Information Technology & Tourism (2019) 21:577–593
https://doi.org/10.1007/s40558-019-00159-w

CASE STUDY

Check for updates

# Open Tourist Information System: a platform for touristic information management and outreach

**Pedro Lopes[1] · Luís Almeida[1] · João Pinto[1] · Justino de Jesus[1] · Didiana Fernandes[1] · Isabel Vieira[1] · Ricardo Gama[1]**

Received: 30 November 2018 / Revised: 11 June 2019 / Accepted: 11 November 2019 /
Published online: 18 November 2019
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

**Abstract**
In this paper, we present a generic, open source, touristic information system. The main objective of this platform is to promote the development of third-party applications based on a previously structured data and validated centralized common source of open information. The proposed system comprises an information management module and an extended set of web services enabling efficient information management as well as its use by third-parties such as web and mobile applications. It was designed with a modular functional architecture with the aim of being easy to implement and adapt in practical environments, as well as being user-friendly and having capability to grow. The entire system structure is discussed in detail. We test the use of the information system in a practical scenario, implementing it for the Douro World Heritage Region, Portugal. The web services developed are profiled in this real word setting through a wide set of performance tests. The results show that the end-to-end execution times of the web services functions make them suitable for standard tourism applications. These results establish concrete performance benchmarks for future applications.

**Keywords** Information system · Information outreach · Tourism · Route planner

## 1 Introduction

Tourism is one of the most important drivers of socio-economic development in many regions of the world. Over the last years, we have witnessed increased digitalization in tourism and its value chains, as the internet becomes the main source of information for tourists and tourism services providers.

✉ Ricardo Gama
rgama@estgl.ipv.pt

[1] School of Technology and Management of Lamego, Polytechnic Institute of Viseu, Av. Visconde Guedes Teixeira, 5100-074 Lamego, Portugal

المنارة للاستشارات

Consequently, there has been a growth in the number of web/mobile tourism applications. The functionalities of these applications are very diverse, ranging from mere tourist information providers to points of interest (POIs) recommendation, tourist route planning help or automatic route suggestion.

For example, applications like "City Trip Planner" (Vansteenwegen et al. 2011), "eCOMPASS" (Gavalas et al. 2015), integrate algorithms with a mobile and web-based construction, capable of considering many user preferences and creating a recommended route on the fly. For a more detailed review of the field, see the surveys (Borràs et al. 2014; Thiengburanathum et al. 2016).

More recent applications (Gavalas et al. 2017; Lim et al. 2018; Silva et al. 2018), have started incorporating recent advances in technologies such as artificial intelligence, big data or mobile sensors, to make smarter user-oriented recommendations. Despite being very diverse, all of these applications commonly require high-quality information about the region where they operate. Validated open information is not always available or is often dispersed throughout various public sources, making fast access and integration in applications difficult.

To overcome these shortcomings, Pereira et al. (2015) suggested an architecture and implementing a common Application Programming Interface (API) to provide access to structured information about POIs, events and itineraries. This API was made available for five European cities and is also used by several private mobile applications companies. More recently, Ribeiro et al. (2018), provided a broad overview of mobile applications for accessible tourism. The authors also identified the involvement of tourism stakeholders, the integration of disperse information in a common database and accessible tourism support frameworks as the main factors to enhance the quality and development of new web/mobile applications.

In fact, with the increase in the use of smart tourism technologies such as travel-related websites and smartphone applications for travel planning and assistance, establishing information systems is becoming increasingly important. This allows tourism data to be centralized and accessible (Arenas et al. 2019) to leverage better interactions between different tourism stakeholders and players. The use of open data (Longhi et al. 2014; Rajapaksha et al. 2017), and particularly open-linked data (Fogli and Sansonetti 2019), can have a deep impact mainly in regions where there is a wide dispersal and often the absence of quality information.

This paradigm shift from traditional approaches to smart tourism destination methodologies (Jovicic 2017; Li et al. 2017) leads to profound changes in the way destinations should communicate with tourists who increasingly seek quick access to relevant content and high-quality information about the places they visit.

In this work, we present a generic open information system that enables efficient touristic information management and outreach. A detailed analysis of a system, as well as its implementation and benchmarking its performance in a real-world setting, are the main contributions of this work. The system proposed consists of an administrative information management module and a set of web services provided through an API. In order to enhance the use of the platform and the information contained therein, in addition to standard information retrieval functionalities, the API includes a broader set of specific tools for tourist data-driven applications. Thus, with a focus on real-world practical applications, the toolkit developed, allows all of



المنارة للاستشارات

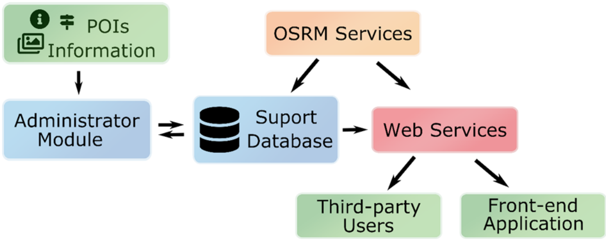


**Fig. 1** The overall structure of our information system

the information and its functionalities to be integrated by web or mobile third-party applications, enriching their potential and usability.

Overall, the information system is designed with easy setup and use in mind. It can be implemented and customized for any touristic region. Furthermore, this platform provides a common regional strategy that enables simple access to and use of information. It also allows previously collected and validated information from public and private tourist agents of the region to be disseminated. The entire system is based on publicly available tourist information and is solely dependent on open geographic information and routing services. As a case study, we have set up our proposed system in the Douro Valley Region, Portugal, a World Heritage Site, making a web platform, which gathers comprehensive, easy to access, touristic information available for third-party agents.

