TÉCNICO LISBOA

Fidel

A tour building and experience sharing web and mobile platform

Filipe Miguel da Silva Teixeira

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor(s): Prof. João Nuno de Oliveira e Silva

Examination Committee

Chairperson: Prof. António Manuel Raminhos Cordeiro Grilo
Supervisor: Prof. João Nuno de Oliveira e Silva
Member of the Committee: Prof. Jaime Manuel Moleiro Serra

June 2017

Dedicated to Fidel...

Acknowledgments

What a long, strange trip it has been. It certainly took more time to complete the masters than I wished for, but in the end, I am proud in what I was able to learn and achieve. This thesis in particular was an ambitious project that took an enormous amount of time to complete, and I would not be able to do it without all the support and help from my family, friends and supervisor. First of all, my deepest thanks to my parents for their love, extended support and comprehension, thanks to whom I was able to resiliently force myself to deliver the best work I could. My sincerest gratitude to Prof. João Nuno de Oliveira e Silva for the collaboration in defining the project goals and stick to them, and continuously pushing the limits, and to INESC-ID for the financial support and for providing me a place to belong to. To Gonçalo Silva and Ruben Machado, I will keep all the coffees and talks, all the support we gave each others as a precious memory, and know we will keep this friendship forever, to João Pereira, thanks for all the good times when needed the most, to Daniel Arrais and Marta Lima, all the work we did together was fantastic and the logic thinking, the code debugging and the moral raising practised in our little circle was, without a doubt, the final push needed to make this delivery a reality. To all my unmentioned friends, thank you for all the support and for being a part of my life.

Resumo

Nos últimos anos várias aplicações orientadas para o turismo têm facilitado a vida dos turistas no acesso a informação acerca de Pontos de Interesse (POI) em todo mundo, ao fornecerem peer-reviews, rotas recomendadas e ofertas de guias turísticos. No entanto, os guias turísticos têm maioritatiamente usado tecnologia que não é adequada às suas necessidades em termos de criação e planemento de rotas e respectivo conteúdo multimédia. Além disso, não existem soluções tecnológicas para partilharem este conteúdo com os seus clientes. Este documento propõe e implementa um modelo de dados e arquitectura para uma aplicação que serve como plataforma de desenvolvimento para guias turísticos criarem pontos de interesse e agregá-los em rotas de forma fácil e intuitiva de forma a acelerar o desenvolvimento de conteúdo e aumentar produtividade. Este sistema agrega informação de múltiplas fontes e permite acesso à sua informação através de serviços web disponibilizados via APIs RESTful. São também propostas e desenvolvidas duas aplicações móveis que têm o objectivo de responder às necessidades dos guias turísticos que utilizam os servicos providenciados pela aplicação web implementada, sendo uma delas utilizada pelos guias, a outra pelos turistas. A primeira tem o objectivo de descarregar as rotas criadas na aplicação web e partilhar o seu conteúdo multimédia com os visitantes, a segunda o de receber este conteúdo. Ambas as aplicações foram desenvolvidas considerando a necessidade de conseguirem funcionar sem a utilização de internet e seguindo princípios de Software Open Source.

Palavras-chave: Turismo, Guias Turísticos, Partilha Multimédia, Plataforma de Desenvolvimento

Abstract

In the recent years, many tourism-oriented applications have facilitated the life of tourists by providing access to information about Points of Interest (POI) around the globe along with peer-review, recommended routes and tour guides offerings. However, tour guides are mainly using technology that is not tailored to their needs in terms of creation and planning of tours and respective media content. In addition, there are no available technological solutions to share this media with their customers. This document proposes and implements a data model and architecture for an application that serves as a development platform for tour guides to create POI and aggregate them into tours as easily and intuitive as possible, in order to accelerate content development and increase productivity. This system aggregates information from multiple, scalable sources and allows access to the information as a webservice via a RESTful API. There are also two mobile applications proposed and developed with the necessities of tour guides in mind that use the webservices provided by the web application implemented, where one is to be used by tour guides and the other by the tourists. The first's main objective is to download the tours created on the web application and share their media with the visitors, the second, receiving this content. Both applications were developed considering the necessity to be able to work without an Internet connection and following Open Source Software principles.

Keywords: Tourism, Guides, Media Sharing, Development Platform

Contents

List of Tables

List of Figures

Glossary

API	Application Programming Interface
CRUD	Acronym for Create, Read, Update, Delete, operations usually related with data base management
CSV	Comma-Separated Values, a common way to store tables in plain text format
DBMS	DataBase Management System
HTTP	HyperText Transfer Protocol
LAN	Local Area Network
MVC	Model-View-Controller
OS	Operating System
POI	Point(s) of Interest
REST	Representational State Transfer. A methodology for interaction with web servers based on state methods
SPA	Single Page Application
SQL	Structure Query Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WFS	Web Feature Server
noSQL	Not Only Server Query Language

Chapter 1

Introduction

Tourism is accepted as one of the world's largest service industries, having experienced continued expansion and diversification, becoming one of the largest and fastest-growing economic sectors in the world [1]. According to the World Tourism Organization (UNWTO), international tourist arrivals grew by 4.4% in 2015 to reach a total of 1,184 million, marking the sixth consecutive year of above-average growth with international arrivals increasing by 4% or more every year since the post-crisis year of 2010, earning 1,522 billion USD [2] in exports. The competitiveness of the market place and the increased expectations of customers have made tourism industry recognize the importance of customer service quality for future revisit and referral business [3, 1], namely tour guides and other front-line staff. Having this importance in mind, providing tour guides with relevant and useful tools is fundamental to a better performance. However, considering the sheer amount of platforms and applications currently developed, tour guides are comparatively very ill-equipped in terms of technological tools at their disposal to provide a better experience to their customers. Technological platforms focused on tourism are generally focused on the tourists only instead, leaving professionals lagging behind.

A tour is a journey for pleasure in which several different places are visited, and each of those places is considered a Point of Interest (POI). From a technical standpoint, a point of interest is "a location for which information is available". In practice, POI are usually places that serve a public function, thus excluding facilities such as private residences, but including many private facilities that seek to attract the general public such as retail businesses, amusement parks, industrial buildings, Government buildings, and natural features [4]. After personally inquiring tour guide professionals, it became clear that tours are usually planned on notebooks or using spreadsheets and, during tours, sharing media is heavily discouraged. When absolutely necessary, any media shared is done by printing the images into papers and passed along the customers, which leads to a distracting and overall unsatisfactory experience [5]. To solve the planning problem, there are a few services that specialize in POI all over the world, although few of them actually provide an easy way to create custom tours with custom POI and the ones that do, do not provide any interface to externally interact with the data and therefore allowing the reuse of the created content for other applications. To share media, there are no solutions with tour guides needs in mind.

The tourism industry would greatly benefit from a platform that would allow the creation of custom POI and tours, tailored for tour guides, to enhance and facilitate the creation of planned tours through the usage of an application designed with these needs in consideration and a way to share this curated content with their customers in order to provide a technologically-enhanced guided visit [6], preferably without requiring the use of an Internet connection to spare mobile data plans.

1.1 Proposed work

This thesis hereby proposes a platform for the custom creation of POI and tours that can be edited in a collaborative way amongst its users, integrating curated information from external POI providers. Users can create public or private tours and POI, a setting that can be changed at any time, and add curated descriptions and media. An approach to these requirements has been done by Mauro Teixeira [7] but entity structuring, user experience and the desire to develop a platform under Open Source Software motivated a full rewrite of that implementation.

In addition, using a mobile application, the created content must be downloadable to a mobile device so its relevant media can be shared on demand to the tourists, who will be presented with a third application consisting on a web page accessible from any mobile web browser that receives the media being broadcast. This will represent a novel approach for a solution to the current media sharing options available to tour guides and the consequent modernization attempt for this classic method. Figure 1.1 demonstrates the proposed architecture described.

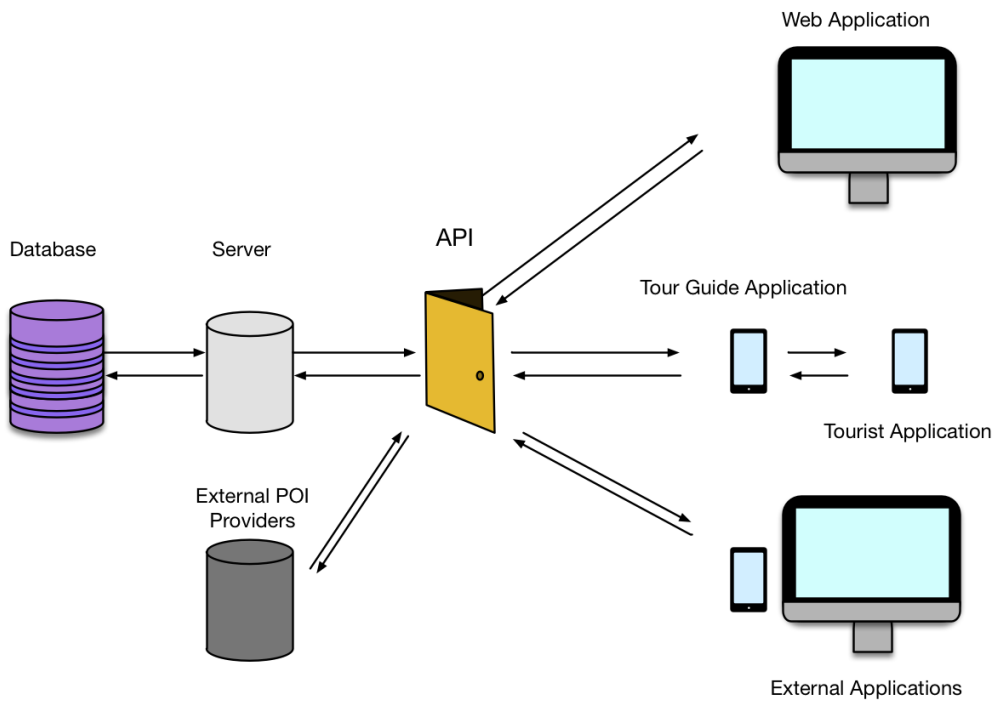

Figure 1.1: Proposed platform architecture.

1.2 Thesis Outline

Chapter two will study the tourism panorama with more detail, review existing solutions for tourism-related problems and gather a list of data sources ranging from POI providers, map services and geocoding solutions. Then, web services architecture will be studied along with data exchange formats involved in those communications. Lastly, mobile operating systems and their existing development platforms and methodologies will be reviewed.

Chapter three will state the requirements for the developed platform, specifically the functional requirements from the three pieces of the proposed platform and its general architecture.

Chapter four will discuss the technologies adopted in order to fulfill the requirements aforementioned and the factors that led to the decision of each component faced against comparable, available solutions.

Chapter five will compare the end product developed with the initial established requirements and discuss whether they were properly satisfied and present evidence, and then study the energy consumption of the mobile solution on a device.

Chapter six will conclude with the achievements of this project and discussion of relevant future work.

Chapter 2

Background

In this chapter, a review of tourism and its importance will be established in order to acknowledge the relevance of the sector for society, to understand tourists expectations and how to improve their experiences. Then, existing platforms and services that are focused on tourism will be studied to understand the existing problems and solutions provided, whether they are sufficient or not and if not, what can be done to effectively solve the identified problems. Following up will be the overview of adequate data models on which to rely on to build the applications proposed by this thesis. After that section, existing data sources for POI and Map functionalities will be studied in order to later on select the most appropriate ones to integrate the developed application. The two consequent sections will provide a technology review of webservices and mobile operating systems and its possible development environments, respectively.

2.1 The importance of tourism

Tourism is generally considered as travel for pleasure or business [8]. As a sector, it is composed by the business of attraction, tour guiding, accommodations, tourist entertainment and operating tours [9]. Portugal2020 [10], the result of a partnership between Portugal and the European Commission, describes tourism as a ”fundamental activity for its ability to generate wealth and jobs, and for its contribution to the regarding of natural, cultural and patrimonial resources” and, as such, is a topic of great macroeconomic interest, as the Portuguese Tourism sector generated a net value of 8.8 billion euros in 2016 from a gross income of 12.6 billion [11], with a total of 53.5 million overnight stays in 2016 [12], where Lisbon took the ninth rank in metropolitan regions with the highest number of night stays in Europe (total, non-resident) [10].

