Delivering Personalized Content to Open-air Museum Visitors Using Geofencing

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

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

Abstract. This paper presents the architecture and implementation of a service that delivers personalized content to open-air museum visitors. The service uses push notifications to deliver this content. Notifications can be delivered to all visitors or personalized - to a group of visitors or to a specific visitor. The service segments museum visitors according to their location and their profile, which is built dynamically over time. For the geospatial segmentation of visitors, geofencing is used - each visitor is assigned to a segment that corresponds to a specific geographic area - part of an open space or exhibit. The service allows localization of visitors by their GPS coordinates or by estimating their distance from Bluetooth Low Energy (BLE) beacons. The geofences are described as a polygons or circles. Geospatial segmentation is implemented using a NoSQL database MongoDB, which has built-in capabilities for working with geospatial queries. Depending on the profile, each visitor falls into one or several segments: professional researcher, non-professional researcher, inspiration seeker, casual visitor, and visitors with disabilities. For each visitor, personalized content is delivered, depending on the segments to which the visitor is assigned. The necessary experiments have been conducted and analyzed to prove the applicability of the service for real-time delivery of personalized content.

Keywords: Open-air Museum, Geofencing, Users segmentation, Distributed services, Push notification.

1 Introduction

The first open-air museums date back to the 19th century. Open-air museums are institutions which, on a scientific basis, have the task of presenting and preserving folk art and cultural heritage in the form of outdoor exhibitions (Slobodova, 2022). At this stage, a variety of open-air museums are created, which can be focused on a specific urban environment, archaeological landmarks, ancient cultures or military equipment. Open-air museums are based on the relationship among science, environment, culture, nature and society. This requires an interdisciplinary approach to the creation of open-air museums. The presentation of exhibits to visitors also presupposes knowledge of multiple humanities and natural sciences.

Digital Presentation and Preservation of Cultural and Scientific Heritage. Conference Proceedings. Vol. 12, Sofia, Bulgaria: Institute of Mathematics and Informatics – BAS, 2022. ISSN: 1314-4006, eISSN: 2535-0366

The design of museums both indoors and outdoors is focused on transforming static exhibitions into interactive exhibitions (ElDamshiry, 2022). The aim is not only to ensure visitor satisfaction but also to increase their engagement with the subject matter presented (Fan, 2022). Combining an enjoyable experience with entertainment should provide an opportunity to learn new facts for the purpose of education (Vassilakis, 2017). This is a hard-to-achieve goal if personalized content is not delivered to any visitor or group of visitors with similar interests. Personalizing content without the use of modern technology is difficult to achieve due to the specific interests of each visitor and the insufficient number of guides.

Most developments related to converting static exhibitions into interactive exhibitions use mobile devices in order to deliver additional information to visitors by scanning Quick Response (QR) tags, touching Near-field Communication (NFC) tags or using augmented reality (Sepe, 2022). Such developments do not consider the specific capacities of visitors to perceive the information and their preferred media formats for presenting this information. Moreover, some visitors remain frustrated by the additional information received and easily refuse to use it at a later stage. This requires delivering personalized content to visitors. It should be generated dynamically, depending on the visitor's location and the specifics of his/her profile. Visitor profiling can be implicit, explicit or combined (Antoniou, 2016). Implicit profiling implies filling in an electronic form or survey. For example, the official mobile app for exploring the Louvre (Lourve, 2022) uses a form to inform the system of the time the visitor intends to spend in the museum; their preferred exhibit types; and whether the visitor has reduced mobility. Since this information may not be entered or may be entered partially, explicit visitor profiling is also required. It is implemented by analysing the visitor's movement within the museum and counting the time the visitor stays near the exhibits.

