

Article

Application Programming Interface for the Cloud-Based Management of Gamified eGuides

Artur Kulpa and Jakub Swacha *

Institute of Management, Department of IT in Management, University of Szczecin, ul. Cukrowa 8, 71-004 Szczecin, Poland; artur.kulpa@usz.edu.pl

* Correspondence: jakub.swacha@usz.edu.pl

Received: 1 May 2020; Accepted: 4 June 2020; Published: 6 June 2020

Abstract: The popularity of smartphones and widespread access to mobile internet removed earlier barriers to reliance on mobile applications run on visitors' own devices for guidance in tourist attractions. At the same time, the tourists' rising expectations call for solutions that can increase their engagement and satisfaction, such as gamification. Despite the availability of platforms for both general-purpose gamification and configurable eguides, until now, there have not been any ready-made solutions of this kind supporting the implementation of gamification for eguides. In this paper, we would like to present a solution filling this gap: the eMused.eu Application Programming Interface, which can be used by mobile applications (web or native) to access both tour content and gamification functionality provided on a cloud.

Keywords: API design; information systems in tourism; gamification as a service

1. Introduction

For various reasons, including cultural (increasing share of individual visitors and decreasing share of group visitors) and economic (increasing wages and decreasing prices of electronics), tourist attractions tend to replace or complement human guides with electronic devices that can be operated by visitors themselves for the same purpose [1] (p. 138). With the widespread proliferation of smartphones [2], this can be simplified to providing the visitors with a mobile application they can use on their own mobile devices in a Bring-Your-Own-Device (BYOD) scheme. However, providing a rudimentary eguide with audio, text and static or moving images can be not enough, as rising tourist expectations call for solutions that can increase visitors' engagement and satisfaction [3] (p. 173). One such solution is gamification; that is, the use of game design elements in non-game domains such as tourism, which has been reported to affect visitors' behavior and perception of the visiting process [4].

Since the early days of mobile eguides, there has been an effort to separate tour content from the software used to present it to the visitors (see e.g., [5] (p. 70)), so that even nontechnical tourist attraction personnel could develop and make changes to the content provided to the visitors. The separation of the gamification layer from the main eguide functionality came later, but it is now a well-established approach (see [6] and works cited therein).

Such a separation opens up the possibility of simplifying eguide development, provided that a set of standard content- and gamification-related functionalities is made available via a dedicated application programming interface (API). While there are well known web platforms providing API for both general-purpose gamification (such as Google Play Game Services [7]) and mobile eguide content (such as the izi.TRAVEL API [8]), there are no ready-made solutions of this kind aimed at the implementation of gamification for eguides.

Such an API has been developed within the framework of an international project involving several tourist attractions from the South Baltic area [9]. In this paper, we would like to present the design and development of this API, named the eMused.eu API [10], which can be used by mobile applications (web or native) to access tour content and gamification functionality provided on a cloud.

2. Related Work

To the best of our knowledge, until now there was no similar published work to the one proposed here, which is aimed at supporting gamified mobile eguides. We are, however, aware of related work, relevant either to gamification or eguide management platforms.

One of the oldest platforms for eguide management is TAP, an open-source solution originally developed by Indianapolis Museum of Art, comprising authoring tools built on top of the Drupal content management system and mobile applications for iOS and web browsers [11]. One of its key features is support for the TourML format for describing guided tours [12]. While the development of the original project stalled around 2015, the work is continued in the Guidepost project [13].

With over 8500 provided tours, izi.TRAVEL is probably the biggest eguide management platform in the world [14]. Despite being operated by a commercial company, it provides a free cloud-based eguide content management system, mobile app and an open API, using which third-party software of any kind can connect to the eguide content hosted at izi.TRAVEL [8].

A somewhat related vein of research is dedicated to the development of platforms for the management of interactive tours based on Internet of Things devices, such as meSch [5].