According to Turismo de Portugal [13], guided tours are the second most desired activity on tourism entertainment and were consequently a focus of several studies to find solutions to make them even more appealing and effective in promoting tourism in Portugal and the return of tourists. One of those studies [14], a survey done to tourists, concluded that 94% of the interviewed were very satisfied with their overall touristic experience in Portugal, but identified infrastructure and touristic information as

points of contact with room for improvement. This conclusion proved the SWOT analysis done by Nunes [15] in 2014, that identified a weakness in the low evidence of Innovation and Development applied to tourism and lack of partnerships between the technological and scientific system and the tourism companies. This analysis also highlighted the rapid change from the destination/geography to the product/experience as deciding factor for tourists as a threat to the sector, a concept already foreshadowed in 1998 by Pine et al. in the book *The experience economy* [16]. The latest strategic plan done by Turismo de Portugal [17], therefore, emphasizes the need of "production and availability of content, development of digital support and technological applications to enhance tourism experiences on Portuguese patrimony and territory". This is interpretable as taking advantage of technology to enrich an experience and make it immersive and interactive [6], an apparently crucial directive for the tour guide professionals who now struggle for a digital platform to enrich the provided experiences.

Looking at the current technological panorama, the usage of smartphone is present in everyday living and greatly impacted routines and habits of the world population [18]. Smartphones and their applications reflect the technical potentials and capabilities of today's tourists [19], a sector with an identified growth potential. Dickinson et al. [20] identified that "Leading visitor attractions and destinations are increasingly adopting smartphone app technology" and as such, not following this trend will likely result in a loss of interest by potential customers in favor of adopting countries. In order to accomplish this task successfully, Brown and Chalmers [21] state that "good tourist technologies are not only those that make tourists more efficient, but that also make tourism more enjoyable" and according to Grün et al. [5], "Providers of mobile tourist guides should especially focus on tourist attractions, events, entertainment and gastronomy, and provide well implemented, customized versions in order to differentiate themselves from others", giving the clear example that "Pushing relevant information to the users can relieve them from the tedious task of looking it up themselves". IPK International [22], a world tourism marketing consultant, concluded that millennials want authentic travel experiences. This "first global generation" seeks authenticity, experience new cultures and are generally happy to provide their personal data to be used in order to create individualized travel experiences, such as through mobile app offering specific personalized access and services. For destinations, this means seamless travel experiences should be offered by using technology to simplify and streamline the enjoyment of experiences.

Although smartphone usage is abundant, it encounters one peculiar pitfall in international tourism, the inconvenience of mobile data plans for Internet usage. This fact forces tourism related applications to consider the offline capabilities with high regard, as free wifi hotspots may be scarce and having solutions fully dependent on Internet connectivity may be unfavored for not taking this point in consideration.

In the next section, tourism-related platforms that try to solve identified problems will be highlighted and reviewed.

2.2 Existing Platforms

First of all, an overview of the existing platforms will be done for us to be able to analyze and highlight unexplored, but needed features.

2.2.1 Platforms Overview

As a goal to satisfy the needs of tourists, tourism industry and developers looking to tackle this problems, several platforms have been designed. Finding the most relevant ones for the problem we wish to tackle is not a trivial task and as such, the solutions presented below are all part of the sector in which we wish to work on, even if each of them may present different approaches to slightly different identified problems. The reviewed mobile and web platforms will not only range from solutions tailored for the tourism industry but also approaches that can be used to fulfill the needs of the sector, even if not their original goal.

Vayable

Vayable [23], presented in figure 2.1, is an online and mobile (iOS only) platform offering guided tours done by locals. Any person with knowledge in an area of interest for possible tours, such as wine tasting, photography or nature walks, can sign up to be a Vayable insider, providing tour guide services for a hourly rate defined by the guide. Vayable provides its insiders with a well detailed profile that highlights their interests and areas of knowledge, area of operation and biography, along with all the tour offerings. The platform act as a billing and scheduling system, receiving a cut from the transactions done between the tourists and the guides. Vayable was founded in 2010 and received venture capital funding as a YCombinator startup, gathering 1.400.000$ initial funding and quickly became the international reference as the ”AirBnB of tours”. This community driven project also features a well-defined and documented API composed by all the functionalities required to fully interact with the website. However, the tour data model used does not consider POI and, as such, is not granulated enough to properly define a tour as any visitor will only be presented with media, location and a description of the tour itself.

Meetrip

Meetrip [24] is an online and mobile platform founded in 2015 very similar to Vayable in objectives and functionalities. The biggest difference is that is targeted at accredited professionals in the tourism industry. This differentiating factor to the Vayable platform came from gathering feedback of distrust in unaccredited tour guides. Like Vayable, tour guides are provided with a scheduling and billing service, along with a detailed profile with an extensive bio, interests, area of operation, listed tours and hourly rate.

Meetrip provides tourists with the opportunity to book a tour guide for a personalized tour, with a hourly rate defined by the guide in an easy and intuitive way, whereas in Vayable, an interested tourist had to personally contact a tour guide and directly ask for this service, a feature not natively supported. As a content creation platform, Meetrip is also very basic as it does not define POI and the online content about the tours is minimal, only relying on the tour description. Most of the planning and routing is left to the tour guide to deal personally, with whatever tools he may have.

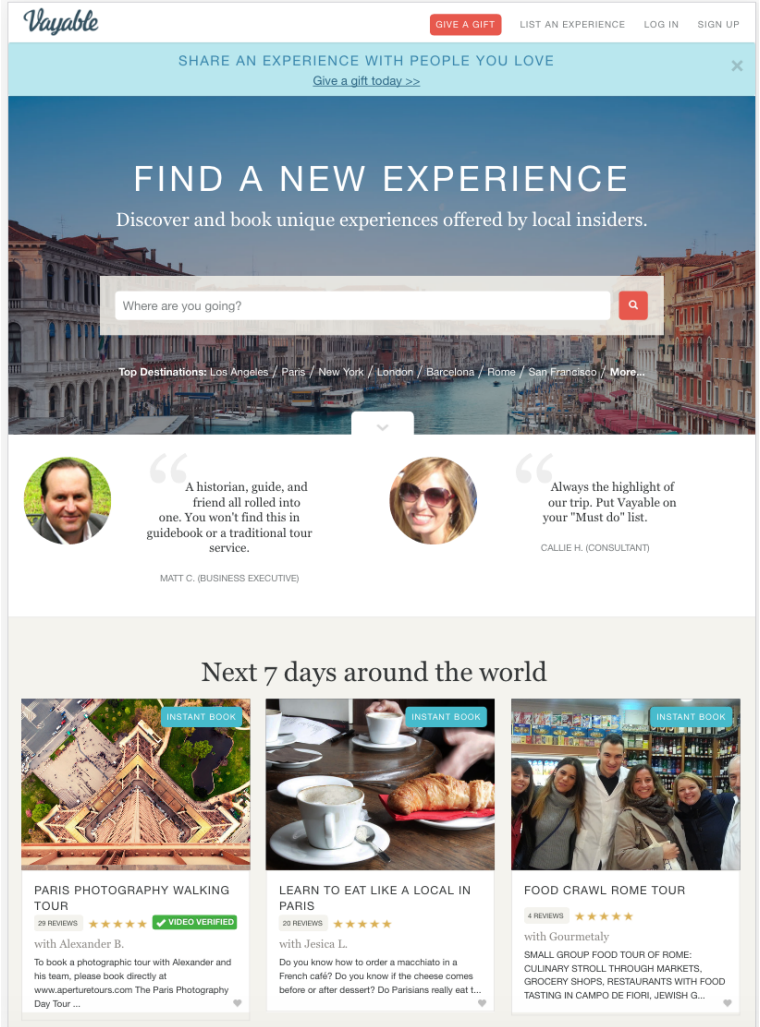

Figure 2.1: Vayable website [23].

Anyroad

Anyroad [25] is a platform to help developing tourism industry, providing a booking management solution, invoicing, scheduling and messaging for tourism professionals. It has a strong focus on social engagement and boosting virality, with connections to several social networks to gather and share feedback. It also addresses brand impact and customer loyalty with a strong emphasis, providing several metrics for pre-experience, post-experience and lifetime, categorizing the customers as detractors, passives or promoters, based on their feedback. Overall, Anyroad's experience relationship management (ERM) empowers the professionals with metrics and tools to reach out to an audience as fast and efficiently as possible, by following five principles; "start selling fast, deliver a customer experience, stay on brand, engagement is gold and know the value of the customer". Defining the tour, with descriptions, media and well defined POI is left out completely though, as tour building tools are non-existent. Even so, it is a platform worth mentioning in the context of this work by what it achieves.

Eurotrip Planner

Eurotrip planner [26], displayed on figure 2.2, is an online application that allows users to create their tours, composed by predefined POI from Europe, and then export them to a mobile application. It is a side project created by Michael Van der Veen and is not well maintained presently, with several broken links on the web page and little information that could be of help for a more detailed study. It is, however, an approach previously unexplored on the tour building scene, as its architecture enables the creation of a detailed tour and the download for a tablet or smartphone where it can be reviewed without the need for an internet connection, an identified important aspect for tourism applications. The POI presented in the application are premade and cannot be customized by the tourists, only used and aggregated into tours, and the lack of API denies external applications interactions.

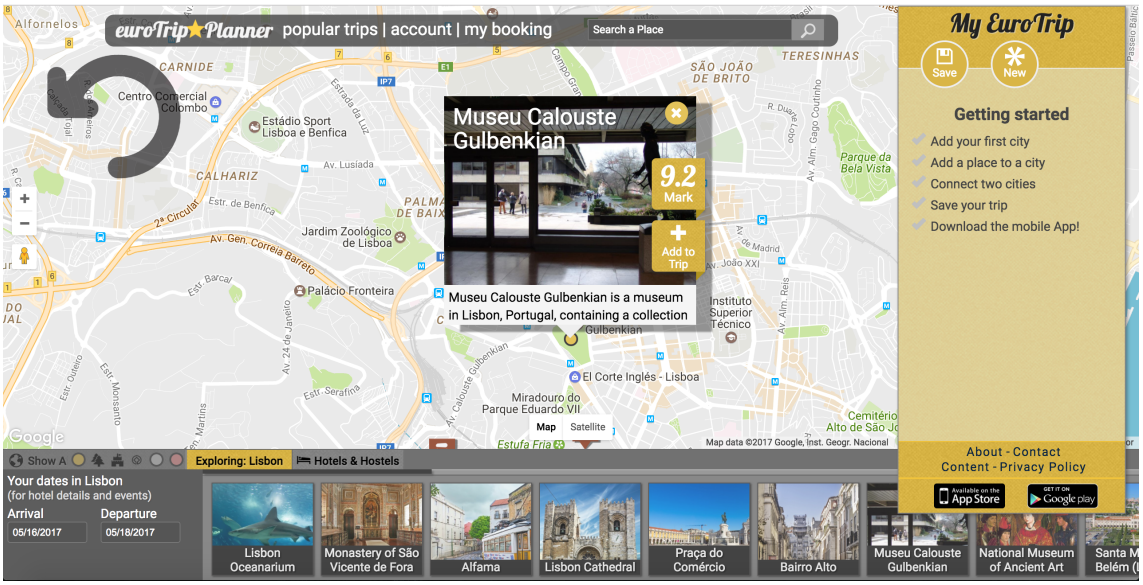

Figure 2.2: Eurotrip Planner website [26].

RPoi

RPoi [7], featured on figure 2.3, was the result of a master thesis done by Mauro Teixeira at Instituto Superior Técnico in which this work took inspiration on. It features a well defined data model of POI, groups of POI, external providers and privacy control over the tours (whether they can be made publicly available or private for the owner or group of collaborators with access to the tour) focused on a community driven initiative to aggregate and curate POI related data. Even considering the fact that it is a thesis work, the user experience and user interface are very poor, with a lot of core functionalities completely obscured due to the unintuitive design. For example, to create a POI, a user must right click the map to be presented with the possibility to do so, a procedure not only obscure but also never mentioned anywhere on the website. The API provided by this platform only supports GET methods, denying any interaction with the web application other than fetch data, making it impossible to create or edit content from external applications.