Tracking visitors can be implemented using data from the GPS receiver embedded in mobile devices or based on distance estimation from Bluetooth Low Energy (BLE) beacons. Beacons are especially suitable to use when localizing small objects and exhibits. For objects with larger dimensions, e.g., a neighbourhood, a building or a part of a building, geofencing is suitable (Shoji, 2021). The term geofence (geo defence) defines a virtual perimeter for a real geographical region. This can be a circle of a certain radius or a polygon that describes the perimeter of the protected object. Using geofences to create Location Based Services (LBS) is called geofencing. In this way, the services can determine when their clients enter or leave the protected regions. The geofencing-based services usually sends push notifications to users' mobile devices when they approach places related to the geofences. The main advantages of geofencing are as follows:

- The service can engage users with personalized info that are right for them.
- Segmentation of users depending on the geographical region in which they are located.
- Analyse user behaviour by combining information about their browsing behaviour and activity.

The global market share of services using geofencing is expected to increase by 23.96% and total $2.21 billion during the period 2022-2025.

There are numerous cloud services that support geofencing and segmentation of users. OneSignal (OneSignal, 2022) is a multiplatform notification service. OneSignal supports segmentation of users based on their GPS position and using custom tags. The problems with OneSignal are basically two: 1) Geospatial segmentation is only a rectangular shaped area; and 2) The number of segments is limited - 6 on the free plan and 20 on the professional plan. The PlotProjects (PlotProjects, 2022) service offers better possibilities related to segmentation of users. The free plan allows defining up to 1000 geofences. With this service, we can describe the contour of a geofence in two ways - by a polygon and a circle with a certain radius. PlotProjects allows sending notifications when approaching both geofences and BLE Beacons described by their Universally Unique Identifiers (UUIDs). Segmentation of users based on custom tags cannot be implemented in this service. The number of supported mobile native and hybrid frameworks is limited. The strengths of PlotProjects may be integrated into OneSignal project, but only with an active paid plan. The specific limitations of these services make it necessary to create a service that allows segmentation of museum visitors without imposing restrictions on the number and type of segments.

The main objective of this paper is the description of the design and preliminary tests of a distributed service for personalized content delivery in open-air museums. The service segments visitors according to their location and profile. Push notifications are used to deliver content to visitors. After tapping on the notification, the visitor receives structured Web content related to the object or exhibit in volume and media formats according to its profile.

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 finally Section 4 concludes the paper and presents some ideas for future work.

2 The Proposed Service Architecture

The level of complexity of the service and the requirement to process geospatial queries in real time implies that the service has a distributed architecture. Several microservices are used for this purpose. Their operation is synchronized through a message broker. The microservice architecture allows the partitioning of a large application into a collection of small independent services, each with its own business logic and access to a specific database. The microservice architecture enables the rapid building of complex programming systems with a high level of reliability.

Fig. 1 shows the architecture of the proposed service. Users of the service (museum visitors) must have a hybrid mobile application that can be installed on Android and iOS mobile devices. Using this app, visitors register to receive personalized information about museum exhibits. Instead of visitors being constantly engaged with their phones, they are notified of the information they want using push notifications. If a visitor is interested in this information, she/she can see it in preferred format after tapping on the notification window, but they can also ignore it.