Regarding gamification, there are a number of cloud-based platforms offering Game Backend as a Service (GBaaS) providing an API to basic gamification functionalities. A good comparison of eight such services offered by commercial vendors (Google Play Game Services, Yahoo Backend Game Service, GameUp, Flox, GameSparks, Fresvii, Kumakore and Photon) was published in [15]. According to this source, the widest coverage of gamification functionality among the compared platforms is provided by Google Play Game Services [7], handling, among others, such mechanisms as Leaderboards, Achievements, Quests and Gifts.

There are also open-source gamification solutions providing APIs to handle the gamification-related tasks. The best-known platform of this kind is the Gamification Servers software developed by National Geospatial-Intelligence Agency in collaboration with The MITRE Corporation, which provides basic gamification functionalities, such as points, badges and leaderboards [16]. A somewhat similar coverage of gamification functionalities is provided by the Gamification API by Komanda Phanzu and Aurélien Thévoz [17].

3. Problem Context

The motivation for the presented work came with the development of BalticMuseums: Love IT!, an international project realized under the Interreg South Baltic Programme 2014–2020 and supported by the European Union from the European Regional Development Fund [9]. The main aim of this project is to develop multilingual BYOD mobile-guided tours providing an enhanced visitor experience, featuring multimedia content and gamification. Integrating the game design elements into the tours is thought to result in making the visitor experience more involving and amusing, and extend the single visit into a long-term relationship.

During an early phase of the project, it became clear that no single eguide application can satisfy the diverse needs of the project partners, comprising museums of various sizes and scopes. The work focused on developing a common eguide gamification framework which could then be adapted by the specific museums [18]. In the technological aspect of the framework, the search for software platforms to recommend for the development of gamified eguides yielded no satisfactory results; the considered GBaaS platforms were either too limited in the provided scope of gamification or too costly in the long run, and the open-source eguide management platforms featured no (or almost no) gamification support at all. This fueled the effort to develop a fully-fledged solution for eguide gamification implementation, rather than a thin wrapper linking eguide content management

systems with GBaaS platforms. From the very beginning, its design goals featured a wide area of possible usage, encompassing not only the diverse project partners but also future adopters. This complicated the requirement elicitation process, which eventually relied on as many as six different sources: gamification literature survey, surveys among museum visitors, hackathons organized with the goal of obtaining a fresh look at eguides, experience reports from the use of existing eguides, interviews with the development teams of existing eguides and interviews with future guide content developers [18].

As a result, we were able to define key requirements grouped in seven functional areas:

- Visitor account, providing authorization for guide users;
- Tour organization, providing storage and access to the content defining the tour structure;
- Tour and user data, providing storage and access to information defining the lowest-level items of both the tour and guide user portfolio;
- Quizzes and surveys, providing storage and access to the content of quizzes and surveys and handling responses;
- Team management, providing support for joining guide users into teams, so that team quests can be solved and team achievements attained;
- Visitor Activity Tracking, handling events generated by guide users and checking actions available for a specific tour element;
- Internationalization, providing functionality related to guide internationalization.

The gathered requirements served as a basis for a list of services that a gamified guide app can rely on to be available and, consequently, the specification of the eMused.eu API to be described in the next section. The API's name stems from the name of the web domain at which the first service conforming to it is hosted, but it is also the name of a collaboration network for the promotion of tourist attractions that provide visitors with gamified BYOD guides, as chosen by its founding members, the BalticMuseums: Love IT! project members [19]. Note that the network is open for all institutions regardless of the technology for implementing gamification they use, and the access to the guide gamification service hosted at emused.eu is free for all the network members.