The remainder of this article is structured as follows: in Sect. 2 we present the proposed touristic information system, detailing its architecture and functionalities. In Sect. 3 we present the concrete implementation details of our proposal in a real word setting. In Sect. 4 we report performance test results of the API functionalities. In Sect. 5 we summarise our results and provide possible future development paths.

## 2 Tourism information system architecture

In this section, we describe the architecture of the proposed system. Specifically, in Sect. 2.1 we describe the overall system design, presenting its main functional blocks. In Sect. 2.2 we present the database structure. The external open geographic information and routing services are characterized in Sect. 2.3, followed by the description of the information management component in Sect. 2.4. API implementation and functionalities are detailed in Sect. 2.5.

### 2.1 Overall system architecture

The overall structure of the information system is composed of several interconnected functional blocks, as seen in Fig. 1. These are contained in a database that stores all relevant information in the context of tourism and its related activities. To enable efficient information management, including by non-technical staff, an administration module is provided. The overall systems design was inspired by the

المنارة للاستشارات

best practices identified in former studies and experiments (Pereira et al. 2015; Arenas et al. 2019). In particular, its structure strive to fulfil the data and functionalities requirements needed in modern applications described in the literature, as well as facilitate and enhance its use by data providers and tourism agents.

Proprietary mapping and routing services usually impose usage limits and other constraints. For these reasons, we resort to alternative publicly available and open services. Finally, a diverse group of web services is provided to third-party users through an API, allowing them to freely access all the previously structured information validated and stored in the database. A simple front-end web application was also created to provide information access to individual users.

The server-side application was developed using the Flask framework. Flask is a micro web framework written in Python that does not require particular tools or libraries ("Flask" n.d.) making it suitable to create easily adaptable and generalized applications. The entire service is then relayed using Apache HTTP Server ("APACHE" n.d.), a free, open-source cross-platform web server.

In the next section, we will detail the various system components, emphasizing their relation and overall importance.

### 2.2 Database structure and implementation

The database structure has the main POI information table at its core, containing the characterization and description features. Concretely, the metadata stored for each POI include the following: id, name, geo-coordinates (latitude and longitude), 0–10 scale POI relevance scores, description, category, sub-region and information source. Focusing on the database query optimization and easier information management, all other relevant information was organized in different tables, linked to the central main POI table through the POI index.

Time related features, like opening/closing times for each weekday, average visit duration, or any other relevant information, is stored in a particular table. Also, in order to speedup application functionalities, we maintain a table with precomputed travel time and distances between every stored POI using different transport modes, see the database architecture diagram in Fig. 2. To guarantee consistency, we adopt metres and minutes for distance and time unit fields respectively.

Another important database component is related to the route data. Since tourist routes can be modelled by POI sequences, the "pois_sequences" table acts as a connecting element between POI and route information tables. Explicitly, every route can be decomposed in smaller paths between POI pairs and its information is kept in this table. Every route segment is characterized by segment id, that is it's relative position in the respective route, route id, starting POI, ending POI and description. Regarding the route itself, the metadata include name, category, description, score, travel mode, time duration and length; the typology indication is stored as well. Finally, all multimedia content, POI or route related, is stored in a separate table. The metadata comprises the image id, name and a short description and id and element type (POI, Route or Route segment) related to the content.



المنارة للاستشارات

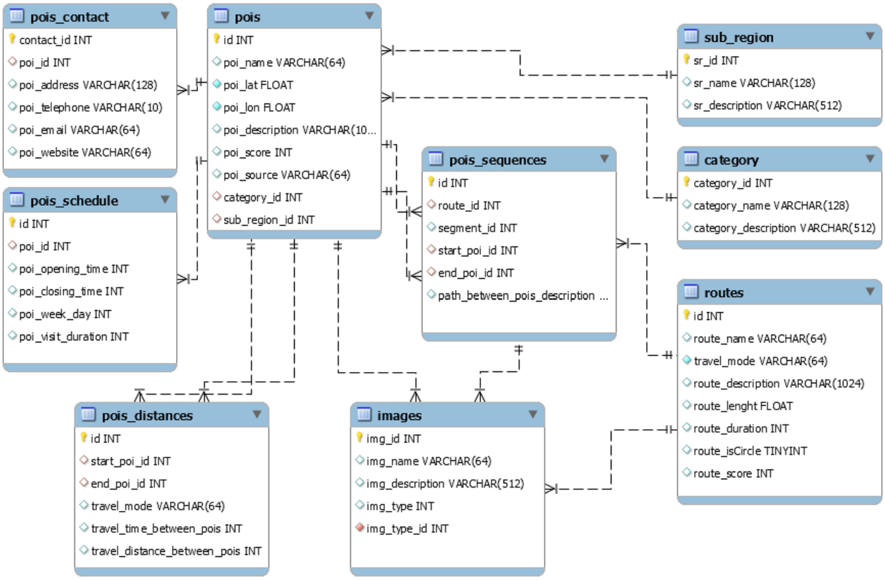


**Fig. 2** Database diagram illustrating the relations among the different tables

Although they are not presented in Fig. 2 for the sake of simplicity, there are several other descriptive and control fields in the database tables. For example, creation and update dates, remarks and notes fields, as well Boolean control variables, to control the need to revise or conceal particular field, were included for a better information management and maintenance. All database records can be customizable to support different contexts according to administrator and applicability requirements.