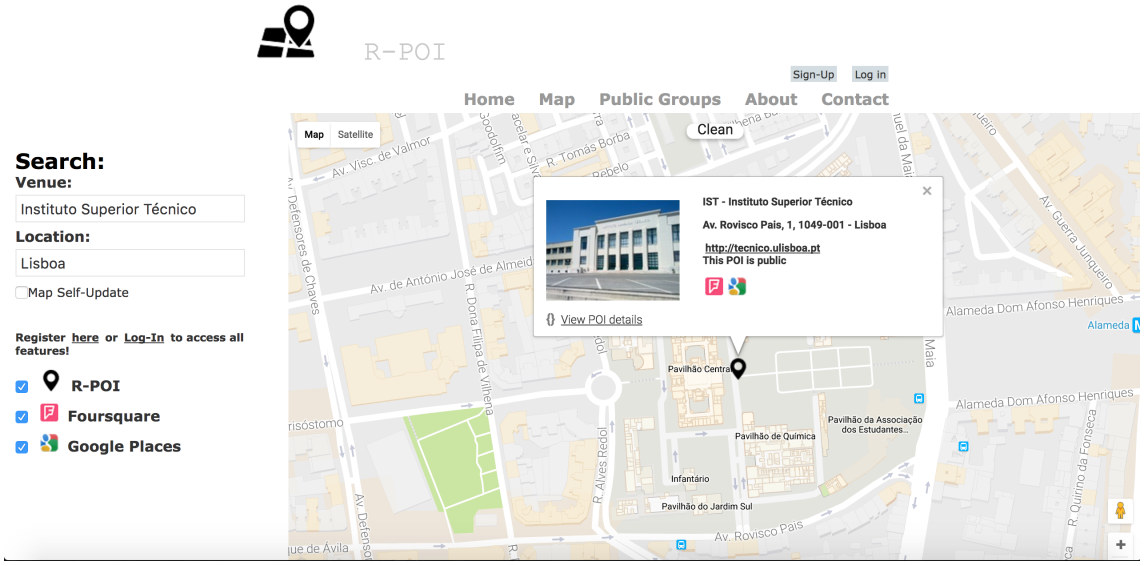

Figure 2.3: RPoi website [7].

Google Tour Builder

Google Tour Builder [27], seen in figure 2.4, was originally created to give war veterans a way to record all the places the military service has taken them, and preserving their story and memories as a legacy for their families. After realizing the potential of the tool, it was made public to be used by anyone who desired to tell a story. It lets users pick locations on the map or search for POI, add notes and media files and building up tours composed of several POI, with a detailed, granular and comprehensive data model for tours. This tool could be a great asset for a tour guide to develop his tours in order to later share with the tourists, but the sharing is done via email. If there is a desire to share a tour with the interested tourists beforehand, this tool serves that purpose well, if it's to be used as mean to an enhanced touring experience, it does not. On top of that, unfortunately, this service provides no API nor clearly defined webservices to reuse the information, making it unfeasible to integrate with third party applications.

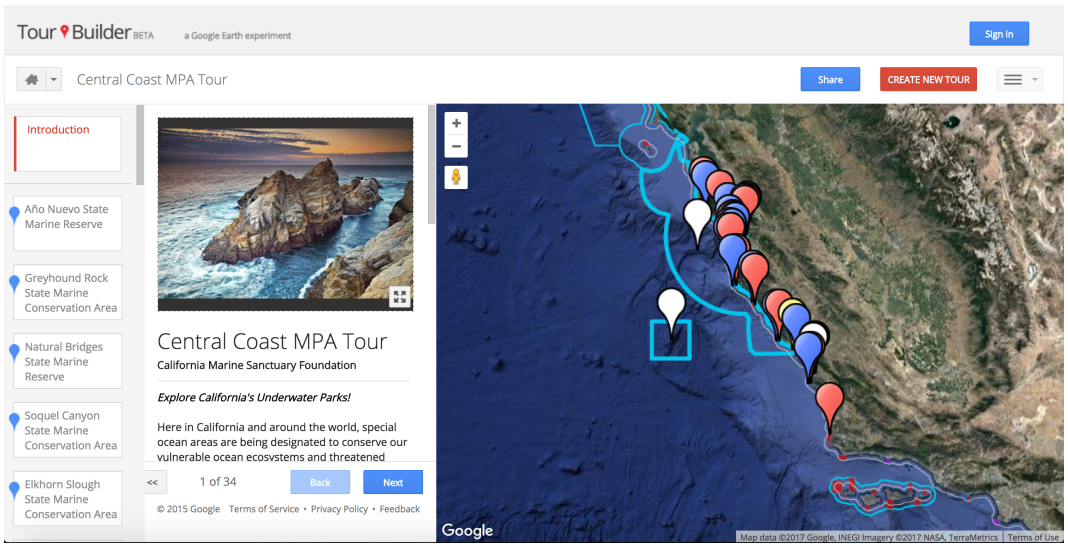

Figure 2.4: Google Tour Builder project [27].

mTrip

mTrip [28] is a very recent mobile application for Android and iOS focused on the travel industry, with several desirable features for travelers and tourism professionals. mTrip serves as a sales point for pre-made, curated tours that can be bought from the application and will then be available for offline usage, with some customization available, namely custom POI addition and all related information regarding accommodation details and hotel check-in. For tour guides, mTrip offers a business solution that consists of a blank app, customizable with a brand, published under the client company identity and offers booking, planning, satisfaction surveys, messaging and social network integration for post experience enhancement.

In November 2016, mTrip (shown in figure 2.5) launched a service for this sector called *Tour Leader*, where a designated tour leader can track the location of the other members, send direct or broadcast messages and push notifications; multimedia sharing is not supported. mTrip also has an password protected API documentation to which was not possible to get more information. Pricing of the platform is undisclosed, forcing interested audience to directly contact the sales representatives.

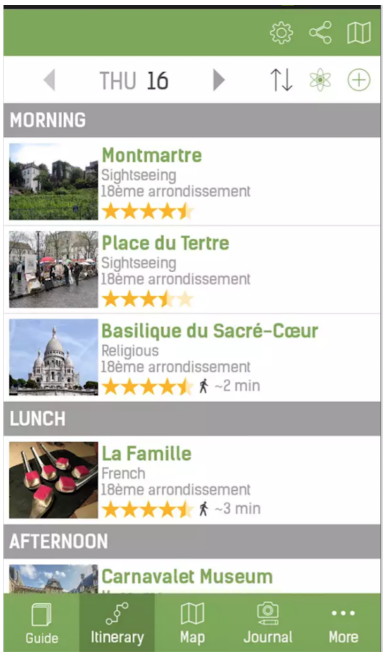

Figure 2.5: mTrip mobile application [28].

2.2.2 Feature Analysis

After the previous overview, it is now possible to compare the provided features of the studied applications and come to the conclusion that there is little to no software available for tour guide professionals to rely upon to create their customized tours and share them with their audience; there are several solutions that allow them to become more visible and easier to reach by tourists, to handle the financial and scheduling aspects of the tours, but to create customized tours, POI and associated media, the only solutions found were the thesis work developed by Mauro and Google Tour Builder, that does not have

the necessities of tour guide professionals in mind. In order to be able to share this curated content during the tours, it is desirable to have a mobile application that uses this information and broadcasts it on request by the guide to the customers devices, something not provided by any of the analyzed platforms.

In order to develop a web and a mobile application suitable for tourism, relevant concepts will be reviewed in the following sections.

2.3 POI Data models

Data models are the data architectural support of an application defined according with the data entities that cover all the necessary information to be supported. As such, the base of a tourism application should be the data model that envelops the points of interest.

The W3C (World Wide Web Consortium) launched, on September 13th 2010, the ”Points of Interest Working Group” [4] with the mission to ”develop technical specifications for the representation of Points of Interest information on the Web”. This work was later continued by the OGC (Open Geospatial Consortium), incorporating the Points Of Interest Standard Working Group. At the time of publication of this thesis, no standard has yet emerged from this group. The guidelines of these projects have defined a POI as an entity with a name and a description, that can be categorized according to not yet defined categories, ranging from type of venue to popular rating, as presented in figure 2.6. The POI can have a time field that represents the start date or end date for that place, a location represented in either coordinates or bounding box. This project tries to standardize a data model for representing POI and is a good start to modeling such application, even if superseded by additional definitions.

The standardization the OGC tries to accomplish has the goal to promote a well-defined interaction between different applications that have POI as a common ground and, therefore, promote the usage of open data.

2.4 Data Sources

To represent POI, tourism-related applications rely on a map interface to visually display the location of the points, even when composed by additional media and content in some platforms. This is based on the nature of a Point of Interest being, in essence, a geographic location and, while the address or even the exact coordinates may not be the most identifiable attributes of the point in question, in a tourism perspective, is one of the most important informations that must be properly perceived by the audience, as it may have a strong relevance for tour scheduling.

Providing a map and serving information about it are two distinct functions, the first is generally done via a tiled map, the second is nominated geocoding, the process of converting a description of a place via coordinates or address into a location on the earth’s surface, therefore mapping information into a point on a map tile. A tiled web map is a map displayed by seamlessly joining several image files, called map tiles, into a map display that can usually cover the entire earth surface. Each time the user pans,

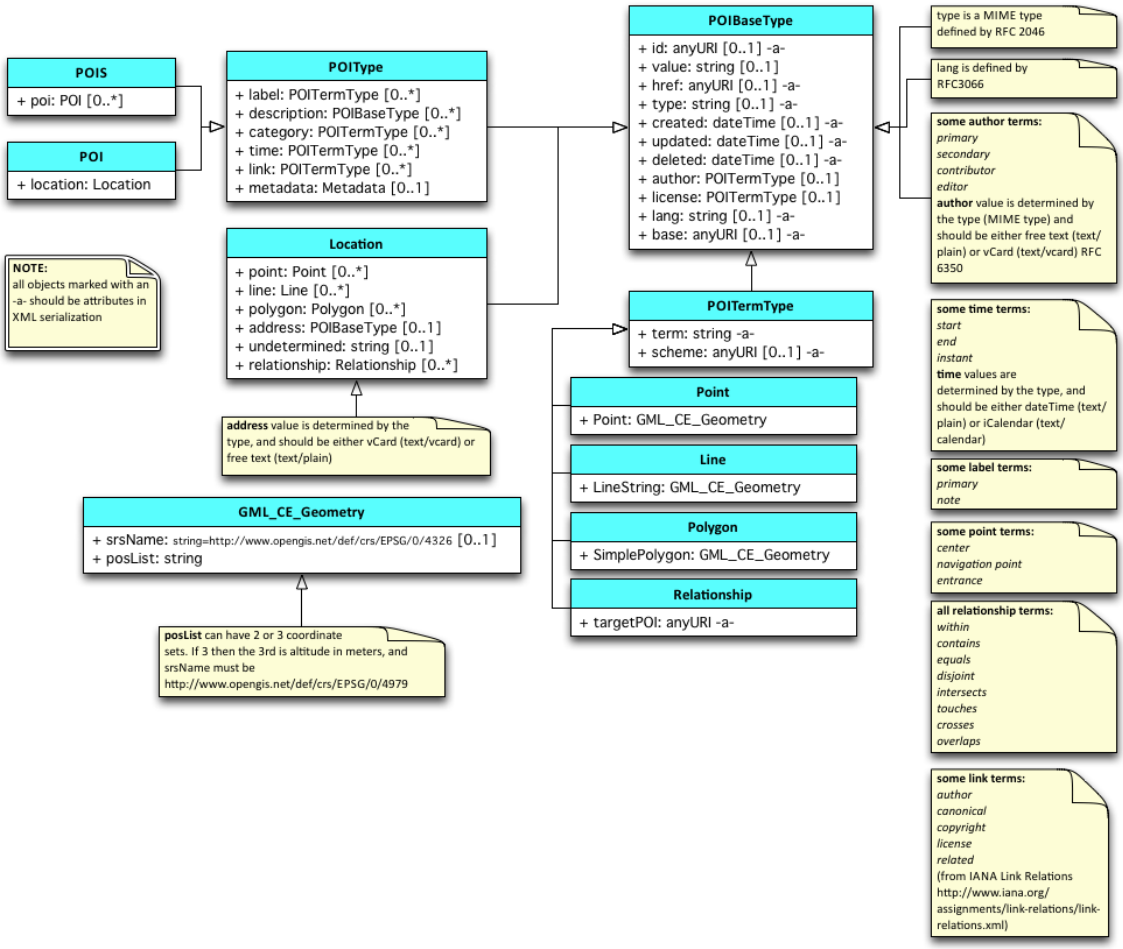

Figure 2.6: W3C Data Model for Points of Interest [4].

most of the tiles are still relevant while new tiles are fetched. Geocoding is the process of converting a description of a place via coordinates or address into a location on the earth's surface, therefore mapping information into a point on a map tile. Advanced geocoding service providers will also feature relevant information to the locations, such as categorization of the place, metadata, full address, state, among other tags. These two services, although intertwined, are very distinct and even if they can be served by a provider that features both, it may not be the case in several scenarios. In addition, there are multiple information providers that act upon geocoding to present additional information to a location that purely rely on external map services, namely, most of the POI information providers that will be covered in the section bellow.

2.4.1 Mapping Services

Amongst map providers, we will mention the following relevant ones that can be embedded on external applications.