4. Functional Coverage of the eMused.eu API

4.1. Visitor Account

One of the advantages of the BYOD mobile guides over rented mobile devices is that they belong to visitors, so they stay with them after a visit, allowing them to prolong their experience and possibly allowing designers to build gamification schemes that span over many visits. In order to make such a long-term connection possible, the visitor has to authorize the guide application. Note that, if the management of a specific tourist attraction does not wish to make such a connection, there is no need for actual authorization; while the eMused.eu API will still require authorization data to distinguish among its users, these could be generated randomly by the guide application itself (merely prompting users about it). There are two authorization methods supported: classic, using a username and password (discouraged, as users are reluctant to create yet another web account); and OAuth2-based [20], allowing authorization via a third-party service, e.g., one that the user may highly probably have already registered to, such as Facebook or Google.

The eMused.eu API endpoints from the Visitor account area are listed in Table 1.

Table 1. Visitor account endpoints.

Name	Description	Endpoint
New User Registration	Create new internal user from the provided data	POST/api/user_profile/
User Sign-in	Verify authorization data and log user in.	POST/auth/token
User Sign-in via social media	Verify authorization token and log user in.	POST/auth/convert-token
User Sign-out	Log out the current authenticated user.	POST/auth/revoke-token
User profile details	Retrieve the current authenticated user profile data (including achieved score, badges, items).	GET/api/user_profile/
User profile update	Update the current authenticated user profile data.	PUT/PATCH/api/user_profile/{id}/

4.2. Tour Organization

The eMused.eu API conforms to the following model of tour content organization: Site > Area > Route > Location > Item.

Site is the tourist attraction hosting the tour. It is assumed that the eguide application is connected to only one site at any moment.

Area is a section of the tourist attraction (in spatial terms) defined by its managers. It could be a separate building or just a part of it, e.g., hosting a temporary exhibition with separate tickets. Many tourist attractions comprise only one area.

Route is a, possibly ordered, set of locations at which respective points of interest are placed. A tour map is connected to a route; hence, only locations placed on a route will be visible on a map for the eguide user. Thus, there can be many routes for a single area, differing in the choice of locations (e.g., long/short) or just the form of presentation (e.g., technical drawing for adults/comic-like drawing for children). The set of supported languages is assigned to a route for the user's information of what to expect from choosing it, though in order for the user to get the actual content in the chosen language, its translation has to be available.

Location denotes a location in the physical sense; that is, a place a visitor can come to.

Item denotes a set of content to be presented to the user in a single eguide application screen (it could therefore also be thought about as the user's location within the application). As the same concept is used to handle the set of contents defining items that are part of the eguide user portfolio (e.g., the virtual guide character he/she chose, his/her avatar or the virtual possessions he/she has obtained), they also form the eguide application screens that could be viewed by users; the endpoints related to items will be presented in the next subsection.

The eMused.eu API endpoints from the tour organization area are listed in Table 2.

Table 2. Tour organization endpoints.

Name	Description	Endpoint
Area list	List available active areas (detailed data included). Filter available.	GET/api/area/
Area details	Retrieve specific active area data.	GET/api/area/{id}/
Area type list	List available area types.	GET/api/area-type/
Area type details	Retrieve specific area type data.	GET/api/area-type/{id}/
Location list	List available active location (detailed data included). Filter available.	GET/api/locations/
Location type list	List available location types.	GET/api/locations-type/
Location type details	Retrieve specific location type.	GET/api/locations-type/{id}/
Route list	List available active routes (detailed data included). Filter available.	GET/api/route/
Route details	Retrieve specific active route data.	GET/api/route/{id}/
Route type list	List available route types.	GET/api/route-type/
Route type details	Retrieve specific route type data.	GET/api/route-type/{id}/
Site data	List active sites.	GET/api/site/

4.3. Tour and User Data

The eMused.eu API uses Item as a universal container for sets of contents to be presented to the user in a single eguide application screen (it could therefore also be thought about as the user's location within the application, as opposed to Location, which specifies the user's location in the real world). Note that this concept also includes items that are part of the eguide user portfolio (e.g., the virtual guide character he/she chose, his/her avatar or the virtual possessions he/she obtained), as they also form eguide application screens that could be viewed by users.