In order for the information system to be more sweeping, all application database queries are performed through the Object Relation Mapping framework SQLAlchemy ("SQLAlchemy" n.d.). This option guarantees the creation of a database-agnostic code and enables the possibility of adopting of a wide variety of database engines.

## 2.3 External open geographic information and routing services

In order to maintain independence from service providers, we rely only on open geographic information and routing services. Explicitly, we use data provided by Open Street Maps (OSM), more specifically road network and geographic information data. At this stage, it is important to ensure the high quality of any system's information; this is especially true for open systems. Analysis of the information quality of several systems has been published using Volunteered Geographic Information (VGI) at a European level (Cipeluch et al. 2010; Girres and Touya 2010; Wang et al. 2013; Zielstra and Zipf 2010) and in China (Haklay 2010). Here, the advantage of OSM in terms of

المنارة للاستشارات

responsiveness and flexibility has been shown. Travel times, distances and routing are calculated using the Open Source Routing Machine software (Luxen and Vetter 2011).

#### 2.3.1 Open Street Maps data

Open Street Maps (OSM) is one of the most popular projects for Volunteered Geographic Information (VGI) on the internet (Mooney and Corcoran 2014). Volunteered Geographic Information, is the collection of spatial data captured by "citizen sensors," where this data is then edited and managed within a collaborative web environment (Goodchild 2009). OSM is constantly changing and updating and contains spatial data and attribution, representing almost every conceivable geographical feature. The successful progress of OSM since its emergence in 2004 is very often attributed to its status as a "crowdsourced" VGI project. Taking advantage of collaborative work, there are studies that address the quality of information that exists when using the VGI service (Ali et al. 2016).

#### 2.3.2 Open Source Routing Machine Server

We set up an Open Source Routing Machine server (Luxen and Vetter 2011). This server handles all routing requests for our application, outputting route polyline representations as well as travel times and distances associated.

The Open Source Routing Machine (OSRM) is a high-performance open-source C++ routing engine designed for use with data from the Open Street Map project and indicates the shortest routes on public roads ("Open Source Routing Machine" n.d.). This service is hosted across three different ports where each port serving a specific travelling profile route. One port for driving, another for cycling and the last for walking, allowing us to choose which profile to execute our query on, altering the results due to different travel speeds and routes taken.

### 2.4 Administrator module

The administrative module is the application management interface that allows database entries to be created, edited, updated or eliminated. Its role is of paramount importance as it ensures the proper operation of our information system. In this sense, special attention was paid to the usability interface. More specifically, by relying on a tabular layout or on a map visualization tool it is possible for users, with authorized access, to manage all of the information contained in the database.

Furthermore, we developed a set of tools to allow the interconnection between the OSRM server functions and the database. Using these, all distances and travel times can be dynamically updated upon database changes. Updates are performed asynchronously, as a background process, thus preventing added latency.



المنارة للاستشارات

### 2.5 Application programming interface

In order to allow stored information to be accessed easily, an Application Programming Interface (API) module was developed. In fact, since the platform has already integrated an OSRM server in parallel with the all-purpose information retrieval functions, specific routing and itinerary recommendation tools have also been created. Concretely, the API functions can be divided into three different functional groups: POI selection, routing, and route recommendations. This set of implemented functions intended to offer not only tools that allow stored information to be integrated by web/mobile third-party applications, but also existing local resources to be optimized, to empower embedded problem oriented solutions easily.

From the practical implementation point of view, the API was built using the Representational State Transfer, or REST, architectural design (Richards 2006). All API calls can be performed using the HTTP protocol and results are returned in JSON format.

The first functional group allows the retrieval of a list of POIs that satisfies several imposed time or distance constraints. Concretely we have implemented the functions described in Table 1.

All F1 to F5 functions also have the number of retrieved POIs as input arguments as well as the sorting options, allowing the results to be presented by importance, time or distance according to user criteria.

In order to extend the range of API functionalities, the group of functions in Table 2 is able to use the existing OSRM services available on our server, thus enabling the use of its native routing features.

Finally, the capability of performing personalized tourist route recommendations was included in our system. The challenge for tourist route recommendations, subject to several preference and time constraints, lies in being computationally difficult. Heuristics that generate good quality solutions with small enough computation times allowing them to be used in real-time Web/Mobile applications have only recently been proposed (Gavalas et al. 2014). In the API, we use an implementation of the heuristic suggested in (Vansteenwegen et al. 2009), which was already tested in a practical setting (Vansteenwegen et al. 2011). Specifically, we have created two functions described in Table 3.

Regarding API security, in the literature, there are several approaches and different implementation perspectives (Karnwal et al. 2012; Wang and Li 2010). We adopt register usage logs by key and client IP address along with the request that was made. On the server side, all requests have maximum time–frequency access to prevent DDOS attacks. This approach allows us to control connections that exceed established limits and to quarantine them for future analysis. The quarantine management process can be specified to be automatically processed or manually performed using the administration module.

