentation.
- Sending push messages to a specific visitor or group of visitors who are from the same segment.

The registration of visitors is realized automatically, without any action of the visitor. For this purpose, when the mobile app is first launched, it registers to use the OneSignal service to be able to receive push notifications. After this registration, the value of the userId is retrieved, which is also used as the client identifier.

Visitor profiling aims to deliver personalized content. For this purpose, the profiling module analyses the following information sent by the mobile app:

- Which exhibits the visitor viewed and how much time was spent on each exhibit.
- Which type of exhibit description the visitor chooses (text, photos, audio, or video).

If it is not possible to profile a customer based on this information, the Web service sends a push notification after opening which the visitor is prompted to complete a short survey.

Visitor *segmentation* aims to assign visitors to a particular interest group or geofence. The main purpose of segmentation is to optimize the sending of push messages. Instead of sending the same message to several visitors in sequence, it can be sent once to visitors from a given segment. For example, all visitors from the "painting" segment can be informed via a push message that the museum has acquired a new painting.

Push notifications are used as an additional channel to deliver messages and content to visitors. Business logic uses OneSignal service to send push notifications to its users. In turn, OneSignal uses Google Firebase service to send the push notifications. This communication channel is mainly used in open-air museums. In this case, visitors travel longer distances and cannot constantly look at the mobile app display. For this reason, the mobile app must be able to run in the background. The Web app sends a push message only when the visitor is in the vicinity of an exhibit described as geofence. Push notification can also be used as a reliable channel to deliver information when visitors need to be evacuated.

2.3 Databases

The business logic from the Web app uses a MongoDB NoSQL database. The connection between the Web app and the database is implemented through Azure Cosmos DB for MongoDB. Three collections are used at this stage: Clients, Exhibits and Geofences.

The *Clients* collection contains information about each client of the service. The userId value obtained when registering the mobile app to receive push notifications through the OneSignal service is used as the key to this collection. For each client, the following information is stored: current location of the client (country, city, museum name, room name and GPS position); profile information (exhibits viewed by their type, time to view each exhibit, answers to survey questions) and the segments to which the client is assigned.

The *Exhibits* collection describes any museum exhibit that has an associated NFC tag or BLE beacon. The sensor ID is used as the key to access this collection. For each exhibit, the following information is stored: short and long name of the exhibit; author, if known; year or estimated period of production; where the exhibit is stored (location); type of exhibit, e.g. painting or sculpture; technique used; material; size and weight; whether the exhibit is an original or a copy; whether the exhibit has historical and cultural value, keywords to describe the exhibit; media description of the exhibit (text, photos, audio files, video files).

Fig. 2 shows structure of the documents from Exhibits collection. For exhibits associated with a beacon, two additional parameters are set: inscopeDistance - distance below which a request is sent to the Web app and outscopeDistance - distance above which the beacon stops being analyzed.

```
  tagid: "043827b2392b80"
  name: "St. Mary Magdalene"
  longname: "Saint Mary Magdalene"
> location: Object
> generalinfo: Object
> phisicaldescription: Object
> keywords: Array
> history: Object
> text: Array
> images: Array
> author: Object
```

a)

```
  tagid: "fda50693-a4e2-4fb1-afcf-c6eb07647801-00010202"
  inscopeDistance: 1
  outscopeDistance: 3.5
  name: "La belle jardinière"
  longname: "La belle jardinière"
> author: Object
> location: Object
> generalinfo: Object
> phisicaldescription: Object
> keywords: Array
> history: Object
> text: Array
> images: Array
```

b)

Fig. 2. Exhibits collection document structure: a) exhibit associated to NFC tag; b) exhibit associated to BLE beacon.

When a visitor scans an NFC tag associated with an exhibit or receives information from a BLE beacon it is near, the mobile app sends a REST request to the Web app to obtain exhibit information. In the request, information about the visitor ID and the sensor ID is passed. The business logic checks in the Clients collection which segments the client belongs to. A find request is then generated to the Exhibits collection to obtain filtered exhibit information depending on the client's preference for the type of media used to describe the exhibit. The response is in JSON format, which the mobile app renders as a Tabbar component. Each tab describes a specific media.

The Geofences collection is used to describe exhibits at open-air museums. In this case, the exhibit may be an entire building or a large object. Objects are described as polygons in Geo JSON format with which MongoDB database works by default. The business logic provides the functionality to check which geofences a visitor falls into at a given point in time, as well as which geofences the visitor is near. For this purpose, MongoDB's built-in capability for working with geospatial queries is used. Using operators $geoWithin and $geoIntersects, the position of the visitor can be checked to see if it falls within one or more geofences. Using operator $near we can get a list of geofences the visitor is close to at a specified minimum and maximum distance in meters. To use the built-in capabilities of MongoDB to convert the geofences into geo hashes for fast processing of geospatial queries, we need to use the 2dsphere geospatial indexing of the Geofences collection documents. If the visitor is near an object, the business logic sends a push message that contains brief information about the object. If the visitor enters a geofence that describes an object, a push message is sent that can be used to obtain detailed information about the object depending on the visitor's preferences.