The eMused.eu API provides controlled access to resources featured in the displayed items (i.e., an unauthorized access to the respective resources is impossible, even knowing their URLs).

The eMused.eu API endpoints from the tour and user data area are listed in Table 3.

Table 3. Tour and user data endpoints.

Name	Description	Endpoint
Badge list	List active badges (detailed data included). Filter available.	GET/api/badge/
Item list	List available active items (detailed data included). Filter available.	GET/api/item/
Item kind list	List available item kinds.	GET/api/item-kind/
Item kind details	Retrieve specific item kind data.	GET/api/item-kind/{id}/
Item type list	List available item types.	GET/api/item-type/
Item type details	Retrieve specific item type data.	GET/api/item-type/{id}/
Resources (BLOB type)	Retrieve specific resource file (if exists).	GET/private-media-api/{path}

4.4. Quizzes and Surveys

Quizzes are an effective means of implementing gamification, as exemplified by modern student response systems such as Kahoot! [21], and are especially helpful in strengthening the educational role of eguides. Regarding surveys, mobile eguides provide a much more convenient method of

collecting data compared to Paper & Pen Personal Interviews or even Computer-Assisted Personal Interviews. The surveys can be applied right after finishing a tour (or even, possibly in parts, during it) and the collected data, thanks to the wireless connection, can be processed right after survey completion. These two different functionalities are coupled because, on a technical level, they are implemented similarly, as surveys can be seen as quizzes without assumed right answers.

The relevant eMused.eu API endpoints are listed in Table 4.

Table 4. Quiz and survey endpoints.

Name	Description	Endpoint
Quiz list	List available active quizzes (detailed data included). Filter available.	GET/api/quiz/
Quiz details	Retrieve specific active quiz data with questions.	GET/api/quiz/{id}/
Quiz fulfillment list	List the current authenticated user's quiz fulfillments.	GET/api/quiz-fulfillment/
Quiz fulfillment details	Retrieve specific fulfillment data (with answers) of the current authenticated user.	GET/api/quiz-fulfillment/{id}/
New fulfillment of quiz	Save and return fulfillment data of the current authenticated user.	POST/api/quiz-fulfillment/
New answer to quiz question	Save, check answer and return response.	POST/api/quiz-answer/
Survey list	List available active surveys (detailed data included). Filter available.	GET/api/survey/
Survey details	Retrieve specific active survey data with questions.	GET/api/survey/{id}/
New fulfillment of survey	Save and return fulfillment data of the current authenticated user.	POST/api/survey-fulfillment/
New answer to survey question	Save, check answer and return response.	POST/api/survey-answer/

4.5. Team Management

Visitors of a tourist attraction expect to have a social experience [5] (p. 76). One of gamification techniques that addresses this need is setting challenges that require cooperation [22].

In order to make it possible for the users to face such challenges, the eMused.eu API provides endpoints for creating and joining teams. They are listed in Table 5.

Table 5. Team management endpoints.

Name	Description	Endpoint
Team list	Retrieve current authenticated user teams list.	GET/api/team/
Team details	Retrieve data of specific team. including achieved score, badges, cards owned by members.	GET/api/team/{id}/
Team create	Create new team (name access code).	POST/api/team-create/
Team update	Update data of specific team.	PUT/PATCH/api/team-edit/{id}/
Team delete	Delete specific team.	DELETE/api/team-edit/{id}/
New membership	Add the current authenticated user to specific team.	POST/api/team-membership/

4.6. Visitor Activity Tracking

Visitors perform various types of actions while using the eguide application. The most obvious is the mere act of visiting a location, but more precise actions can be aimed at specific content elements (to be viewed or skipped) or gamification elements (e.g., puzzles solved or failed to solve). The eguide application reports events generated by defined user actions to the server with the necessary context information as defined in [23] (pp. 2043–2044) (note that the eMused.eu API follows the set of data proposed there, not the textual description format). The eMused.eu API does not expose the rules, which are processed server-side by the dedicated rule engine [24], which checks conditions depending on the event parameters and the current game state, possibly triggering relevant rules resulting in awarding points, badges or virtual items, or moving the user to hidden route items; such a result is sent back in the response to the API call.