المنارة للاستشارات

**Table 1** List of APIs first functional group with the indication of their input arguments and outputs

| Name | Inputs | Outputs |
|---|---|---|
| G1F1 | Geographical point coordinates, maximum allowed distance<br>Optional: category, sub-region, sorting, number of POIs | List of POIs within a maximum geographical (Haversine) distance from a user-defined geographical point |
| G1F2 | Geographical point coordinates, maximum allowed travel time<br>Optional: category, sub-region sorting, number of POIs | Returns all POIs stored information, whose travel time between them and the geographic location defined by the user is less than a given time |
| G1F3 | POI Id, maximum allowed distance<br>Optional: category, sub-region sorting, number of POIs | List of POIs within a maximum geographical (Haversine) distance, from a user-defined existing POI |
| G1F4 | POI Id, maximum allowed travel time<br>Optional: category, sub-region sorting, number of POIs | Returns all POIs whose travel time between them and user-defined POI Id is less than an allowed given time |
| G1F5 | At least one of these parameters has to be given<br>Optional: category, sub-region sorting, number of POIs | Returns all POI belonging to a category and/or subregion |
| G1F6 | POI Id | POI database stored information |

## 3 Implementation on a world heritage region

The Douro Region in Portugal has been included in the UNESCO list of World Heritage Sites since 2001. It has a geographic area of 4032 $km^2$ and consists of 19 administrative sub-regions. It is known worldwide for its landscape and enotourism related activities, and it is visited by thousands of domestic and international tourists annually (Rebelo et al. 2015).

Although some tourist related web and mobile applications are available, see for example ("Douro Valley" n.d.; "Douro Alliance" n.d.), they present very limited functionalities being generally limited to a searchable list of POI information and only covering a particular touristic sub-region.

With the aim of testing the proposed tourist information system and evaluating its applicability in a practical scenario, we set it up for the Douro Region and deployed it. We also strove to enable the development of new or existing applications, making it possible to include new practical features, such as smart POI and route recommendation, among others.

Considering the particular characteristics of this touristic region, ten categories were defined, see Table 4 for details. All other database features were maintained as previously described.

To guarantee the quality of the touristic information stored, all of the data was collected from the various regional public entities and other open data sources whose quality could be confirmed. Information regarding more than 700 POIs, see Fig. 3, was stored in the database, comprising a total of approximately 120 MB of stored information. Within the defined categories, the collected sample is representative of the



المنارة للاستشارات

**Table 2** List of APIs second functional group with the indication of their input arguments and outputs

| Name | Inputs | Outputs |
|---|---|---|
| G2F1 | Starting POI Id; Ending POI Id<br>Profile: {Car, Walking} | Travel time and distance; Route Polyline encoding |
| G2F2 | Starting POI Id; Ending Geographical point coordinates<br>Profile: {Car, Walking} | Travel time and distance; Route Polyline encoding |
| G2F3 | Starting Geographical point coordinates; Ending POI Id<br>Profile: {Car, Walking} | Travel time and distance; Route Polyline encoding |

**Table 3** List of APIs third functional group with the indication of their input arguments and outputs

| Name | Inputs | Outputs |
|---|---|---|
| G3F1 | Starting/ending POI Id, number of days, Time budget, Starting time<br>Optional: category and/or sub-region constraints | Returns a sequential list of POIs and their information data, representing the optimized recommended travel route |
| G3F2 | Starting/ending geographical point coordinates, number of days, Time budget, Starting time;<br>Optional: category and/or sub-region constraints | Returns a sequential list of POIs and their information data, representing the optimized recommended travel route |

universe of POIs in the Douro Region, comprising the most important accessible touristic information. The data is available upon request.

The tourist information system, as well as the OSRM servers, are running on a cloud computing platform (Lopes and Pereira 2017). The Virtual Machine (VM) that supports the implementation has 12 GB of RAM with 6 cores at 1.7 GHz (Intel Xeon CPU E5-2603 v4).

## 4 Performance tests

As concluded in recent studies (Yang et al. 2018; Chen et al. 2016; Yan et al. 2017), real-world web applications often present database driven performance issues caused by latencies and delays in their services. One of the main factors for an application's success is its loading time (Bermbach and Wittern 2016; Yang et al. 2018). Consequently, proper system performance evaluation is of critical importance to ensure the system's usability in practical contexts. To this end, we conducted a series of performance evaluation tests.

### 4.1 API requests simulation

Our simulation tests were designed to mimic a real-world practical setting, in order to obtain realistic performance estimates. For every function, we made 4096 calls using 4 asynchronous threads with randomly generated inputs.



المنارة للاستشارات

**Table 4** List of categories and their descriptions in our real world implementation

| Categories | Description |
|---|---|
| Museums | Local museums or historical displays |
| Parks and gardens | Nature related places |
| Monuments and points of interest | Points of tourist and cultural interest |
| Theatres | Places with live performances and other entertainment activities |
| Vineyards and wine shops | Places related to wine production and sales |
| Viewpoints and landscapes | Sightseeing places |
| Popular festivals | Periodic local popular and religious festivities |
| SPAS | Spas and relaxation facilities in the region |
| Intangible heritage | Specific cultural elements, such as gastronomy and crafts |
| Historic centres | Historical city centres |