Google Maps

Google maps [29] is a web mapping service developed by Google that provides several map-related services, such as satellite imagery, 360º panoramic views of streets (Google Street View), real-time traffic conditions (Google Traffic) and route planning for traveling by foot, car, bicycle or public transportation. Google maps offers an API that allows maps to be embedded on third party websites and applications. To use this service, a customer can opt for one of two plans; It has a free/standard plan with a quota of up to 25.000 map loads per day, with a price of $0,50USD /1.000 additional map loads, up to 100.000 daily. This plan has a requirement which states that the client application must be free to use and public to anyone, except if the client application is a mobile application that is on an app store and only uses the APIs that are specific for those cases (Google Maps Android API or Google Maps SDK for iOS). For all other cases, namely any web application that is either access restricted or paid in any form (subscription or registry fee), one must sign for the premium plan, with undisclosed prices. This service is well documented, heavily used across the Internet and relatively easy to implement.

OpenStreetMap

OpenStreetMap [30] is a web mapping service developed by OSM Foundation built by a community of enthusiast and background diverse mappers that contribute and maintain data about roads, trails and POI all around the globe. OSM emphasizes local knowledge, with contributors using aerial imagery, GPS devices and low-tech field maps to ensure that OSM data is as accurate and extensive as possible. All this data is open, meaning that its usage is free for anyone as long as OSM and its contributors are credited. To use OSM services extensively (in other words, if a lot of traffic is expected), users are asked to set up their own tile servers, a well documented process on replicating the OSM database data to an owned server, as OSM Foundation is a non-profit organization that relies on donations in order to keep their servers running and, as such, exhausting the few resources provided by good will is unethical. This service is extensively used by a lot of third party applications and its integration with leafletJS makes the process of integration with a web application a relatively simple task.

Bing Maps

Bing maps [31] is a web mapping service developed by Microsoft that also provides several map-related services, launched in response to Google Maps as a rebranding of its existing service since 2000, Microsoft Visual Earth [32]. Bing Maps offers an API that allows the embedding on third party applications as well, with two kinds of licenses available to developers, free and a paid. The free quotas presented by Bing Maps are much shorter than the quotas provided by Google Maps, as it has a limit of 125.000 transactions per calendar year. Paid licenses have undisclosed pricing that is only available upon form request and direct contact with a sales representative. The API operations are also heavily and detailedly documented, presenting their implementation not a great challenge as well.

2.4.2 Location Services

This section will cover some geocoding solutions available to properly translate a location into a well defined coordinate pair.

Nominatim

Nominatim [33] is a search tool from the OSM Foundation that is the core geocoding tool of OpenStreetMaps. In addition to providing geocoding and reverse geocoding services, it also provides metadata for the searched locations, such as city, state and zip code. It is a tool powerful enough to find POI by name as a search parameter (for example, ”Instituto Superior Técnico” correctly interprets the location where the Portuguese University is located on), making it a very elastic service to rely on to search for common designated POI as a means to efficiently map a search parameter with a geographic location with the best possible User Experience. Like OpenStreetMaps, all information on Nominatim is open data and can be used freely by anyone as long as its creators are credited and for intensive usage, setting up a mirror server is also advised.

Google Places

Google Places [34] is the local listings of the Google services that fetches data from the same database used by Google Maps and Google+ Local. Places features more than 100 million establishments, geographic locations and prominent Points of Interest that are updated frequently through owner-verified listings and user-moderated contributions. The Google Places API webservice returns information about places using HTTPS requests, with a free quota of 10000 requests per day, and a billing plan if more than that is necessary.

2.4.3 Social Networks POI providers

Having reviewed map and location services, we will now focus on two Social Networks Information providers, Foursquare and Facebook.

Foursquare

Foursquare [35] is a technology company that focus on user location intelligence to build meaningful consumer experiences and business solutions. It provides two applications to serve that purpose, Foursquare City Guide and Foursquare Swarm. City Guide is a community-reviewed platform for Points of Interest categorized into popular topics, such as nightlife, fun or shopping, for everyday activities. It provides user recommendations and ratings for the venues and gives suggestions based on the search history. Swarm is a gamification of the everyday life, as it allows users to check-in venues and compare themselves with others, to the point where the most active user in a venue gets to be ”crowned the mayor”. Its objective is to incentivize people to explore the world around them more often and keeping track of the journey. Foursquare provides APIs for the venues in its database that can be embedded

onto third party applications, providing a solid POI database for population of content, with descriptions, location, user reviews and user uploaded media.

Facebook

Facebook[36] is the most popular social network in the world that not only serves as an online point of presence for any individual, entity or group, but also provides several services, namely a service called Graph API. Graph API is the primary way to read and write to the Facebook social graph, composed by a massive amount of data. Amongst the available search types, it is worth mentioning in the scope of this project the ”search place” type, allowing queries to search a specific location or a location plus a radius (in meters), or simply by name.

2.4.4 Institutional Databases

In this section, institutional databases that serve POI related open data to foster external applications will be covered and studied.

Direcção Geral do Património Cultural

The DGPC [37] is responsible for the management of the cultural heritage in Portugal, headquartered in Lisbon. It is tasked with the study, investigation and disclosure of all kinds of patrimony and management of architectural and archaeological patrimony in all the country. Amongst all the services it provides, DGPC offers an open data API with information about a vast number of POI in the Portuguese territory on a platform called Informação Geográfica that can be consumed as a RESTful service via WFS, without established quotas. It also has two databases related to archeology and patrimony currently without an associated webservice.

Dados Abertos de Lisboa

The DAL platform [38] (Lisbon Open Data) provides datasets from the city produced by the Lisbon city council and by partner entities. Data access is free, in order to enhance the reuse and the production of goods and services that will add value to the contents available. Amongst the vast datasets available, there are several focused on patrimony and tourism that can be consumed from its RESTful APIs.

2.5 Webservices and data exchange formats

As said by John Wilbanks, VP Science of Creative Commons [39], ”numerous scientists have pointed out the irony that right at the historical moment when we have the technologies to permit worldwide availability and distributed process of scientific data, broadening collaboration and accelerating the pace and depth of discovery ... we are busy locking up that data and preventing the use of correspondingly

advanced technologies on knowledge." This is an evident phenomenon on the amount of reimplementation of the same solutions from different providers all over the Internet and should serve as an incentive to design and create solutions with an open data mindset, not only to use already existing data from other platforms, but also to empower external applications, contributing to a better scientific and technological panorama [40]. According to the W3C definition, a web service is described as "a software application identified by a URI, whose interfaces and bindings are capable of being defined, described, and discovered as XML artifacts. A Web service supports direct interactions with other software agents using XML-based messages exchanged via Internet-based protocols." This definition was established back in 2004 and is currently a little outdated, as new data exchange formats have emerged and became massively implemented, namely JSON that will be covered below, that took the role away from the implicit XML exclusivity. A more broad and easy to understand definition is done by Alonso et al. [41], as "an application accessible to other applications over the Web". The apparent vagueness of this definition makes it resilient to the test of time and fully explains the base concept accurately.

In order to properly access other applications over the Web, there is a need for an interface to connect upon, which is commonly known as Application Programming Interface, typically defined as a set of HTTP request messages, along with a definition of the structure of response messages, in a web context. One of the most common architectures built to satisfy the needs to integrate external applications and facilitate web communications, taking advantage of web services and APIs is known as REST, that stands for REpresentational State Transfer. REST is an architecture style that encourages applications to be simple, lightweight and fast by specifying constraints, such as uniform interface, that when applied to a web service induce desirable properties, such as performance, scalability and modifiability. In this architectural style, data and functionality are considered resources and are accessed via URIs, that are then acted upon by using a set of simple, well-defined operations, by taking advantage of the HTTP different operands associated with CRUD operations (Create, Read, Update, Delete) and response codes [42], as briefly explained in table 2.1.

Table 2.1: HTTP Method mapping with CRUD operations regarding RESTful Architecture base usage.

HTTP Method	CRUD Operation	REST base usage
GET	READ	Get resource
POST	CREATE	Create resource
PUT	UPDATE	Update resource
DELETE	DELETE	Delete resource

Webservice APIs fostered the integration of third party applications into the development of any web platform with the premise of code reutilization, as it is now much easier to explore services that provide desirable capabilities to new and ongoing projects without the absolute necessity to reimplement already developed technologies, whether it is licensed or open data. All these factors bolster the programming ecosystem, making it richer and more sophisticated.

Webservices are interacted through data exchange formats, as XML and more recently JSON. XML stands for Extensible Markup Language and an XML document is composed of storage unities called entities that contain the relevant data properly structured [43]. Although meant to be human-readable,

XML documents are highly verbose and have machine-machine communication as the main goal. This high number of characters involved in XML documents made its usage on the web panorama increasingly undesirable, considering the web speed. JSON stands for JavaScript Object Notation, emerged in 1999 and steadily rose to be the preferred data exchange format for asynchronous browser/server interactions, as it is very focused and simple, much less verbose as comparable in figure 2.7, and written in JavaScript [43], the most prominent programming language on the web technologies, allowing easy and seamless interoperability with other JavaScript code. This trend is easily attributed to the popularity rise of the JavaScript language, to which JSON is highly correlated to. JSON is not strictly better than XML, as the later is a more powerful tool with functionalities in its core that JSON does not possess, but Nurseitov et al. concluded that "JSON is faster and uses fewer resources than its XML counterpart" [44], a desired trait in the REST architecture.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<DatabaseInventory>
  <DatabaseName>

<GlobalDatabaseName>production.cubicrace.com</
GlobalDatabaseName>
    <OracleSID>production</OracleSID>
    <Administrator EmailAlias="piyush"
Extension="6007">Piyush
Chordia</Administrator>
    <DatabaseAttributes Type="Production"
Version="9i" />
    <Comments>All new accounts need to be
approved.</Comments>
  </DatabaseName>
  <DatabaseName>

<GlobalDatabaseName>development.cubicrace.com<
/GlobalDatabaseName>
    <OracleSID>development</OracleSID>
    <Administrator EmailAlias="kalpana"
Extension="6008">Kalpana
Pagariya</Administrator>
    <DatabaseAttributes Type="Development"
Version="9i" />
  </DatabaseName>
</DatabaseInventory>
```