The eMused.eu API endpoints handling visitor activity tracking are listed in Table 6.

Table 6. Visitor activity tracking endpoints.

Name	Description	Endpoint
Action list	List active actions for the area/location/item. Filter available.	GET/api/actions/
Action details	Retrieve specific active action data.	GET/api/actions/{id}/
Event registration	Update data of specific team. Returns achievements data if achieved by this event (points, items, badges).	POST/api/events/
Events list	List the current authenticated user's events with related achievements data (points, acquired items, badges).	GET/api/events/
Event details	Retrieve specific event data for the current authenticated user with related achievements data (points, acquired items, badges).	GET/api/events/{id}/

4.7. Internationalization

One of the key advantages of mobile eguides is their ability to deliver content in a language preferred by the visitor without the need to provide human guides that would be able to speak that language (the only requirement is to have the content translated).

The eMused.eu API endpoints provided to support eguide internationalization are listed in Table 7. Note that the endpoints listed there require no prior authorization.

Table 7. Internationalization endpoints.

Name	Description	Endpoint
Language list	List defined languages.	GET/api/language/
Language details	Retrieve specific language data.	GET/api/language/{id}/
Country list	List defined countries.	GET/api/country/
User interface translation	List available UI translations.	GET/api/ui_translation/
Language list	List defined languages.	GET/api/language/

5. Implementation

The eMused.eu API follows the REST paradigm: its endpoints are accessed using standard HTTP methods, such as GET and POST, targeting a specific URL. The data serialization format used was JSON.

The reference service conforming to the eMused.eu API [10] was implemented in Python 3. The eguide management panel was developed using the Django framework. The solution uses the PostgreSQL database management system and Nginx web server. The following external Python modules were used: bootstrap4, django-cors-headers, django-extensions, django-filters, django-qr-

code, django-crispy-forms, django-private-storage, django-rest-framework-social-oauth2, sorl-thumbnail.

Due to its size, ensuing from the complexity of the solution, it is difficult to present the actual implemented data model on a single figure; a simplified data model featuring the key elements of the object types referred to in the list of endpoints is shown in Figure 1.