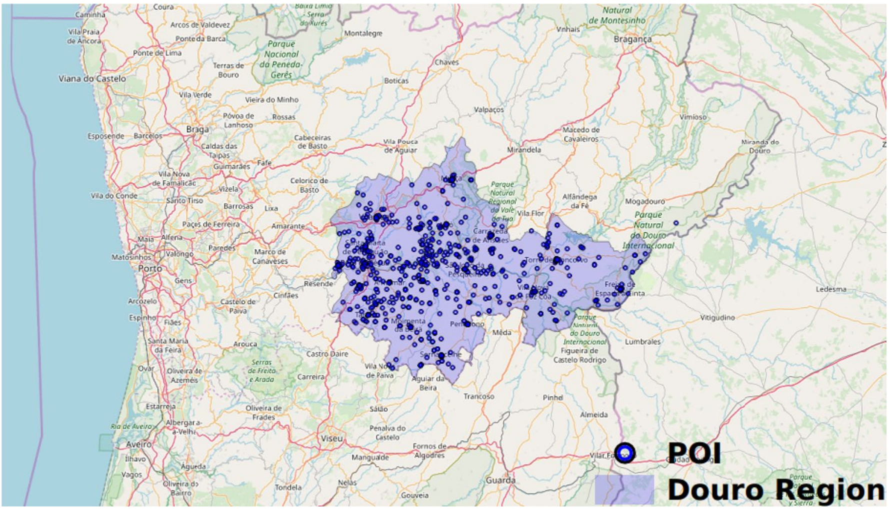


**Fig. 3** Geographic distribution of the POIs. The shaded polygon identifies the Douro Region, Portugal

Specifically, the time arguments were sampled from a uniform distribution over a [60; 2700] second interval and distances were sampled from a uniform distribution over a [1; 20] kilometer interval. Every POI id, category and sub-region ids were sampled uniformly from the available database ids and all of the geographical point coordinates were generated by uniformly sampling latitude/longitude coordinates inside the geographical area of Douro Region. All HTTP calls were made on an AWS m4.xlarge instance (“AWS” n.d.).

### 4.2 Statistical analysis

Median statistics (with 95% confidence intervals) [95% CI] were computed for all of the measurements. Bootstrapping was used to obtain 95% confidence intervals



المنارة للاستشارات

for the mean and median. Specifically, we resampled (across http calls) 999 times with replacement to obtain a distribution over the performance metric of interest. The two-sided 95% confidence interval bounds were then computed as the interval between 2.5 and 97.5 percentiles of this empirical distribution.

### 4.3 Results

Although some tourism information systems have been proposed in the literature, to the author's best knowledge, there's a lack of studies with performance assessments in this particular area. On the other hand, there are some recent studies about of web/mobile applications usability, with a particular focus on the desirable execution time requirements (Bermbach and Wittern 2016; Yang et al. 2018). Therefore, we focus our analysis on three main variables: end-to-end time, the percentage of server time, and size of the returned request in Kbytes.

#### 4.3.1 POI selection functions

The first group of functions allows database queries to be performed under several practical constraints. In terms of performance, we can divide them into two sub-groups, as we can see in Fig. 4. For these functions we have obtained median end-to-end times, see Table 5.

As we can see, the measured execution times allow these functions to be included in real-time applications. In general, all functions presented narrow time distributions with small variance, see Fig. 4.

As expected, the overall end-to-end time is strictly related to the amount of information retrieved. This is most notable on the performance of G1F1, G1F2, G1F3 and G1F4 functions, as we can see in Fig. 5.

In turn, the amount of information returned in the query is also dependent on the amount of information stored for each POI, and therefore dependent on the particular implementation, tourist region or application to be given to the system. In this sense, end-to-end execution times can be shortened by restricting the returned information fields or the number of POI's.

In order to analyse the API's scalability over time, and to infer its behaviour with the growth in the number of POIs, information about them, or the number of potential calls, we carried out performance tests for the G1F1 to G1F4 functions on the sampled versions of our database and with a higher number of asynchronous calls. Specifically, we have tested two new versions of the original database created by randomly generating four and eight times more POI's and sampling all of the other features at random, thereby preserving the distribution of the amount of information across POI's. We made 4096 calls, using 4, 8 and 16 asynchronous threads, to simulate the increase in API access traffic. To ensure that the simulations can be comparable, we imposed a maximum of 10 POI's on the returned functions response.

المنارة للاستشارات

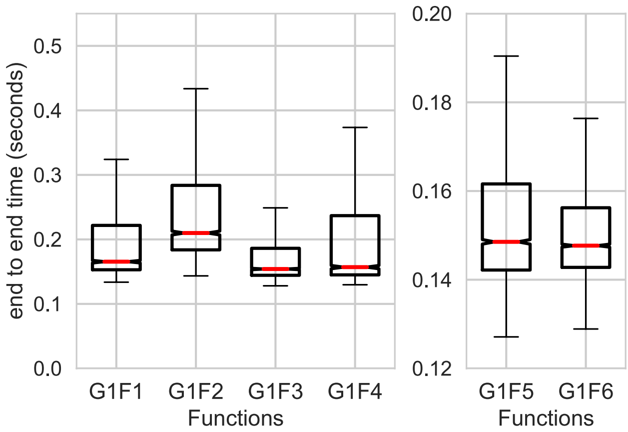


**Fig. 4** Overall end-to-end execution times of the first group of functions

**Table 5** Measured end-to-end times, in seconds, for the first functional group

| Function | Time (s) |
|---|---|
| G1F1 | 0.165 (0.164, 0.166) |
| G1F2 | 0.210 (0.207, 0.212) |
| G1F3 | 0.154 (0.153, 0.154) |
| G1F4 | 0.157 (0.155, 0.158) |
| G1F5 | 0.149 (0.148, 0.149) |
| G1F6 | 0.148 (0.147, 0.148) |

As the results show, see Fig. 6, the overall reduction in performance registered with the increase in the size of the database, 52% average increase on the median end-to-end time for the four functions, does not limit its use in regions with up to eight times more POIs and information. On the other hand, an increase in traffic can have a higher impact on the API performance, even on a system with a database on the same order of magnitude as ours. Nevertheless, this situation can always be circumvented making more hardware resources available to the system, and thus, enabling better response times.

#### 4.3.2 Routing functions

This group's performance is strictly related to the OSRM routing functions. Their performance is very stable, with median times of 0.148 (0.148, 0.148), 0.149 (0.149, 0.150) and 0.149 (0.149, 0.150) seconds respectively, see Fig. 7).