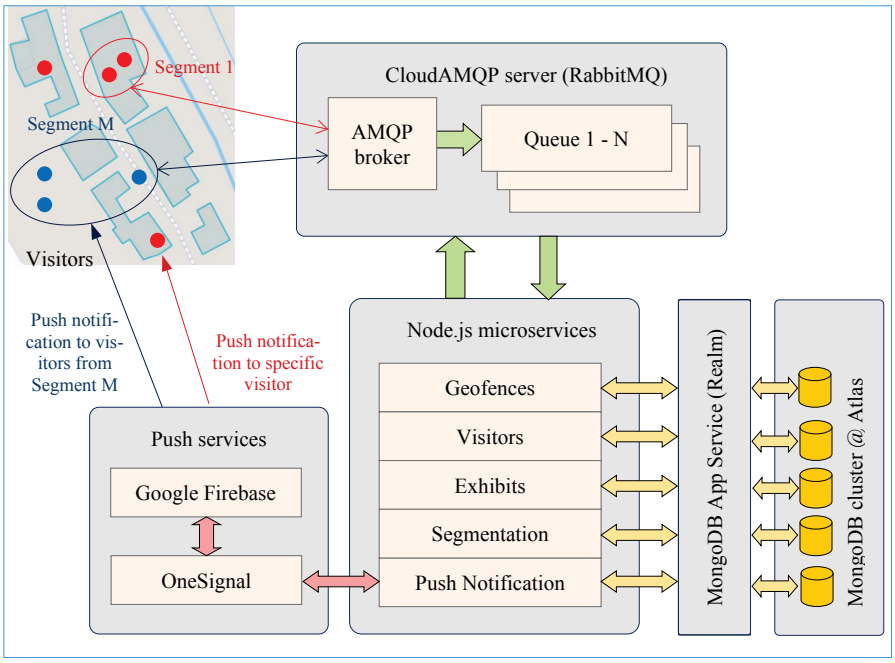

Fig. 1. General overview of the service architecture

The service delivers information depending on the position of visitors and their profiles. For this purpose, visitors are segmented depending on what object or exhibit they are in proximity to. The entire business logic is implemented through five Node.js microservices. Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Each microservice works with its own database. The NoSQL database MongoDB in the Azure cloud infrastructure is used. The interface between the microservices and the databases is implemented using the MongoDB App Service (former name Realm).

Communication between the service components is implemented through a message broker. In this development, RabbitMQ server hosted at Google cloud is used. Multiple message queues are used through which the necessary communication channels between clients and microservices and between microservices themselves are established. Advanced Message Queuing Protocol (AMQP) is used for this purpose. It guarantees reliability in message delivery.

Google Firebase is used to send push notifications to museum visitors. Access to this service is implemented using the OneSignal REST API. Google Firebase allows sending notifications to all visitors of a segment(s) as well as to a specific visitor. For this purpose, when the mobile application is initially launched, the client registers to use the Google Firebase service. After successful registration, the mobile app receives a unique "player id". This id is delivered via the RabbitMQ server to the Visitors microservice. This microservice records information about each registered user into the database. The

Visitors microservice has the task of creating and updating the state associated with each visitor's profile. For each client of the service, the "external user id" is also obtained through which push notifications can be sent to the specific user.

The Geofences microservice allows a quick description of geofences in GeoJSON format. Three types of GeoJSON objects are supported: a circle with a certain radius, a simple polygon and a multipolygon (polygon with "holes"). This microservice allows exporting the data as a GeoJSON file in order to manually create the database describing the geofences. It is also possible to automatically build and update this database. To make this microservice work, the Mapbox GL-JS library is used, which supports the use of GPS maps. The data is obtained from open sources such as OpenStreetMap and NASA. The Geofences microservice also allows the description of geofences using geohashes. A geohash is a string of a certain length that is a unique identifier of a specific geographic region with a rectangular shape. The longer the string, the more accurately the region is described. Points that are in close geographical proximity will have the same geohash. This allows a very quick calculation of whether a point (visitor's position) is inside or outside a region.

The Exhibits microservice describes the museum's exhibits. The description of exhibits in the database is consistent with two standards - ObjectID and Dublin Core. The ObjectId standard allows for a standardized description of collections of archaeological, cultural, and artistic objects, while Dublin Core is used to describe digital resources (video, images, webpages, etc.). Each piece of information about an exhibit is associated with segments to which each museum visitor is assigned (professional researcher, non-professional researcher, inspiration seeker, casual visitor, and visitor with disability). Using the Exhibits microservice, museum designers can easily describe their exhibits.

Push notification microservice uses access to OneSignal service to send push notifications. This microservice can send push notifications to all museum visitors, to a group of visitors from one or several segments, or to a specific visitor. The OneSignal REST API is used for this purpose. Two types of push notifications are supported:

- A notification, which contains the URL of a resource that dynamically generates information related to a specific exhibit. By using custom tags to the resource, content personalization is allowed. Tapping on such a notification launches the browser on the mobile device through which the resource corresponding to the received URL is rendered.
- A notification that contains custom tags that describe the exhibit ID and the segments to which the visitor belongs. Tapping on this type of notification activates a mobile app resource that displays the exhibit information.