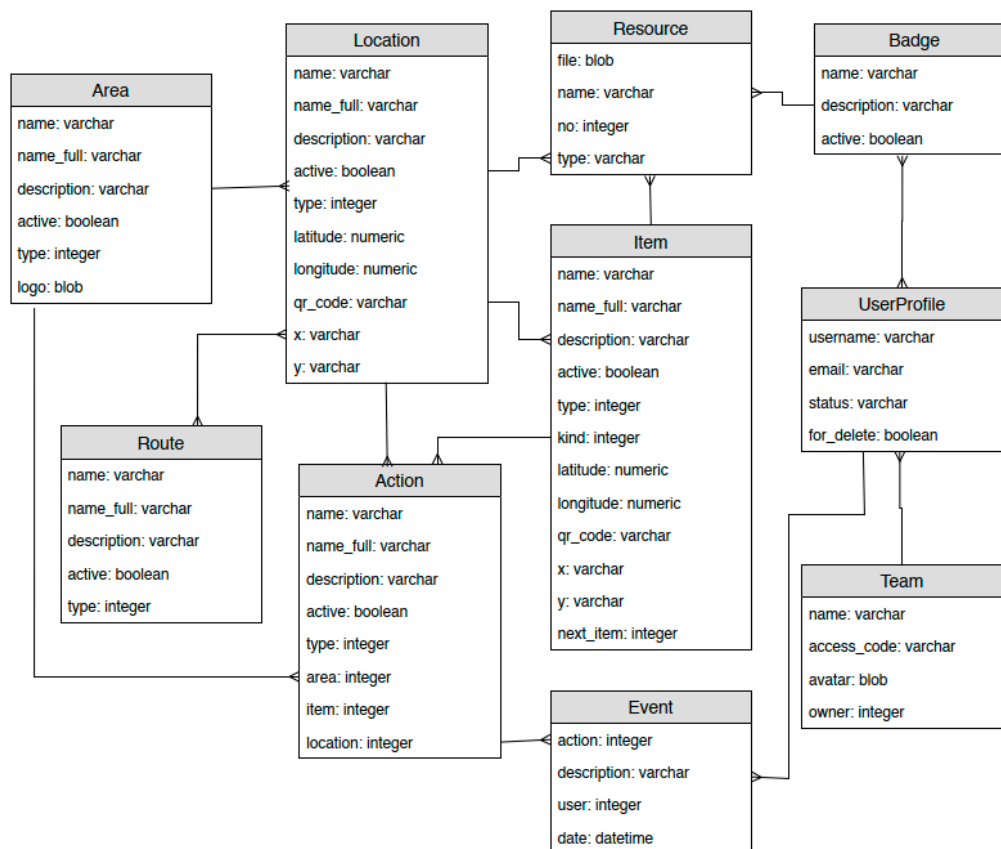


Figure 1. A simplification of the data model referenced by the API.

The reference service conforming to the eMused.eu API was programmed mostly in Python. Table 8 lists primary statistics for the source code implementing the functionality behind the API (including the eguide management platform user interface).

Table 8. Code statistics.

Technology	Files	Lines	Classes	Methods/Functions
Python	165	12,991	666	420
JavaScript	1	106	-	56
CSS	1	78	-	-
HTML	35	4213	-	-

6. Verification

6.1. Functionality

In order to check whether the eMused.eu API actually fills the gap between existing gamification-specific and eguide-specific APIs, its key functionality elements were compared to one example of the latter (izi.TRAVEL API [8]) and one example of the former (Gamification API [17]). As can be seen from Table 9, this goal was achieved, as the eMused.eu API covers the functionality of both the existing types of APIs and extends them with elements that were found in neither (the support for Grouping routes and Surveys).

Table 9. Functionality comparison between the presented API, an eguide API and a gamification API.

Functionality	The presented API	izi.TRAVEL API	Gamification API
Grouping routes (Area)	+	-	-
Separate content for a route	+	+	-
Defined locations	+	+	-
Controlled access to eGuide content	+	+	-
User profile and gamification achievements	+	-	+
User event log	+	-	+
Event-triggered rules	+	-	+
Quiz	+	-(+)	-
Survey	+	-	-

Legend: + supported; - unsupported; -(+) defined in the model but not supported by any endpoint.

6.2. Performance

In order to check whether using the API does not hinder the performance of the eguide apps depending on it, performance tests using a database containing a mix of test and real-world (obtained from the BalticMuseums: Love IT! project [9]) eguide data were performed on a reference implementation of a service conforming to the API. The obtained time measurements for respective endpoints show that the API can ensure smooth eguide operation even when hosted on a virtual machine with limited resources (see Table 10). Note that, though the measurements for two endpoints (/api/route/and/api/item/) are distinctly high, they do not mirror real-world usage scenarios, in which they will be used almost all the time with parameters specifying one item or route for which to retrieve data (rather than retrieving data for all items or routes, as was the case in this performance test).

Table 10. Statistics on the most aggravating endpoints (1000 requests per endpoint).