المنارة للاستشارات

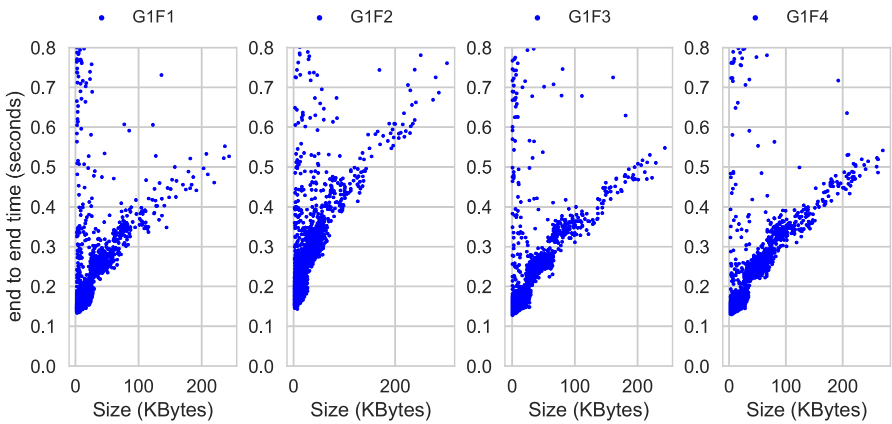


**Fig. 5** Scatter plots of the total size of the information retrieved versus the end-to-end execution time

The order of magnitude of overall end-to-end execution time, allows these groups of functions to be integrated with other more general and compound functionalities.

### 4.3.3 Tourist route recommendation functions

Finally, for the last group of functions, the values of the recorded times are substantially higher than those of the previous functions. Specifically, the route construction functions presented end-to-end times of 1.391 (1.334, 1.450) and 1.400 (1.346, 1.472) seconds respectively, see Fig. 8.

As these functions rely on an optimization heuristic, most of their end-to-end time comes from server processing, with more than 50% of the requests taking more than 76.4% (75.2, 77.5) and 80.9% (80.0,81.8) of the overall processing times for functions 1 and 2 respectively, see in Fig. 9.

In our practical implementation, we chose not to impose any restriction on the number of POI's available for the algorithmic route construction for this group of functions, in addition to the restrictions that follow the input parameters described above. Since the execution times for this group of functions strongly depends on the number of existing POI's, this allows us to benchmark the performance for the worst-case scenario. In a practical setting, the problem has more restrictive constraints such as imposing limits on POI scores and categories, number of allowed visits or sub-region preferences.

Analysing the end-to-end execution times, we can conclude that the API performance allows its functions to be included in real-time, stand-alone third-party applications. Moreover, for group 2, the end-to-end execution times are fast enough to allow its integration to be part of more general functionalities.

المنارة للاستشارات

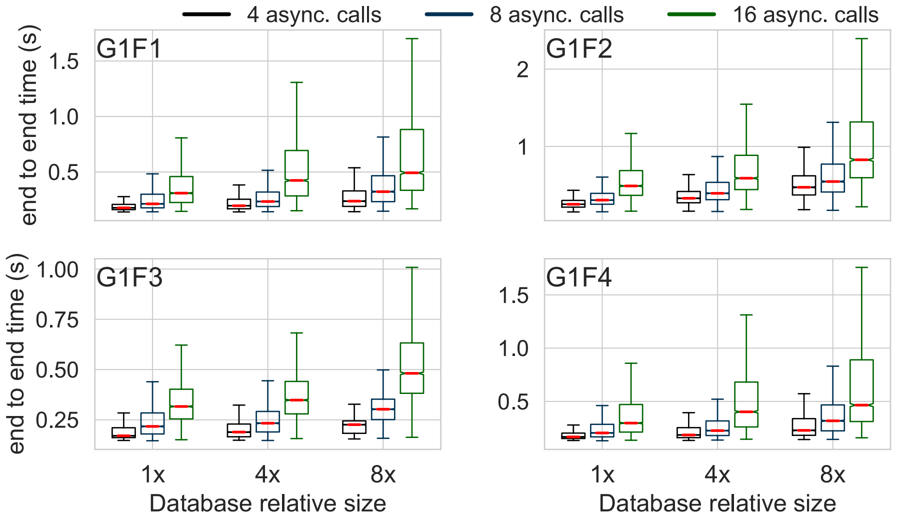


**Fig. 6** Box plots with overall end-to-end execution times of the G1F1 to G1F4 functions on the original database and on four and eight times larger versions of it

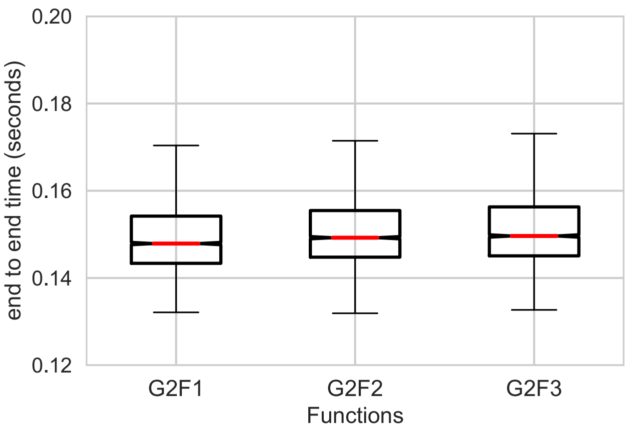


**Fig. 7** Overall end-to-end execution times of the routing functions

## 5 Conclusions

In this paper, we presented an open source Touristic Information System. Its architecture is modular allowing it to be developed and adapted to any practical setting easily. It is composed of two main components that allow the interconnection with the database, specially designed to contain touristic information relevant to modern tourist applications.