3 Experimental Results

The proposed service architecture is validated in simulated environment and real environment - Ethnographic Open-air Museum "Etar". We have built a prototype of the proposed architecture which includes: (1) Microservices; (2) Databases @ Azure cloud; (3) RabbitMQ server; and (4) Mobile app. All microservices described in Section 2

have been implemented. We will discuss in more detail how the Geofences and Segmentation microservices are implemented and work because they are directly related to the topic of this paper.

3.1 Geofencing

The Geofences microservice is used to implement the basic functionality of the service - geofencing. This microservice builds a database that describes each geofence in GeoJSON format according to the syntax required to use MongoDB operators for geospatial queries. Microservice Geofences runs as a Node.js application that uses the following programming frameworks and libraries:

- AMQP - a library for creating AMQP 0-9-1 clients for Node.js.
- Express - a web framework for Node.js. This framework is used to communicate with the microservice through a browser.
- Realm Web - package for authentication and communication with MongoDB databases using HTTPS protocol.
- Mapbox GL-JS - an open source JavaScript library that uses Mapbox GL to render interactive GPS maps. It is used to visualize museum objects on a GPS map in order to find the contours of geofences.
- Mapbox GL Geocoder – this JavaScript library adds a geocoding control to a GPS map, allowing users to search for objects on the map.
- Turf - JavaScript library for geospatial data analysis and processing.
- A JavaScript library has been developed to support geohashes - filling a polygon with geohashes using different strategies and precision.

Fig. 2 shows the user interface of the Geofences microservice. After successful login, the geocoder searches for the desired location, region, point of interest, etc. In this case it is the Ethnographic open-air museum “Etar”. The contours of all geofences are then entered. From the menu at the top right, you can select a contour description using a polygon or a circle with a given radius. All entered geofences can be visually edited or deleted.

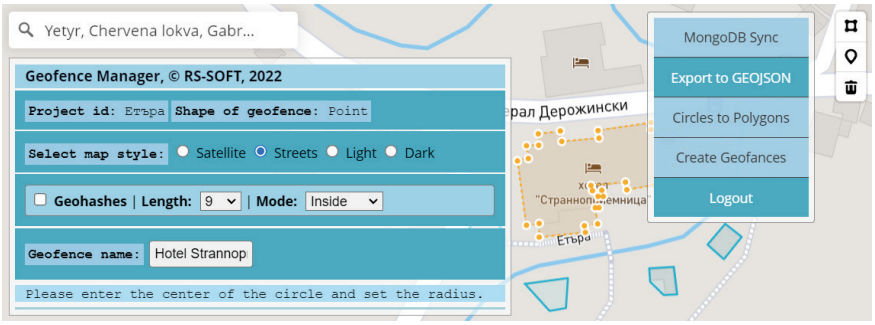

Fig. 2. Geofences microservice user interface

The microservice provides the ability to automatically convert geofences described by a circle to a polygon (button “Circles to Polygons”). This is necessary if specific geospatial queries to the MongoDB Geofences database are used. At any time, using the

"Export to GEOJSON" button, it is possible to export the created geofences to a file in GeoJSON format. The contents of this file can be imported directly into the MongoDB collection. It is also possible to automatically create and update the database using the "MongoDB Sync" button. Experiments show that it takes about 15-20 minutes to create 38 geofences from the Ethnographic Open-air Museum "Etar".