```json
{
   DatabaseInventory: {
      DatabaseName: [
         {
            GlobalDatabaseName: "production.cubicrace.com",
            OracleSID: "production",
            Administrator: [
               {
                  EmailAlias: "piyush",
                  Extension: "6007",
                  value: "Piyush Chordia"
               }
            ],
            DatabaseAttributes: {
               Type: "Production",
               Version: "9i"
            },
            Comments: "All new accounts need to be approved."
         },
         {
            GlobalDatabaseName: "development.cubicrace.com",
            OracleSID: "development",
            Administrator: [
               {
                  EmailAlias: "kalpana",
                  Extension: "6008",
                  value: "Kalpana Pagariya"
               }
            ],
            DatabaseAttributes: {
               Type: "Development",
               Version: "9i"
            }
         }
      ]
   }
}
```

Figure 2.7: Syntax comparison between XML and JSON [45].

When dealing with geographical data, a subset of XML is commonly used, named GML, which stands for Geographic Markup Language. GML offers standard ways to describe spatial features and their corresponding properties in terms of GML Schemata, including coordinate reference systems, geometry, topology, time, units of measure, and generalized value [46]. In order to take full use of GML capabilities and, considering it is only a data exchange format, the Open GIS Consortium (OGC) [47] launched an implementation specification for a service that allows a client to retrieve geospatial data encoded in GML, called Web Feature Service (WFS). The WFS is written in XML and uses GML to represent features, but

the database could be in any format [46], as there is no direct connection between the client requesting data and the server's database.

A comparison of the webservices provided by the studied platforms is done in table 2.2.

Table 2.2: POI providers features comparison.

Data Source	Scope	Access Type	Data exchange format
Foursquare	Worldwide	RESTful API	JSON
Facebook Graph	Worldwide	RESTful API	JSON
Nominatim	Worldwide	RESTful API	JSON/HTML/XML
Google Places	WorldWide	RESTful API	JSON/XML
DGPC	Portugal	WFS/WMS	GML/XML
DAL	Lisbon	RESTful API / WFS	CSV/JSON

2.6 Mobile Operating Systems and Mobile Development

Smartphone usage is an ever rising worldwide phenomenon, with ever increasing capabilities, functionalities and computing power, a market that grew 1.1% year over year in 2016Q3, with 363.2 million shipments, according to data from the International Data Corporation (IDC) Worldwide Quarterly Mobile Phone Tracker [48]. Therefore, development of mobile applications is becoming abundant due to the sheer size of potential customers. Currently, as observable in table 2.3, Android and iOS combined sum up to more than 99% of the Operating Systems marketshare and, as such, developing for both these platforms is fundamental.

Table 2.3: Worldwide Smartphone OS Market Share [48].

Period	Android	iOS	Windows Phone	Others
2015Q4	79.6%	18.7%	1.2%	0.5%
2016Q1	83.5%	15.4%	0.8%	0.4%
2016Q2	87.6%	11.7%	0.4%	0.3%
2016Q3	86.8%	12.5%	0.3%	0.4%

2.6.1 Development Frameworks

Currently, there are three main methods to develop applications for these platforms; native, web or hybrid development. Native development is targeted for one OS in particular and uses the language available by the platform to develop APPs upon, Java for Android and Objective C or Swift for iOS, and brings out the best performance available to the applications as they are directly tied to the OS and corresponding hardware API functionalities.

Web development for mobile is generally no different from web development in general nowadays, as developers creating web applications want to fulfill the requirements of as much audience as possible and that includes mobile since smartphones became powerful enough to make Internet browsing a common task. Using dynamic layouts, a website can be very different seen as mobile or as desktop

version in order to enhance the user experience according to the use case. Web applications can also take advantage of several web APIs [49] to interact with the devices hardware, namely camera and microphone, but it is yet limited in capabilities compared to native development. An emerging concept of web applications for mobile appeared in 2015 called Progressive Web Apps [50] and was an idea pitched and developed by Google that empowered the web applications for a mobile scenario by giving them a native look and providing some additional features like pin the application to the homescreen, which makes it feel like a ”true” application, and offline capabilities by keeping the web application in cache.

While developing a web application with mobile support, which has the benefit to automatically serve both iOS and Android as it is interpreted by the web browser, is enough to provide a service when there are no hardware requirements, native development must be done separately for each platform for maximum usage of hardware resources and performance and therefore suffers from the lack of code reutilization and different architectural approaches. This led to a rise of popularity in hybrid development, which tries to gather the advantages of both these methodologies.

Hybrid mobile development is mainly dependent on web programming technologies, as HTML, JavaScript and CSS, with native wrapper code for accessing native APIs like Camera, Contacts, File System, WIFI, etc. Hybrid mobile development tools are gaining popularity in the world due to their characteristic to compile the application source code for multiple supported OSs [51], therefore allowing the same code to be deployed in different platforms, cutting development time and required multiple programming language knowledge, a trend which Phonegap can take credit for.

Phonegap [52] was bootstrapped at the 2008 iPhoneDevCamp by Nitobi as a way to simplify cross-platform mobile development using standard web technologies. In 2012, Nitobe was acquired by the Apache Software Foundation and Phonegap became part of the open source Apache Project under the name Apache Cordova (name of the street where Nitobi offices are located), helping to reassure that the framework will remain a stable and available tool with community support. This tool’s plugins allow a web application to be deployed to a mobile platform, acting as a wrapper for native OS API calls to the sensors and hardware, as outlined in figure 2.8.

In order to make the hybrid mobile development a wholesome process, Ionic appeared as an integration solution that relies on Apache Cordova for the OS interactions and builds a mobile compliant UI on top of AngularJS, a UI framework developed by Google. Ionic [54] was developed by Drifty, first released in November 2013 and defined itself as ”Cordova with the power of AngularJS”. It is a framework packed with a generously large community, good documentation and an overall good development experience.

In 2011 appeared a different kind of mobile development with Xamarin [55], a development framework that compiles C# to ”native” Android and iOS applications owned by Microsoft. Each platform has a different compiler and the code developed is mostly reusable for cross platform deployment. It has the advantage of efficiency against hybrid applications, but it is dependent on the IDE and the time to deploy is far greater [56] than solutions like Ionic.

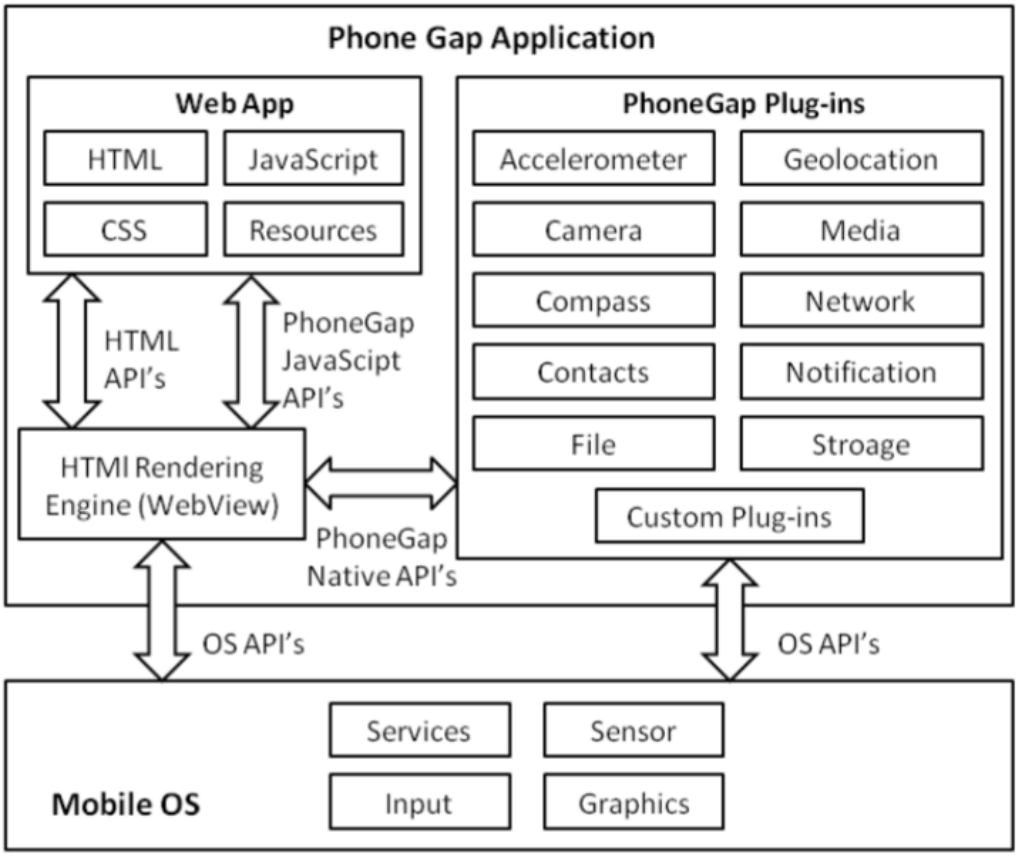

Figure 2.8: Complete schema of Phonegap architecture and interfacing amongst components [53].

Chapter 3

Fidel

The platform presented in this thesis will address identified needs from tour guides and tourism professionals that serve guided tours, either indoors or outdoors. The next section will thoroughly address the proposed solution, following up with the requirement analysis to properly implement the aforementioned value proposition. The next two sections will cover the data model tailored for the application and the general architecture.

3.1 Proposed Solution

Mobile technology in tourism still has a lot of untapped potential and as such is focus of several initiatives that try to promote active development of platforms that solve identified problems, a fact highlighted through the previous chapter. Amongst those, there is an identified need for a simple way to create customized tours and POI with a good, intuitive user experience. Also, there is an urgent and ill-explored necessity to have a platform that allows the real time sharing of tour and POI related content with customers as, after the existing platforms review, there was none to be found that could preemptively solve this need in particular.

International tourists make the majority of the tour guide customers [13] and are a good focus group to profile to extract habits and tendencies.

As such, having mobile applications that require the tourist to have an Internet connection is unfavorable and generally undesirable as it requires them to purchase additional data plans abroad. Therefore, tourists must be able to get the content shared by the tour guides without relying on Internet access.

This thesis proposes a platform composed of a three-part solution to this problem; a web application for creation of customized POI and tours, a mobile application for the tour guide to download tours from the web application and share their media with the tourists, and a mobile application for tourists to receive the broadcast media without resorting to an Internet connection. The users of the platform described in this section are mostly tour guides and tour planners, either amateurs or professionals of the tourism-industry. They will use the web application to create content, organize tours based on new or existing POI and discover existing published public tours. After that, they will take advantage of the

mobile application to download public or owned tours to the device's local storage and broadcast the desired media to the tourist applications. As such, the whole solution must contemplate the required operations in order to fulfill all the aforementioned functionalities.

3.2 Requirement Analysis

To be able to achieve the proposed solution, several decisions must be made regarding the requirements that will need to be met. First of all, there will be three applications: web application, tour guide mobile application and tourist mobile application.

The web application must be able to run as personal, community or enterprise. Personal would mean that it can be run independently from a personal server and/or personal computer, providing that all configurations necessary to make this possible are well defined and well documented. Community would mean that it can be run as a community server, available to the public and where everyone interested in using these services can do so free of charge. Enterprise would mean that the web application is capable of running on the cloud and be provided as a service to an end user, group of users or enterprise. In all of these variants, an access control privacy policy is required, meaning that it must be possible to create tours that can be either private or public for other users, and transfer or share ownership of tours and POI created on the application. The tour guide application should be available for Android and iOS, use the device local storage to save content and access the networking hardware configuration. The tourist mobile application will be a platform independent mobile tailored web application that receives content from the tour guide application.