المنارة للاستشارات

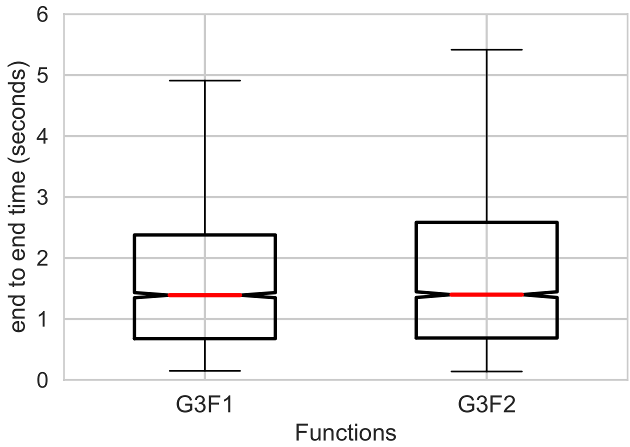


**Fig. 8** Overall end-to-end execution times of the tourist route recommendation functions

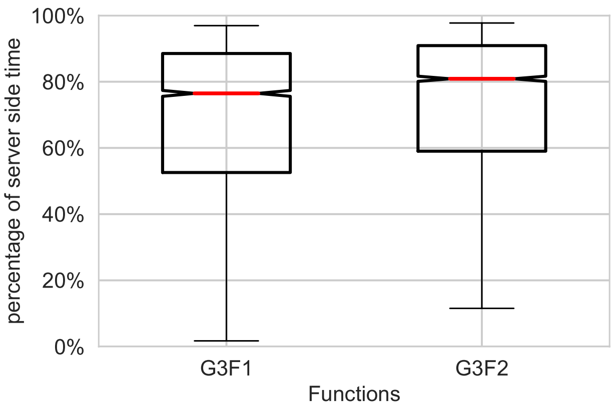


**Fig. 9** Percentage of server time among end-to-end time for the tourist route recommendation functions

To the best of our knowledge, the proposed Open Tourist Information System is the only available open source application that provides an easy-to-use administration module and an API, making it suitable for easy customization and deployment for small local regional contexts or even for larger tourist regions.

Specifically, Open Touristic Information Systems, such as the one proposed here can, not only add value to the regional tourism, but also serve as a platform of cooperation between tourist agents of the particular tourist destination. Given the importance cooperation has for support of a tourist destination over time, the existence of a platform that serves as the basis for uniting agents and tourism resources will serve as a tool for creating integrated products. This model can leverage the elaboration of carefully produced content, centred on the values of the territory and the needs of those who visit it. So, centralizing and enabling the dissemination of quality open tourist information encourages and facilitates the development of new



المنارة للاستشارات

featured applications adding important value to the tourism and their positioning in the touristic market.

We have conducted a series of benchmark performance tests on real-world implementation of the system proposed and concluded that all API functions end-to-end execution times are suitable for its integration in third-party real-time applications, reinforcing the practical usefulness of our proposal. Moreover, scalability tests showed that its usefulness and applicability remain viable in larger tourist regions. These results establish concrete performance benchmarks for future applications.

The source code for the proposed Tourism Information System, as well as detailed deployment instructions, can be found in GitHub.[1]

**Acknowledgements** The authors are deeply thankful to Nuno André and the anonymous reviewers for their helpful and constructive comments. This work was financed by the Instituto Politécnico de Viseu and Caixa Geral de Depósitos under the PROJCI&DETSCGD0017 project. Furthermore, we would like to thank the CI&DETS for their support.

## References

Ali A, Sirilertworakul N, Zipf A, Mobasheri A (2016) Guided classification system for conceptual overlapping classes in OpenStreetMap. ISPRS Int J Geo-Inf 5(6):87

APACHE (n.d.) https://httpd.apache.org/. Retrieved 14 March 2019

Arenas AE, Goh JM, Urueña A (2019) How does IT affect design centricity approaches: evidence from Spain's smart tourism ecosystem. Int J Inf Manag 45:149–162

AWS (n.d.) https://aws.amazon.com/. Retrieved 14 March 2019

Bermbach D, Wittern E (2016) Benchmarking web API quality. Int Conf Web Eng 2722:277–284

Borràs J, Moreno A, Valls A (2014) Intelligent tourism recommender systems: a survey. Expert Syst Appl 41(16):7370–7389

Chen TH, Shang W, Jiang ZM, Hassan AE, Nasser M, Flora P (2016) Finding and evaluating the performance impact of redundant data access for applications that are developed using object-relational mapping frameworks. IEEE Trans Softw Eng 42(12):1148–1161

Cipeluch B, Jacob R, Winstanley A, Mooney P, Ciepluch B, Jacob R, Mooney P (2010) Comparison of the accuracy of OpenStreetMap for Ireland with Google Maps and Bing Maps. In: Ninth international symposium on spatial accuracy assessment in natural resources and environmental sciences

Douro Alliance (n.d.) http://www.douroalliance.org. Retrieved 14 March 2019

Douro Valley (n.d.) http://www.dourovalley.eu/. Retrieved 14 March 2019

Flask (n.d.) http://flask.pocoo.org/. Retrieved 14 March 2019

Fogli A, Sansonetti G (2019) Exploiting semantics for context-aware itinerary recommendation. Pers Ubiquit Comput. https://doi.org/10.1007/s00779-018-01189-7