3 Experimental Results

The proposed service architecture is validated in simulated environment. We have built a prototype of the proposed architecture which includes: (1) Web app @ Microsoft Azure cloud; (2) MongoDB database @ Azure cloud; and (3) Mobile app for Android OS. The Web app sends push notifications through a Firebase Cloud Messaging (FCM) service. FCM can be used to send push messages to mobile applications for Android and iOS operating systems. To use FCM, a Google Firebase access registration is required. Next, an Android OS App project is created with Cloud Messaging enabled. The Web service uses FCM indirectly through a service called OneSignal, which provides an API for sending push messages to a specific user or users in a specified segment.

NFC tags from different companies and BLE beacons from Kontakt.io were used to test the application (see Fig. 3).

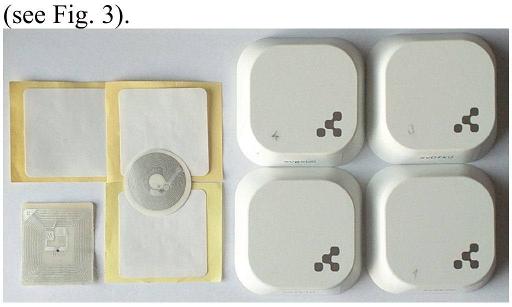

Fig. 3. NFC tags and BLE beacons used to test the app.

A hybrid mobile application has been developed, as well as plugins for it, through which the application can scan for the presence of nearby BLE beacons when not in focus and can read and transmit to the Web app the GPS position of the client. Scanning NFC tags is only possible if the app is in focus. The mobile app has been tested with various mobile devices and Android OS versions 9, 11 and 12. Fig. 4a shows the notification indicating that the foreground service through which the reading of the client's GPS position is implemented is active. A debug mode is provided to display the information that the foreground services retrieve (see Fig. 4b).

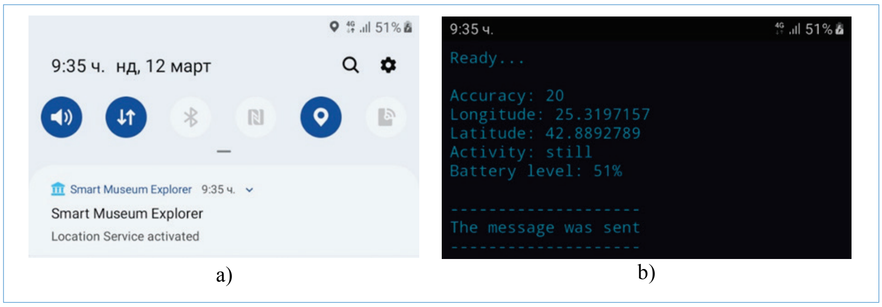

Fig. 4. Foreground service (Android 11) for reading GPS positions: a) notification from the service; b) information that the service retrieves (debug mode).

The Exhibits collection contains information for 20 exhibits associated with real sensors (10 NFC tags, 10 BLE beacons) and 1000 exhibits associated with virtual sensors. The Clients collection contains information about 10 real mobile clients and 1000 virtual clients. The Geofences collection contains information about 38 geofences from the Ethnographic open-air museum "Etar". Experiments have been done to analyse the response time for queries to the database using the MongoDB Compass application. The retrieval of information from each collection is in real time - no more than 4ms, including all geospatial queries. Web service response delay due to Internet connection latency depends mainly on the position of the servers where the Web app is deployed. When using servers in Europe this time is under 500ms. To this latency must be added the time to retrieve the images associated with the exhibit. All media resources are uploaded to repository servers. Only the URL to the resources is stored in the database (see Fig. 5).


```
images: Array
  0: Object
    title: "Full size"
    url: "https://i.imgupx.com/QMVOYDIw/picture-1.jpg"
  1: Object
    title: "Head 1"
    url: "https://i.imgupx.com/KQyvSzUz/picture-2.jpg"
  2: Object
    title: "Head 2"
    url: "https://i.imgupx.com/UfbArqvU/picture-3.jpg"
  3: Object
    title: "Head zoomed"
    url: "https://i.imgupx.com/OxxMjAnv/picture-4.jpg"
```


Fig. 5. Description of the photos in the database

When an NFC tag is scanned as well as when approaching a BLE beacon, the mobile app receives exhibit information that is filtered depending on the visitor profile. This information is displayed in the same way regardless of the type of sensor it is associated with. Fig. 6 shows what a visitor who prefers the exhibit description to be in text and photos sees. Similar information is obtained when a visitor enters a geofence that is associated with an exhibit. After opening the push notification, the application is launched if it is not running, and similar structured information is displayed.