In addition, all these projects should be developed as Open Source Software as it provides a cost benefit against close proprietary software and allows a high degree of flexibility and extensibility, assuring long term viability [57].

Having these base guidelines defined, there is now a need to establish a well defined behavior to be expected of the applications developed. As such, the following sections will cover the functional requirements to be followed.

3.2.1 Web Application Functional Requirements

In the web application, the creation of a POI starts with searching a name and/or a location to get a geolocation of the desired point. Then, optionally searches the nearby area for possible matches from external providers to not only link it to already existing data but also as a base guideline for description, address and media. Finally, the user inserts all the additional information he desires in the POI description.

The created POI must be reusable for other tours, either by the same user, for a group of authorized users, or public in general, if flagged appropriately.

To create a tour, the user must specify the name and description, along with a list of POI and other optional media; All content must be editable and deletable from the creator, as all content is owned by

himself.

As far as the web application goes, the requirements that must be fulfilled in its implementation are as following:

1. A potential user must be able to register on the application and then login with his credentials;

   The user must be a well defined and established entity reachable by the relevant audience. As such, an aspiring user must be able to register himself/herself in the application after providing his/her full name, email address and password. To login later, email and password must be provided to the application in order to authenticate himself/herself.

2. A POI is defined by a Name, Latitude, Longitude, Address, Description, Owner and Access List;

   To define a POI from a tour guide perspective, these are the most relevant fields that should exist. Access List will be composed of all users that have access to edit the POI information.

3. A POI should support the storage of multiple photos or reference to photos;

   When sharing media on a tour, photos and images in general come across as the single most desirable feature. Adding this media to the platform must be available via direct upload or URL referencing for media already hosted on external applications.

4. A POI should be able to store references for multiple POI Information Providers;

   Diverse external POI providers will be a good baseline for the creation of content from curated sources that may be used for enhancing POI descriptions, correctly map addresses and coordinate pairs.

5. A POI, when created, will be part of the user personal POI collection;

   A user must be able to keep track of his created and owned POI not only to reuse them on different tours but also to edit their content, like changing the description and adding or removing media.

6. POI can be added or removed from tours;

   Tours cannot be considered fixed entities and offer the adequate elasticity to be able to add or remove POI from its constitution. By removing a POI from a tour it must not be deleted from the database, as that POI can be reused by the owner for other tours.

7. The same POI can be part of several tours;

   Adding a POI to several tours must be a feature present in the application as they may be several different tours that use that POI and forcing the creation of a duplicate with the same information is an undesirable overhead.

8. Tours can be private or public, and this parameter must be editable;

   The decision to make tours public or private must not be final and, therefore, changeable at any time. For example, private tours can be made public in order to promote the sharing of content with other tour guides using the platform, just as public tours can be made private by the owner and edited to further personalize its content for business practice.

9. Tours can be co-owned by other users, when the owner gives them access to it;

   Tour guides may be freelancers or part of established companies. As such, it is necessary to support the concept of ownership to a group of individuals that are part of a group responsible to serve those tours and cooperatively add and edit its content.

10. POI from public tours or accessible tours can be copied to the user collection;

    Public tours and owned/co-owned tours are composed from POI with the same attributes. POI from a public tour are effectively public to any user. Therefore, the web application must support a feature that allows to add a copy of an accessible but not owned POI to a user POI collection in order to integrate it later on on different tours. In this case, a copy must be done essentially for two reasons: Firstly, this allows the copying user to keep all the data on the POI at the time of the copy, becoming its owner at that moment and therefore be able to enjoy all POI related features, namely making it private and edit its content. Secondly, by doing so, any interaction with the copied POI does not affect the original and, as such, will not disrupt other tours in which that POI is being used.

11. All POI and tours must be editable and removable;

    Following the same principles as topic five, tours must also be editable and deletable, and POI must be permanently deletable if required. Deleting a POI or removing it from a tour are two distinct operations, as a POI removed from a tour is not deleted as it is still part of a user collection and a deleted POI is completely erased from the application.

12. A user must be able to list POI from Information Providers in the nearby area of his POI address, in order to find the best matches to cross-reference;

    Sometimes it will not be possible to get exact matches from external providers to a given POI name. In order to solve this, a user must be able to check the nearby area of an input location to get the best matches from the POI providers to analyze and properly link if a satisfying match is found.

13. A RESTful API webservice must be provided, allowing external authorized entities to interact with the application.

    As this web application will effectively serve as a backbone for the tour guide mobile application, there must be a well defined and implemented RESTful API to allow proper communication between these components. Moreover, it will serve as a gateway to allow other external applications to interact with the web application and all its defined functionalities, effectively serving as a tour/POI provider webservice.

3.2.2 Tour Guide Mobile Application Functional Requirements

In the tour guide tailored mobile application, a user must be able to download tours created in the web application, effectively storing them in the device's local storage, enable broadcasting from the device

and share selected media on demand.

For this application, the requirements that must be fulfilled in its implementation are as following:

1. User authenticates with the web application;

   Using the previously mentioned RESTful API, a user must be able to properly authenticate himself/herself with the web application using an Internet connection.

2. Owned and co-owned Tours are browsable and downloadable from the web application into the mobile local storage;

   After authentication, a user must be able to browse his/her tours and download them to the device. Doing so will allow the inspection of its content even without an Internet connection, a fact bound to happen when sharing media without resorting to it.

3. Downloaded tours must be erasable from the device;

   When presented with a collection of downloaded tours, a user must have the power to delete any of them from the device without affecting the stored content on the web application.

4. Mobile network is configurable in order to serve content to the tourist application;

   Before being able to share the media content with tourists, this mobile application must provide a configuration option to set up the mobile network. Doing so will present all the required information to properly start the broadcast

5. Media from tours and POI is selected from the downloaded tours and broadcast to all tourist applications.

   After network configuration, a user must be able to easily share the desired media on demand to all tourists that are actively listening.

3.2.3 Tourist Mobile Application Functional Requirements

The tourist mobile application should be as simple and effortless as possible. The device running this application will be receiving the media shared by the tour guide even without Internet access if not available.

For the tourist mobile application, the requirements that must be met are as listed.

1. Configure network to connect with the tour guide mobile application broadcast service;

   In this step, the tourist must apply the configurations necessary to his device to promptly be able to receive the media from the tour guide without the use of Internet.

2. Listen for incoming media.

   After the initial configuration, no more action should be required by the tourist that now has his/her device actively listening for incoming media.

3.3 Data Model

A data model is the result of connecting multiple data entities in order to satisfy the imposed requirements in a usable and implementable way, while Data Entities are implementable units of Information. Having completed the requirement analysis phase and therefore understanding what is required as functional perspective, transforming these requirements into a data model represents the transformation of a functional specification into a technical one.

The following data entities represent one possible way of fulfilling the requirements for all the applications proposed.

1. User: Represents a user of the application, which can then be an owner of POI and tours;

   A user entity must be composed of an unique id, full name, email and password fields. This entity will be related with the tours and POI in order to establish the ownership and access.

2. POI: Basic structure for a Point of Interest;

   According with the previous functional requirements establishment and having the W3C proposal [4] in consideration, a POI shall have a unique id, name, description, coordinate pair, address, a boolean flag to check whether is marked as a public POI or not (in that case, it will be private), owner, an access list with all the users with permissions to modify and view it in case it is private, a list of media associated with the poi and, finally, the list of external providers entities linked.