Gavalas D, Konstantopoulos C, Mastakas K, Pantziou G (2014) A survey on algorithmic approaches for solving tourist trip design problems. J Heuristics 20(3):291–328

Gavalas D, Kasapakis V, Konstantopoulos C, Pantziou G, Vathis N, Zaroliagis C (2015) The eCOMPASS multimodal tourist tour planner. Expert Syst Appl 42(21):7303–7316

Gavalas D, Kasapakis V, Konstantopoulos C, Pantziou G, Vathis N (2017) Scenic route planning for tourists. Pers Ubiquit Comput 21(1):137–155

Girres JF, Touya G (2010) Quality assessment of the French OpenStreetMap dataset. Trans GIS 14(4):435–459

Goodchild M (2009) NeoGeography and the nature of geographic expertise. J Locat Based Serv 3(2):82–96

[1] https://github.com/SiitESTGL/SIIT.



المنارة للاستشارات

Haklay M (2010) How good is volunteered geographical information? A comparative study of OpenStreetMap and ordnance survey datasets. Environ Plan B Plan Design 37(4):682–703

Jovicic DZ (2017) Current issues in tourism from the traditional understanding of tourism destination to the smart tourism destination, 3500

Karnwal T, Sivakumar T, Aghila G (2012) A comber approach to protect cloud computing against XML DDoS and HTTP DDoS attack. In: 2012 IEEE students' conference on electrical, electronics and computer science, pp 1–5

Li Y, Hu C, Huang C, Duan L (2017) The concept of smart tourism in the context of tourism information services. Tour Manag 58:293–300

Lim KH, Chan J, Leckie C, Karunasekera S (2018) Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency. Knowl Inf Syst 54(2):375–406

Longhi C, Titz J-B, Viallis L (2014) Open data: challenges and opportunities for the tourism industry. In: Tourism management, marketing, and development. Palgrave Macmillan, New York, pp 57–76

Lopes P, Pereira F (2017) Proposal for a federation of hybrid clouds infrastructure in higher education institutions. In: Rocha Á, Correia AM, Adeli H, Reis LP, Costanzo S (eds) Advances in intelligent systems and computing, vol 570. Springer International Publishing, Cham, pp 197–206

Luxen D, Vetter C (2011) Real-time routing with OpenStreetMap data. In: Proceedings of the 19th ACM SIGSPATIAL international conference on advances in geographic information systems. ACM, New York, pp 513–516

Mooney P, Corcoran P (2014) Analysis of interaction and co-editing patterns amongst OpenStreetMap contributors. Trans GIS 18(5):633–659

Open Source Routing Machine (n.d.) https://www.openstreetmap.org/. Retrieved 14 March 2019

Pereira RL, Sousa PC, Barata R, Oliveira A, Monsieur G (2015) CitySDK Tourism API—building value around open data. J Internet Serv Appl 6(1):1–13

Rajapaksha P, Farahbakhsh R, Nathanail E, Crespi N (2017) iTrip, a framework to enhance urban mobility by leveraging various data sources. Transp Res Procedia 24:113–122

Rebelo J, Caldas J, Guedes A (2015) The Douro Region: wine and Tourism. Almatourism J Tour Cult Territ Dev 6(11):75–90

Ribeiro FR, Silva A, Barbosa F, Silva AP, Metrôlho JC (2018) Mobile applications for accessible tourism: overview, challenges and a proposed platform. Inf Technol Tour 19(1–4):29–59

Richards R (2006) Representational state transfer (REST). In: Pro PHP XML and web services, pp 633–672

Silva CA, Toasa R, Guevara J, Martinez HD, Vargas J (2018) Mobile application to encourage local tourism with context-aware computing. Springer, Cham, pp 796–803

SQLAlchemy (n.d.) https://www.sqlalchemy.org/. Retrieved 14 March 2019

Thiengburanathum P, Cang S, Yu H (2016) Overview of personalized travel recommendation systems. In: 2016 22nd International conference on automation and computing, ICAC 2016: tackling the new challenges in automation and computing, pp 415–422

Vansteenwegen P, Souffriau W, Vanden Berghe G, Van Oudheusden D (2009) Iterated local search for the team orienteering problem with time windows. Comput Oper Res 36(12):3281–3290

Vansteenwegen P, Souffriau W, Van den Berghe G, Van Oudheusden D (2011) The city trip planner: an expert system for tourists. Expert Syst Appl 38(6):6540–6546

Wang HD, Li L (2010) Research of process concealment based on technology of intercepting API calls. In: 2010 3rd International conference on computer science and information technology, vol 7, pp 412–414

Wang M, Li Q, Hu Q, Zhou M (2013) Quality analysis of open street map data. In: 8th International symposium on spatial data quality, vol 40, pp 155–158

Yan C, Cheung A, Yang J, Lu S (2017) Understanding database performance inefficiencies in real-world web applications. In: Proceedings of the 2017 ACM on conference on information and knowledge management-CIKM'17, pp 1299–1308

Yang J, Subramaniam P, Lu S, Yan C, Cheung A (2018) How not to structure your database-backed web applications: a study of performance bugs in the wild. In: ICSE, pp 1–11

Zielstra D, Zipf A (2010) Quantitative studies on the data quality of OpenStreetMap in Germany. In: Proceedings of GIScience

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

المنارة للاستشارات

Reproduced with permission of copyright owner.
Further reproduction prohibited without permission.