Endpoint	Mean [s.]	Std [%]	Median [s.]	Min [s.]	Max [s.]
/api/actions/	0.077	9.000	0.074	0.068	0.089
/api/area/	0.065	14.210	0.061	0.056	0.085
/api/area-type/	0.054	28.790	0.049	0.039	0.090
/api/badge/	0.081	19.340	0.078	0.062	0.111
/api/country/	0.048	18.990	0.045	0.040	0.067
/api/item/	1.139	7.170	1.100	1.052	1.304
/api/item-kind/	0.051	31.140	0.041	0.039	0.086
/api/item-type/	0.051	31.350	0.042	0.040	0.095
/api/language/	0.050	15.360	0.046	0.042	0.065
/api/locations/	0.313	3.930	0.310	0.298	0.331
/api/locations-type/	0.049	13.450	0.047	0.041	0.058
/api/quiz/	0.127	5.490	0.124	0.119	0.141
/api/route/	1.424	5.430	1.393	1.382	1.651
/api/route-type/	0.045	17.260	0.040	0.038	0.059
/api/site/	0.065	21.870	0.058	0.053	0.092
/api/survey/	0.049	13.800	0.045	0.043	0.061
/api/team/	0.056	19.500	0.058	0.043	0.076
/api/ui_translation/	0.057	19.790	0.051	0.047	0.084

Test environment: FreeBSD 11.2-RELEASE-p4 operating system, PostgreSQL 9.6.10 database management system, nginx 1.14.0 web server, Python 3.6.6 interpreter, Django 2.1.2 and Django Rest 3.9.1 frameworks, running on a machine featuring two Intel Xeon E5-2620 v4 2.10 GHz processors,

256 GB of RAM and 12 fast SSD drives within a virtualized environment limited to 2 GB of RAM, 45 GB of disk space and 45 simultaneously running processes.

7. Conclusions

The eMused.eu API was designed for the cloud-based management of gamified eguides to provide a ready-made solution for tourist attractions willing to provide their visitors with mobile eguides featuring gamification configured according to their specific needs. It covers seven main functional areas which are crucial for the operation of eguides: Visitor accounts, Tour organization, Tour and user data, Quizzes and surveys, Team management, Visitor Activity Tracking and Internationalization.

The developed API met its design goals. The coverage of the eMused.eu API includes the functionality areas of existing both gamification-specific and guide-specific APIs. The performance tests on a reference implementation of a service conforming to the API show that the delay caused by calls to the API is acceptable in practical conditions.

The eMused.eu API obviously has its limitations. While it was designed to satisfy the needs of tourist attractions of various types, and make it as universal as necessary to implement various gamification schemes, there are too many game design elements for a one-solution-fits-all to be possible. It will nonetheless be our future work to discover the functionality elements which are missing from the eMused.eu API in its current version, but are looked for by many developers of gamified eguides, and consequently extend the future releases of the API to include them.

In order to sustain the availability of the eMused.eu API in the cloud, the BalticMuseums: LoveIT! project partners decided to form a cooperation network [19], the responsibility of which it will be to maintain the hosting of the services conforming to the API and make decisions regarding future upgrades of the API. The network is open for other museums that wish to make use of the eMused.eu API, with the only expected commitment to be participation in the server upkeep costs.

Author Contributions: Conceptualization, J.S.; methodology, A.K.; software, A.K.; validation, A.K.; investigation, J.S.; resources, A.K.; data curation, A.K.; writing—original draft preparation, both authors; writing—review and editing, both authors; visualization, A.K.; supervision, J.S.; project administration, J.S.; funding acquisition, J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was part-financed from the European Regional Development Fund under the Interreg South Baltic Programme, grant number STHB.02.01.00-32-0055/16.