3. Tour: Basic structure for a tour;

   According with the previous functional requirements establishment, a tour shall have a unique id, title, description, a boolean flag to check whether is marked as a public tour or not (in that case, it will be private), owner, an access list with all the users with permissions to modify and view it in case it is private, a list of media associated with the tour and, finally, the list of POI that compose it.

4. POIMedia: Represents the media for a POI;

   This entity will be composed of a unique id, title, description and an URL to the location where the media is stored. If the media is directly uploaded into the application, this URL will be the internal location of the uploaded media

5. TourMedia: Represents the media for a tour.

   Follows the same principle as the POIMedia entity, but for tours.

Using the data entities defined above, it's possible to define a fairly simple data model that supports all the desired functionalities without overhead. Note that, regarding POI, many use policies of external providers do not allow the storage of information provided by them [35, 37] and, as such, only an ID is stored to retrieve the desired information when requested by the user, and thus complying with the terms of use.

Having defined the data entities, figure 3.1 represents the UML diagram for the application.

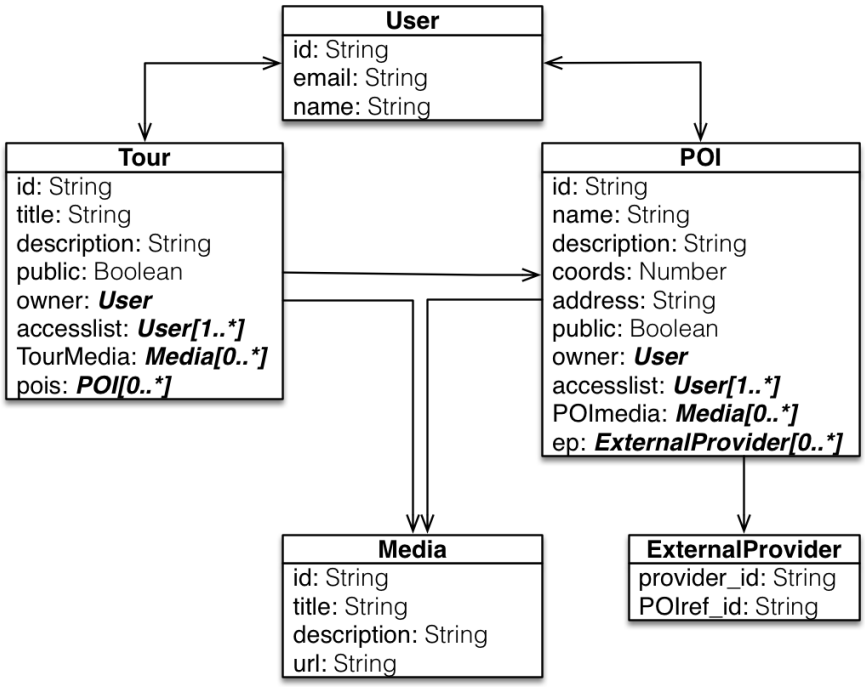

Figure 3.1: Application Data Model.

3.4 Platform Architecture

In order to fulfill the several functional requirements aforementioned in the previous section, a platform architecture must be defined. The platform architecture shall consist in a web server with RESTful APIs that handle the application logic and serve a well defined interface for external application interaction, a Database system to store all the entities and data, a frontend component to run on the web application client-side to serve as a pleasant and intuitive gateway to the underline application logic, the tour guide mobile application connecting to the webserver via the defined APIs and the mobile web application for the tourists directly connect to the tour guide mobile application. Figure 3.2 shows the base architecture that supports the tailored data model and established functional requirements.

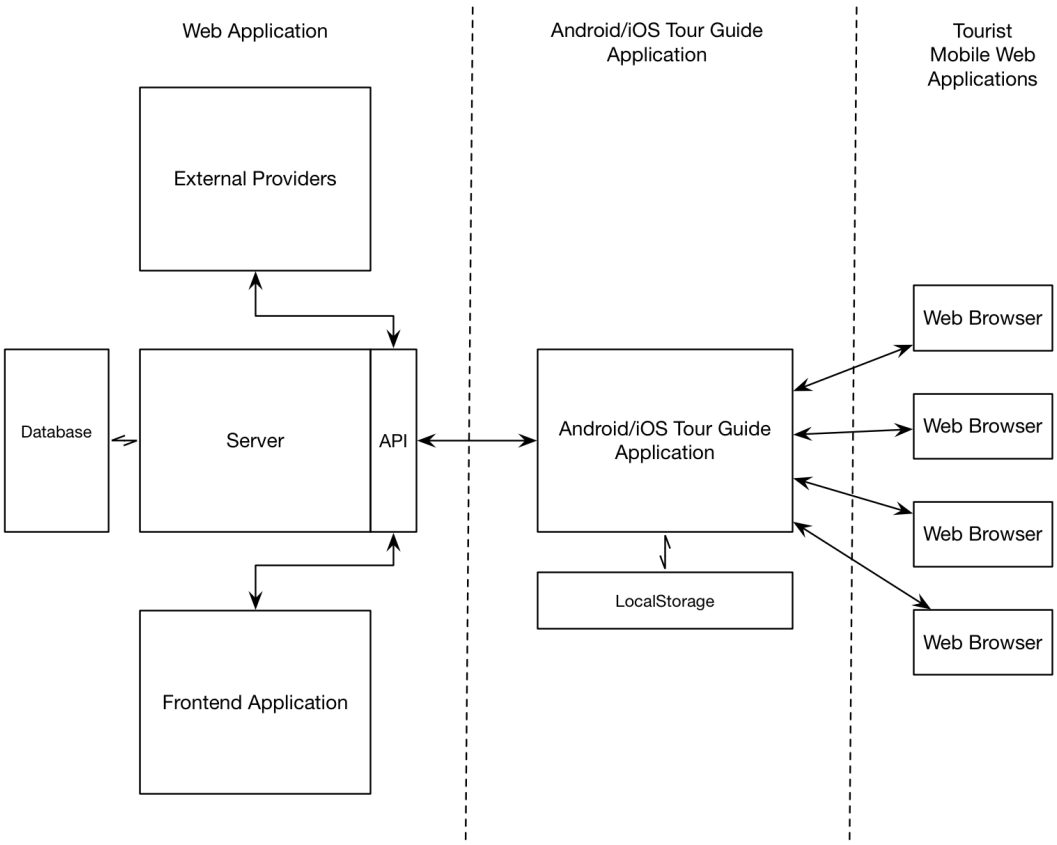

Figure 3.2: Base platform architecture.

Chapter 4

Implementation

Now that the functional requirements, data models and platform architecture are already drafted, it is necessary to choose the technologies that are going to be implemented in order to make the end product and provide information about additional networking based configurations required to the proper functionality of all components.

4.1 Chosen Technologies

This section will cover the frameworks adopted for the development of the web application and both mobile applications, along with the reasoning behind the choices made, the database management system and the map and external POI providers picked from the reviewed ones.

4.1.1 Web application

In software engineering terms, frontend and backend refers to the separation of concern between the presentation layer, commonly associated with the term ”client side”, and the data access layer, commonly associated as ”server side”. The whole web application, as defined in the platform architecture, is composed by a frontend solution, a backend solution, a database management system and map and external POI providers.

Frontend Development

In the web ecosystem, the presentation layer has three key components: HTML for proper structuring of the static layout of the presented page, CSS for the styling of the presented objects and JavaScript for dynamically change the presented layout from client side events. Although seemingly simple, there are countless frameworks that present different programming paradigms and different rationales but, in the end, they are composed of these three components. Of these three, JavaScript is the one with the highest amount of library implementations that drastically change the way the code is structured and the application logic is layered. While pure JavaScript is still widely used, there several highly

popular JavaScript based libraries that take a higher level approach, namely AngularJS, React, Ember and jQuery. Angular is a Model-View-Controller framework, which is a software architectural pattern for implementing user interfaces that divides the logic into three interconnected parts in order to separate internal representations of information from the ways that information is presented to and accepted from the user, creating a cycle explained in figure 4.1.

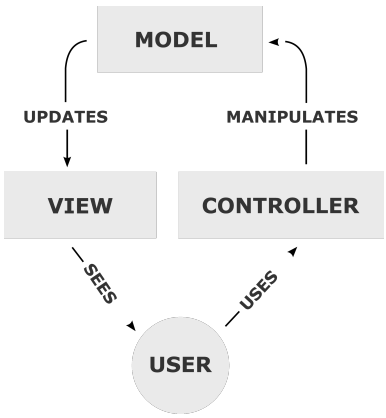

Figure 4.1: MVC architecture.

After carefully reviewing existing frontend technologies, it became apparent that there is not a best choice, and adoption of specific technologies is based on preference. As such,

Angular is also a Single Page Application (SPA), a paradigm that states that the server sends a single page to the client side and further interactions with the backend are done via the established APIs, providing a user experience with little page refreshes for dynamic presentation and a minimum of redundant information exchange.

Backend Development

The data access layer has several technologies that can be used to not only serve web pages but also to provide webservices and RESTful APIs and interact with the database management system, whether it is local or external.

Historically speaking, Apache/PHP combination has been prevalent and is still widely adopted as solid and stable solution. However, with the ever increasing demand in network traffic, it is starting to unveil some aging in these technologies. Nginx emerged as a gateway to serve HTML and PHP files that outperformed Apache by over 2.5 times faster for Input/Output operations [58] and then, NodeJS appeared and outperformed both of these solutions [59, 58]. According to Chaniotis et al., ”NodeJS offers client-server development integration, aiding code reusability in web applications, and is the perfect tool for developing fast, scalable network applications”.

NodeJS keeps growing in both usage and functionalities [60] and currently beats all other backend solutions in the benchmarks done by Lei et al. [59]. It has an enormous community support, is currently highly adopted, even by massive technology corporate solutions such as Paypal, and has numerous libraries that add functionalities to its core easily discoverable via the Node Package Manager (NPM). Amongst them, there is one called ExpressJS that acts as a fast, unopinionated, minimalist web

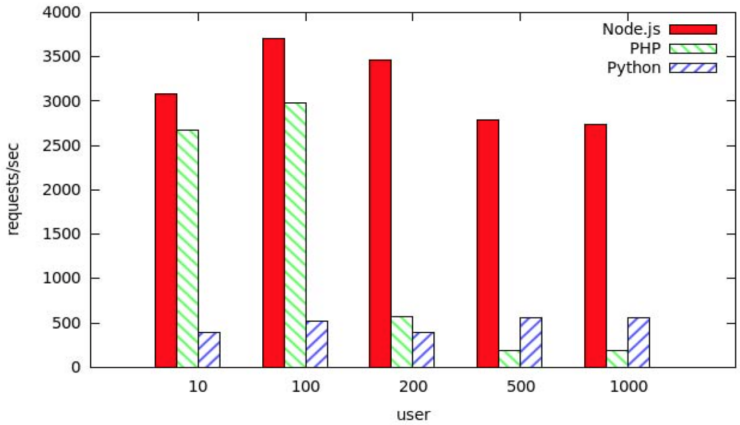

Figure 4.2: Results for ”Hello World” mean time requests [59].

framework that is subject to twelve million downloads per month. ExpressJS provides all the core web framework functionalities with a minimal amount of code required.

As an effort to provide an Open Source Software solution and using the latest technologies, NodeJS and ExpressJS were used as the backend development framework in favor of proprietary software or an Apache/PHP solution. Asides from the benchmarking, the JavaScript based language makes it easier for singlehandedly develop both frontend and backend as they end up being the same programming language, softening the learning curve required to fully develop the proposed platform.

Database Management System

Relational databases are widely used in most of the applications and they have good performance when they handle a limited amount of data. To handle a huge volume of data like Internet, multimedia and social media the use of traditional relational databases is inefficient. To overcome this problem the “NO SQL” term was introduced [61]. Non relational databases have been gathering followers due to the flexibility it provides to applications, as it requires no tables and allows unstructured, vastly diverse documents. In addition, benchmarks performed by Győrödi et al. [61] show that MongoDB, a noSQL database management system, outperforms mySQL in all four basic operations: Insert, Read, Update and Delete.

If there is an already developed and working application that uses an SQL database, modifying it to use a solution like MongoDB is far from easy, as the architecture paradigm is completely different. In these cases, sticking with the already implemented databases is advised. However, presented with a need to start an application from the ground, implementing a noSQL solution will bring the best performance available.

Once again following the desire to use cutting edge open source technology, MongoDB was used as the last part of the stack. All documents are stored as JSON objects [62] and interaction with the NodeJS server is simplified with the usage of the mongoose module, that acts as middleware between the server logic and the database logic [63].

External POI Providers

After carefully reviewing appropriate providers to interact with the application, some of them were integrated. According to the scope and data exchange format established in chapter 2, Foursquare and DGPC were chosen as they consist on two different approaches that set the required logic to implement all the reviewed ones.

The application interacts with the external providers via their webservices. A request is send to their RESTful API and server-side logic is implemented in order to properly handle the responses from each of them. Foursquare's response [35] is via JSON and requires little handling to be well integrated into the application. DGPC data, however, is transmitted via WFS [37] and as such, needs to be parsed locally into a more suitable format before being smoothly integrated onto the application logic.

Map Services

The review of the existing map services highlighted advantages and disadvantages of each one of them. While Google Maps [29] represent the most widely adopted service for mapping purposes, its usage policy and request limit rate could prove troublesome at a later stage. These factors, in addition to the desired Open Source and community driven development of OpenStreetMaps [30], led us to the decision to implement OpenStreetMaps as the platform mapping service. Following the same principles of this choice, Nominatim was used as the geocoding tool for the same reasons.

To integrate the chosen map service with the presentation logic, an open source map library called LeafletJS was used. LeafletJS serves as a container for map services to be presented and interacted upon.

4.1.2 Tour Guide Mobile Application

The mobile application was developed for both iOS and Android and all the architectural decisions and implementations will be covered in the following sections

Mobile Development Framework