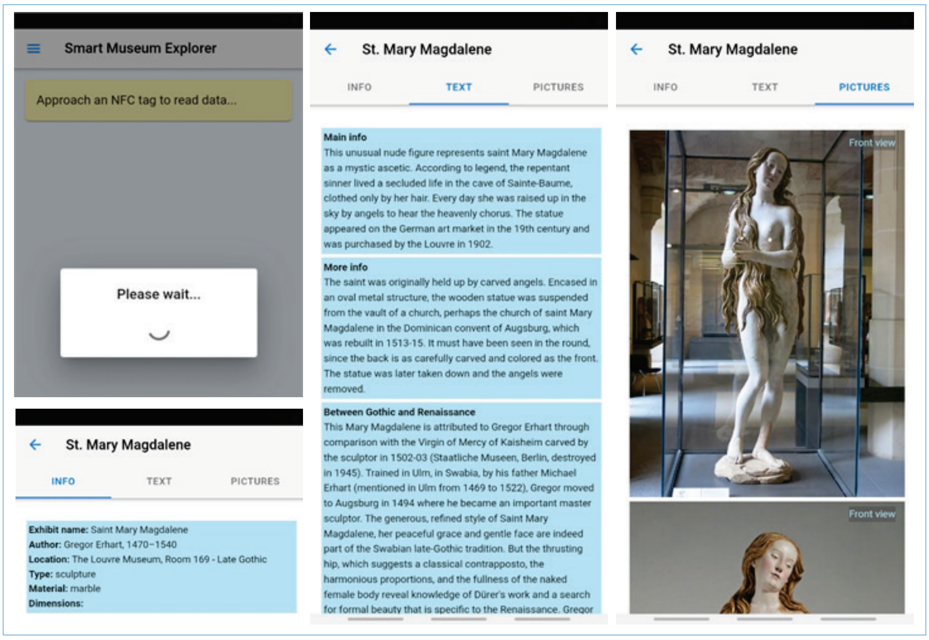

Fig. 6. Information about an exhibit associated with an NFC tag (Android 9)

4 Conclusions and Future Work

The paper presents architecture of the app which delivers penalized content to museum visitors. At this stage, the proposed architecture is validated in simulated environments. The delivery of personalized content is based on dynamic segmentation and profiling the visitors depending on their location and preferences. To improve visitor engagement, notification of available new content is implemented through push notifications. The experiments show that geospatial requests are implemented in real-time using MongoDB's collections hosted at Microsoft Azure cloud.

In the future it is foreseen to analyze the engagement and satisfaction of visitors of real museums. By analyzing these factors, museum professionals can gain valuable information about the needs and preferences of their visitors and use this information to improve the overall museum experience. In this way, they can ensure that museums remain relevant and engaging for different visitor groups and that they continue to serve as important educational and cultural institutions.

Acknowledgements.

This research has been partially funded by Bulgarian Ministry of Education and Science [Project № 2209E].

References

Antoniou, A., Katifori, A., Roussou, M., Vayanou, M., Karvounis, M., Kyriakidi, M., & Pujol-Tost, L. (2016). Capturing the visitor profile for a personalized mobile museum experience: An indirect approach. In Cena, F., Desmarais, M., Dicheva, D., & Zhang, J. (Eds), *24th ACM Conference on User Modeling, Adaptation and Personalisation (UMAP 2016), Workshop on Human Aspects in Adaptive and Personalized Interactive Environments (HAAPIE)* (pp. 1-10). https://ceur-ws.org/Vol-1618/HAAPIE_paper1.pdf

Dragović, N., Stankov, U. & Vasiljević, Đ. (2018).Contactless Technology as a Factor of Tourism Industry Development - A Review of Current Practices and Future Directions. *Economic Themes, 56*(2) 179-202. https://doi.org/10.2478/ethemes-2018-0011

ElDamshiry, K. K., ElFouly, H. (2022). Open Air Natural History Museums Transformation: From static to interactive. *Advance Engineering Science*, *54*(2), 991-1004.

Fan, Y. L. (2022). Impact of generativity on museum visitors' engagement, experience, and psychological well-being. *Tourism Management Perspectives*, 1-14.

Petrelli, D., & O'Brien, S. (2018). Phone vs. Tangible in Museums: A Comparative Study. In *Proceedings of the CHI18 ACM CHI Conference* (pp. 1-12).

Shoji, Y. A. (2021). Location-based Reminder for Memorizing What Visitors Learn at a Museum. In *BIRDS+ WEPIR@ CHIIR* (pp. 79-87).

Vassilakis, C. P. (2017). exhiSTORY: Smart exhibits that tell their own stories. *Future Generation Computer Systems*, 542-556.

Received: March 15, 2023
Reviewed: April 24, 2023
Finally Accepted: May 05, 2023