Acknowledgments: We would like to thank all the partners of the BalticMuseums: Love IT! project and subcontractors who knowingly or not contributed to the requirement elicitation and testing processes of the eMused.eu API.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Krosbacher, M. Authenticity and the Use of Multimedia at Cultural Tourist Attractions. Ph.D. Thesis, Technological University, Dublin, Ireland, 2010.
2. Smartphone Ownership is Growing Rapidly around the World, but Not Always Equally. Available online: <https://www.pewresearch.org/global/2019/02/05/smartphone-ownership-is-growing-rapidly-around-the-world-but-not-always-equally> (accessed on 2 February 2020).
3. Tussyadiah, I.P. Technology and Behavioral Design in Tourism. In *Design Science in Tourism*; Fesenmaier, D.R., Xiang, Z., Eds.; Springer: Cham, Switzerland, 2017; pp. 173–191.
4. Lee, B. The Effect of Gamification on Psychological and Behavioral Outcomes: Implications for Cruise Tourism Destinations. *Sustainability* **2019**, *11*, 3002.
5. Not, E.; Petrelli, D. Empowering cultural heritage professionals with tools for authoring and deploying personalised visitor experiences. *User Modeling User-Adapt. Interact.* **2019**, *29*, 67–120.
6. Swacha, J. Architecture of a dispersed gamification system for tourist attractions. *Information* **2019**, *10*, 33.

7. Google Play Game Services. Available online: <https://developers.google.com/games/services> (accessed on 4 April 2020).
8. izi.TRAVEL API. Available online: <https://izi.travel/en/api> (accessed on 4 April 2020).
9. BalticMuseums: Love IT! Available online: <http://bmloveit.usz.edu.pl> (accessed on 4 April 2020).
10. eMused.eu API. Available online: <https://cms.quistorp.emused.eu/docs/> (accessed on 4 April 2020).
11. TAP CMS Authoring Tools. Available online: <https://github.com/IMAmuseum/tap-cms/wiki> (accessed on 4 April 2020).
12. Stein, R.; Proctor, N. TourML: An Emerging Specification for Museum Mobile Experiences. In *Museums and the Web 2011*; Archives & Museum Informatics: Philadelphia, PA, USA, 2011.
13. Guidepost. Available online: <https://gitlab.com/americanart/guidepost> (accessed on 4 April 2020).
14. Audio Guide Usage: iPhone Versus Android in 2018. Available online: <https://izi.travel/en/blog/audio-guide-usage-iphone-versus-android-in-2018> (accessed on 4 April 2020).
15. Paiva, J.; Leal, J.; Queirós, R. Gamification of learning activities with the Odin service. *Comput. Sci. Inf. Syst.* **2016**, *13*, 809–826.
16. Gamification Server. Available online: <https://github.com/ngageoint/gamification-server> (accessed on 4 April 2020).
17. Phanzu, K.; Thévoz, A. Gamification API. Available online: <https://aurelienthevoz.docs.apiary.io> (accessed on 4 April 2020).
18. Swacha, J.; Muszyńska, K. Towards a Generic eGuide Gamification Framework for Tourist Attractions. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, Melbourne, Australia, 28–31 October 2018; ACM: Melbourne, Australia, 2018; pp. 619–625.
19. eMused.eu. Available online: <https://emused.eu/> (accessed on 4 April 2020).
20. The OAuth 2.0 Authorization Framework (RFC 6749). Available online: <https://tools.ietf.org/html/rfc6749> (accessed on 4 April 2020).
21. Aleksic-Maslac, K.; Sinkovic, B.; Vranesic, P. Influence of gamification on student engagement in education. *Int. J. Educ. Learn. Syst.* **2017**, *2*, 76–82.
22. Nicholson, S. A RECIPE for Meaningful Gamification. In *Gamification in Education and Business*; Reiners, T., Wood, L.C., Eds.; Springer: Cham, Switzerland, 2015; pp. 1–20.
23. Swacha, J. Representation of Events and Rules in Gamification Systems. *Procedia Comput. Sci.* **2018**, *126*, 2040–2049.
24. Kulpa, A.; Swacha, J. Design and Implementation of Rule Execution Mechanism for an eGuide Gamification Web Service. In *Information and Software Technologies*; Damaševičius, R., Vasiljevičienė, G., Eds.; Springer: Cham, Switzerland, 2019; pp. 173–181.