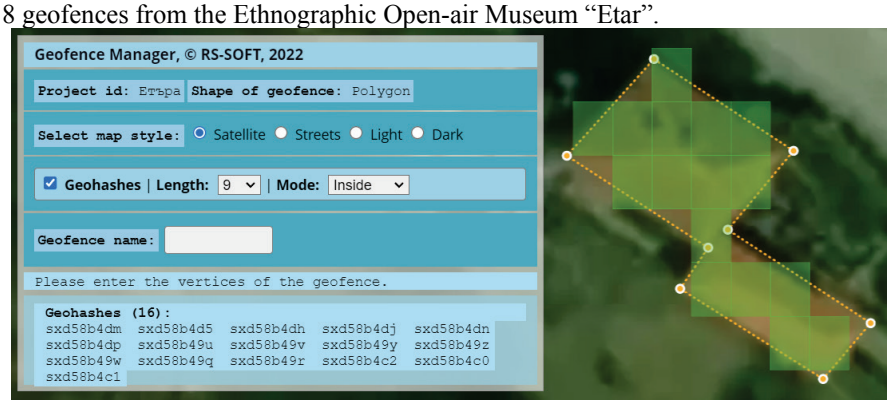

Fig. 3. Transform a GeoJSON Polygon to a list of geohashes that cover it

Fig. 3 shows how the polygon geohashing module works. When Geohashes mode is active, for each selected geofence the list of geohashes by which the polygon is filled is found. For each geofence a different filling mode and precision can be selected depending on its area. Two filling modes are supported:

- Inside - all geofences for which the center is inside the geofence are searched.
- Intersect - searches for all geohashes that fall inside the geofence or intersect with the contour of polygon at a specified minimum coverage factor. For example, a minimum coverage factor of 0.6 returns all geohashes for which at least 60% of the area of geohash box is inside the geofence contour.

3.2 Segmentation

The service supports user segmentation to deliver personalized content to users. Two types of segmentation are used:

- Location-based segmentation - each visitor is assigned to a segment that is associated with a specific geofence.
- Profile-based segmentation - each visitor is assigned to one or more segments depending on their profile.

The Segmentation microservice main task is to collect information about each visitor of the museum and on its basis to realize their segmentation. Geo segmentation is implemented entirely through MongoDB's built-in geospatial data handling capabilities. Finding which visitors are in a specific geofence or which geofences are in a given proximity to a visitor is implemented with a single geospatial query, without the need to process the result on the microservice side. The $geoWithin and $geoIntersect operators can be used to find which users are within a geofence. If the visitor's position is

described by a GPS point both operators can be used. If the visitor's position is described by a geohash it is appropriate to use the $geoIntersect operator.

Table 1 shows the time in ms required to find visitors falling within the contour of a selected geofence. For this purpose, five databases were created with different numbers of visitors, whose positions were randomly generated within the region in which the Ethnographic Open-Air Museum "Etar" is located. The analysis was implemented using the MongoDB Compass app and Explain Plan tool. In the absence of database indexing, doubling the number of visitors results in approximately doubling the query execution time. When using the 2dsphere index, query execution time is less than or up to 1ms. There is no significant difference between the service times of $geoWithin and $geoIntersect geospatial queries. When the geofences to which a visitor(s) is located are to be obtained (minimum and maximum distances are specified), the $near and $nearSphere operators can be used. The response time for these queries is under 1ms when using the 2d or 2dsphere index. The very good results when using indexing are to be expected, since with 2d and 2dsphere indexing MongoDB computes a geohash for GPS coordinates within the specified location range and then indexes the geo hash values (Gonçalves, 2021).

Table 1. Time in ms to find visitors that fall into a geofence.

MongoDB geospatial operator		Number of visitors				
		500	1000	2000	5000	10000
$geoWithin	without indexing	1	2	4	7	16
	2dsphere index	<1	<1	1	1	1
$geoIntersection	without indexing	<1	1	4	7	15
	2dsphere index	<1	<1	1	1	1

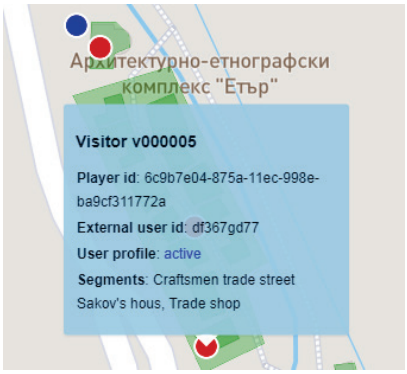

Fig. 4. Graphical representation of users' segmentation

Fig. 4 shows the graphical interpretation of the Segmentation microservice operation. Visitors that are within the geofence boundaries are shown as red dots and the rest are shown by blue dots.

3.3 Mobile App

A hybrid mobile application (HTML, CSS and JavaScript) has been developed using the Cordova framework (CLI v.11.0.0). The user interface is built using a Framework7 v.6.3.0. The app can be installed on Android and iOS mobile devices. The mobile application implements the following basic functionality:

- Registers the app to receive push notifications via Google Firebase.
- Reads and processes raw data from the built-in GPS receiver, accelerometer, and BLE beacons.
- Visualizes a form with question(s) through the answers to which the visitor's profile is built.
- Visualizes personalized content generated by the back-end part of the service.
- Sends messages to the AMQP broker using the RabbitMQ REST API in order to register visitor; when the visitor's location changes; and when the visitor's profile is updated.

Fig. 5 shows screenshots from the mobile app. They are obtained when a visitor is in close proximity to the Jewelry-making workshop. The visitor receives all the available information (text, pictures, audio, video and links to Web resources) as he/she is assigned to the "professional researcher" segment.

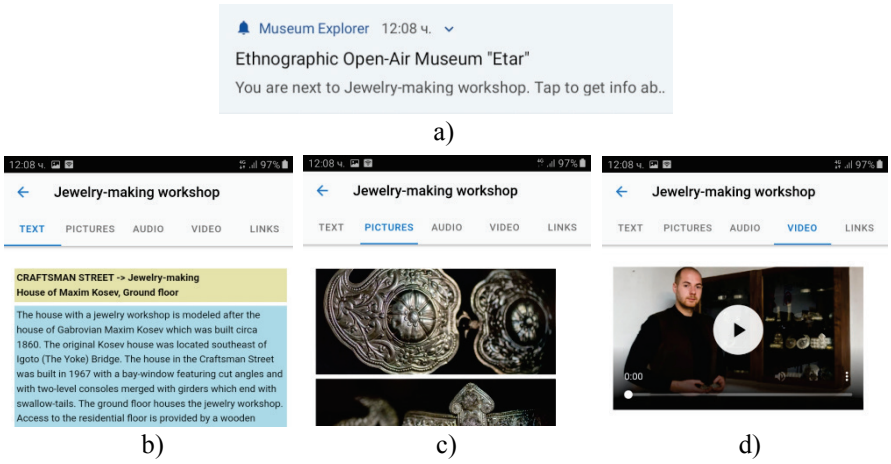

Fig. 5. Mobile app screenshots: a) push notification; b) text content; c) pictures; and d) video

4 Conclusions and Future Work

The paper presents distributed architecture of the LBS which delivers personalized content to open-air museum visitors. The proposed architecture is validated in simulated and real environments. The delivery of personalized content is based on dynamic segmentation of visitors depending on their location and profile. To improve visitor engagement, notification of available new content is implemented through push messages.

The experiments show that geo segmentation is implemented in real-time using MongoDB's built-in geospatial query capabilities.

In the future it is foreseen to analyze the engagement and satisfaction of real visitors of the Ethnographic Open-Air Museum "Etar".

Acknowledgements.

This research has been partially funded by the Bulgarian Ministry of Education and Science, project № 2203E and project № 2209E.

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.

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

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

Gonçalves, H. C.–P.–B.–S.–B. (2021). Spatial Data Handling in NoSQL Databases: A User-centric View. *Proceedings XXII GEOINFO*, (pp. 167-178).

Lourve. (2022). *The Louvre App*. Retrieved June 1, 2022, from The Louvre museum: https://www.louvre.fr/en/new-app

OneSignal. (2022). *OneSignal - Customer Messaging Delivered*. Retrieved June 1, 2022, from https://onesignal.com/

PlotProjects. (2022). *The best geofencing software plugin for mobile apps*. Retrieved June 1, 2022, from https://www.plotprojects.com/

Sepe, F. M. (2022). Making Smarter Museums Through New Technologies. In *Handbook of Research on Museum Management in the Digital Era* (pp. 1-24). IGI Global.

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

Slobodova, L. R.–P.–K.–M. (2022). Open-air Museums–the Future of the Presentation of Spiritual and Architectural Heritage. *Muzeologia* , 5-18.

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

Received: June 12, 2022
Reviewed: July 07, 2022
Finally Accepted: July 15, 2022