Instead of developing a native application, considering the audience and the project requirements, hybrid mobile development was chosen in favor of native. The biggest advantage is that most of the code is transpiled for both required platforms simultaneously, substantially cutting the time to develop. In this case, Ionic was chosen as a good fit due to its maturity and having AngularJS as a base for the UI, a JavaScript Framework already used in the web application and therefore, having a proficiency bonus. Ionic provides the ability to develop a mobile application as if it was a web application that is then rendered as webviews on the mobile. It relies on Cordova plugins that serve as JavaScript wrappers to native functions of mobile Operating Systems; In Android it will run the corresponding Java functions, in iOS it will run the corresponding Objective-C/Swift functions. Several mobile development frameworks rely on Apache Cordova, such as React Native and OnsenUI and its popularity is gaining even more

traction due to the increased performance of mobile devices and shorter development times. In terms of performance, native development still has the advantage but the gap is closing as these tools evolve with all the popularity and support they have experienced, and hybrid development has the advantage of being a solution with a shorter necessary development time to reach an end product.

All the necessary device hardware functionalities were covered by Open Source plugins developed by the enthusiastic Cordova community. One of the implemented plugins, Cordova NanoHTTPD, provides nano web server functionalities in order to make the broadcast a reality. The author of this thesis had a pull request merged on this GitHub project that fixed identified library problems that made the application crash unexpectedly.

4.1.3 Tourist Mobile Application

The tourist Mobile Application is the gateway to access the shared media by the tour guide. Considering that, in this project, it must only receive media, the desired functionalities were achieved by developing a mobile web application. This means it is usable without any additional installation, as every smartphone already has a web browser installed by default, further benefiting the desire to not require any kind of Internet access by the tourists.

4.2 Implemented Architecture

The implemented platform is equipped with the following basic operations provided

1. CRUD operations on POI - User can create, view, update and delete POI, although update and delete only if authorized;

2. CRUD operations on Tours - User can create, view, update and delete tours, although update and delete only if authorized;

3. CRUD operations on Tour and POI Media - User can create, view, update and delete media. Adding media can be done by either uploading or referencing via URL;

4. Add POI to collection - User can add POI from public or accessible tours to his personal POI collection, by creating a copy of it;

5. Add user to access list - User can add other users to have access to his tours;

Figure **??** demonstrates the implemented web application with all the technology choices considered. The next subsections will cover each application specifications with more detail.

4.2.1 Web application

The web application implementation choices gave order to the following specifications:

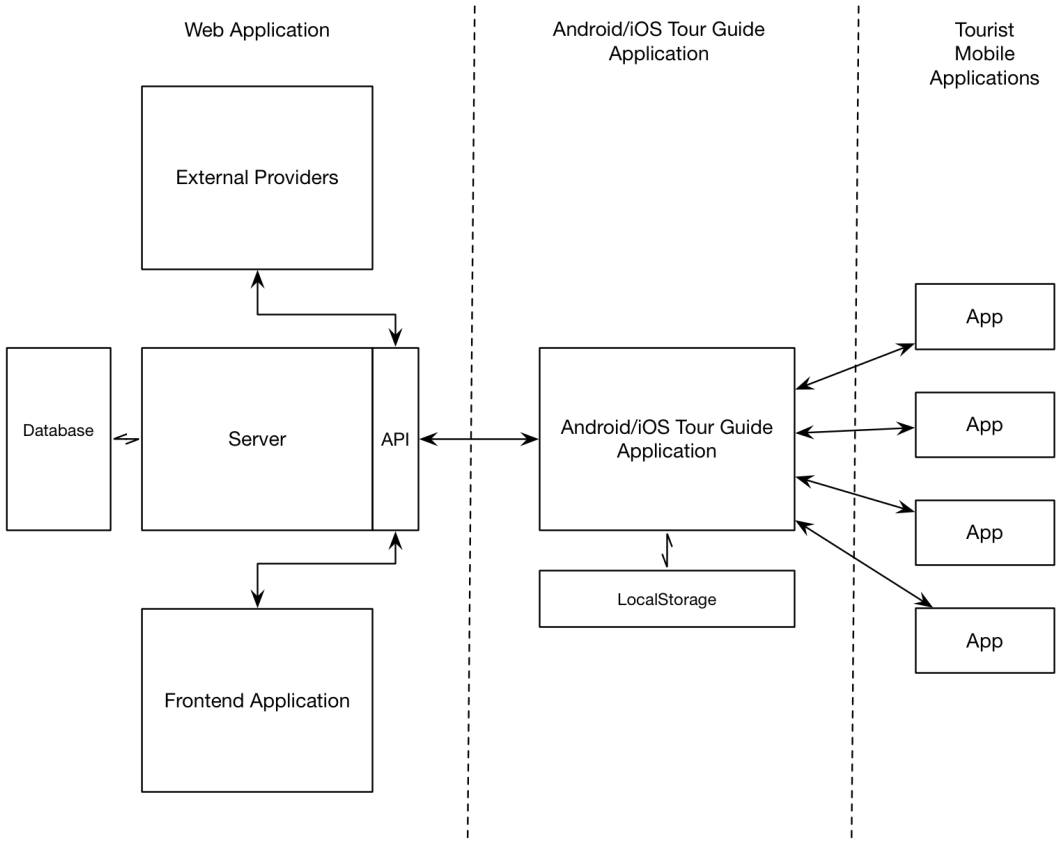

Figure 4.3: Detailed implementation architecture.

1. Clients use a compatible web browser to access the web application (Google Chrome, Mozilla Firefox, Safari, Opera, Microsoft Edge);

2. The web server serves the Single Page Application to the client and provides the webservices API routes in parallel;

3. The API routes are used whenever any request to handle database data is made, either internal or external;

4. The SPA has routes, views, controllers and services to act as an interface between the client and the server;

4.2.2 Tour Guide Mobile Application

The tour guide mobile application has the following core features:

1. Tour Guides Log in the Application to authenticate themselves with the web application;

2. After authenticated, they can browse their collection of tours and download them to local storage;

3. Having downloaded tours, they can use the application without internet connection and start broadcasting to other devices;

4. As soon as broadcasting is enabled, network configuration will be prompted to the user to turn on the mobile hotspot, wifi or mobile data, pointing the generated IP address for the broadcasting server to be supplied to the tourists. At this moment, the tour guide application is effectively serving as a dedicated web server accessible via LAN or internet;

5. While browsing the tours and while broadcasting is enabled, media can be selectively picked to be shared and, if so, is broadcast to all listening devices.

4.2.3 Tourist Mobile Application

Assuming the installation of a web browser as an implemented architecture, the specifications for usage are as the following:

1. Turn on the device's wifi and connect to the tour guide created hotspot;

2. Open web browser and point to the IP provided by the tour guide;

3. Tourist is now actively listening for new media to be shared by the tour guide.

The next chapter will provide the functional analysis and properly compare the provided with the required features.

Chapter 5

Results

In this chapter, the functional evaluation of the developed platform will be presented and power consumption will be analyzed.

5.1 Functional evaluation

This section presents the interfaces of the platform developed and demonstrate how they adhere with the previously established requirements. Continuing with the structured separation of concern, the web, tour guide and tourist applications are split into three different subsections that will properly delve into the topics.

The coverage of the functionalities may not follow the same order as presented in the chapter four in favor of the order in which they are executed by a user, enabling a more fluid sequence of events.

5.1.1 Web Application Functional Evaluation

The web application was deployed into a server at INESC-ID and is currently accessible from any browser at `http://146.193.41.162/fidel`, provided the existence of an Internet connection. Greeted with a quick explanation of the application's functionalities in the home page, the visitor can now proceed to browse existing public tours or authenticate himself.

Authentication

A Login button is accessible at all times from the navigation bar at the top to an unauthenticated user, that will guide him/her to the authentication forms. When clicked, the login form is presented containing a link to the register form presented in figure 5.1 in case of a first time user. After either logging in or registering, the user is authenticated into the application and the login button is traded for the user's full name and additional, now revealed, functionalities.

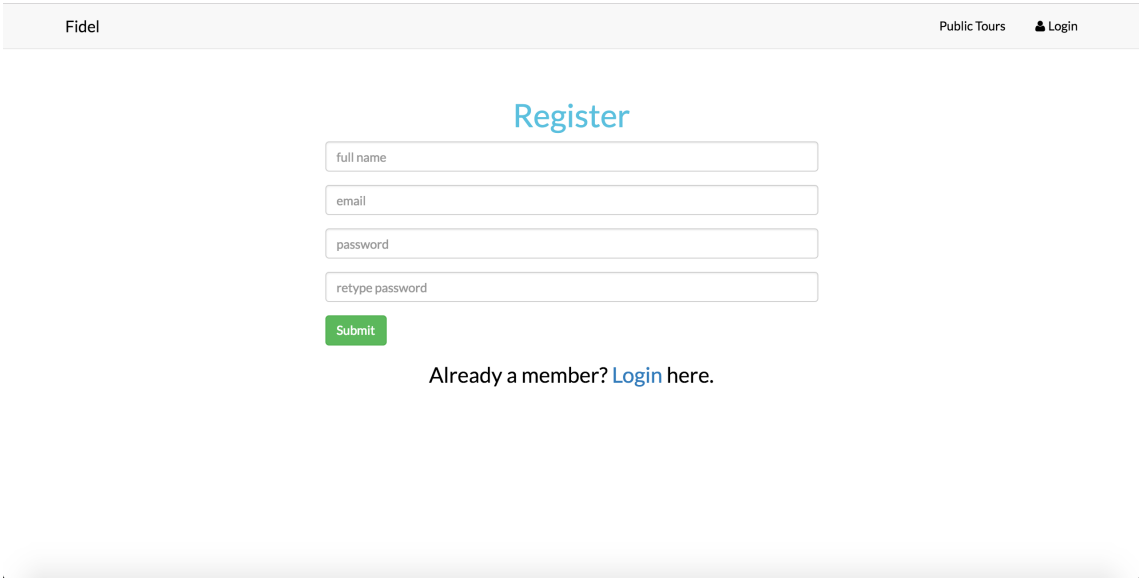

Figure 5.1: Register form of the FIDEL web application.

Tour creation

One of the revealed functionalities leads to the *Mytours* page, where a new user will only be able to create a tour. Clicking on that single button leads to the form demonstrated on figure 5.2 with placeholder text on the boxes indicating the purpose of each field, title and description respectively.

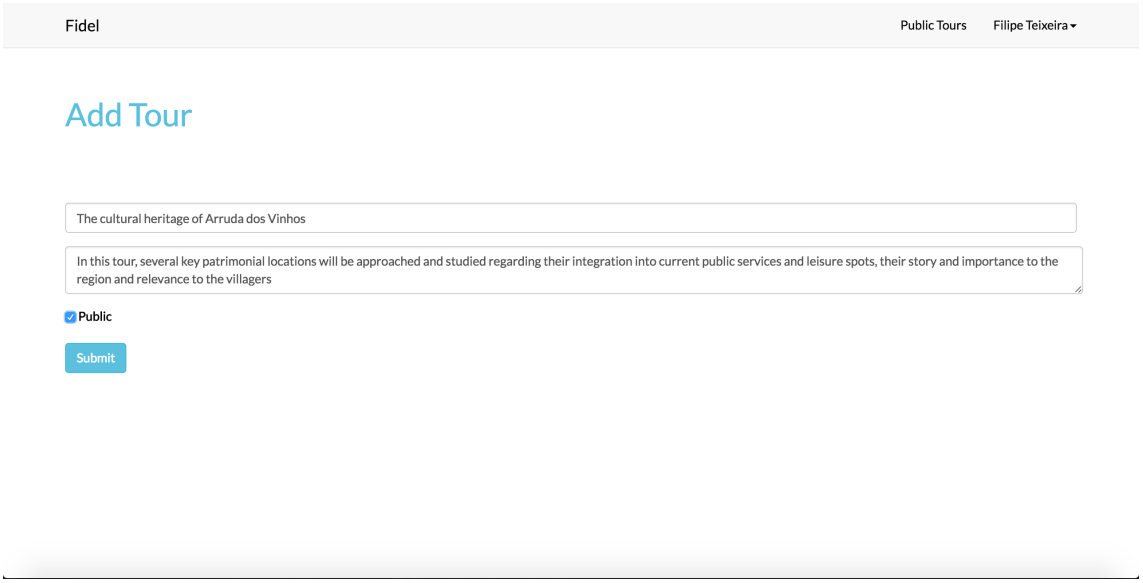

Figure 5.2: Tour creation form of the FIDEL web application.

After submitting, the user will be redirected to the newly created tour and all its content, still minimal for now, and a range of possible interactions with the tour content. At this point, adding tour media, new POI to the tour, edit its content or adding another user of FIDEL as a collaborator are presented as options, as seen in figure 5.3.

Figure 5.3: Tour options.

POI creation

Adding a new POI to the created tour can be done from the form presented in figure 5.4 following the link from one of the previously mentioned options. In this form, a user can either choose an already existing POI from his/her collection or create a new one. When inserting the required information, a location on the map can be achieved from searching by given name or address, clicking on the respective magnifier. Although not presented to the user, the system is keeping track of the coordinate pair of the searched locations and save this information when submitted. After this step, a search in the vicinity is available to locate related POI in the surrounding map area from either Foursquare, DGPC Protected Patrimony (PP) or DGPC Non-Protected Patrimony (PNP). Then, favorable matches can be linked with the POI being created, a process finalized when clicking the submit button. At this stage, like in the case of the tour creation, there is not the possibility to add the media. Media addition is done at a later stage.

Media addition

The process of adding media to a tour or to a POI is essentially the same, but from different starting points. As tours have a set of options evidenced in figure 5.5, POI too have an *Add Media* option. When clicked, the user will be presented with the modal of figure 5.5 that accepts drag and drop upload of media, simple media browse upload or external URL reference. In the last case, a title for the referenced media is required, in all other cases, the uploaded media's name will be the same as the original filename.

Tour, POI and Media editing

At this point, a new user would already been presented with ways to authenticate, create tours and POI, and add media to each of them. Furthermore, giving collaboration permissions to other users and edition

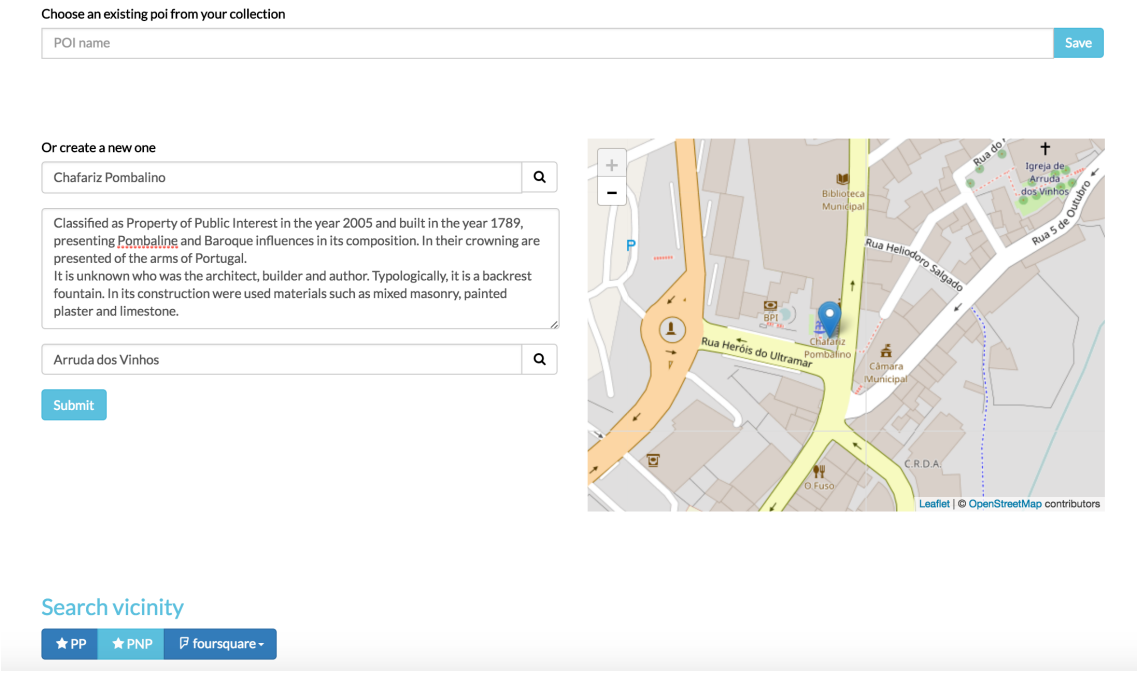

Figure 5.4: POI creation.

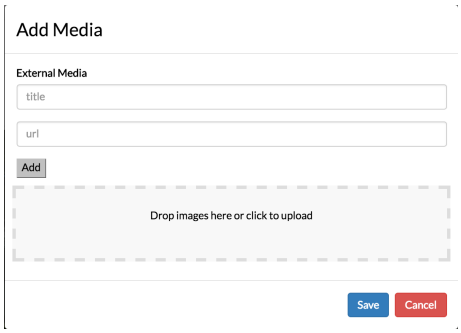

Figure 5.5: Media upload modal.

of tours and POI were displayed as options. The main differences between the editing and creation of POI and tours are the already filled information in the respective fields and the display of the media gallery associated, that can be reordered or removed from this scenario. Also, in the case of tours, the list of POI is provided so they can be removed or reordered, and a new button with the option to delete the tour is presented next to the *Submit* button, as seen in a zoomed out view of the page in figure 5.6.

To edit media, simply click the target image from the tour or POI screen and the *edit* button under options will lead to the same functionality.

Tour permissions and POI reusability

When a user creates tours or POI, they are automatically added to his/her tour collection (*My Tours*) and POI collection (*POI Collection*). The tour collection aggregates the tours that will be accessible from the mobile application and helps tour guides keep track of their created content. POI in the POI collection will be ”owned” by the user and can be reused in other tours, either existing or new, as hinted on the first input field of figure 5.4. The POI in this collection (figure 5.7) will also link to a slightly different view that

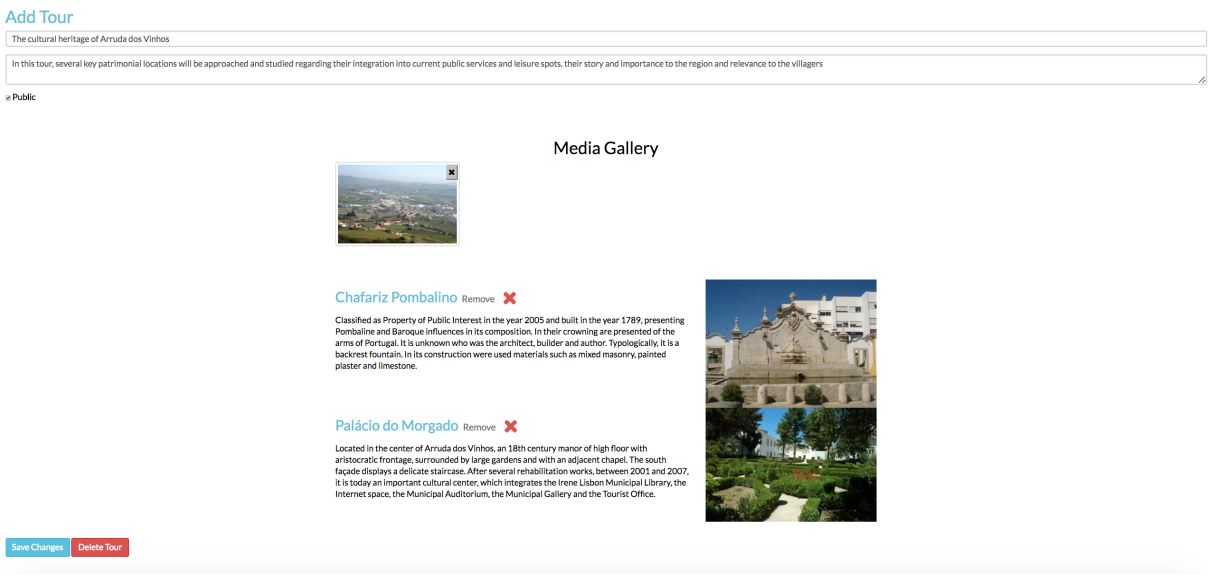

Figure 5.6: Edit tour view (zoomed out for layout exemplification).

is complemented with the information of which tours that POI is currently enrolled in addition to all the previously presented information.

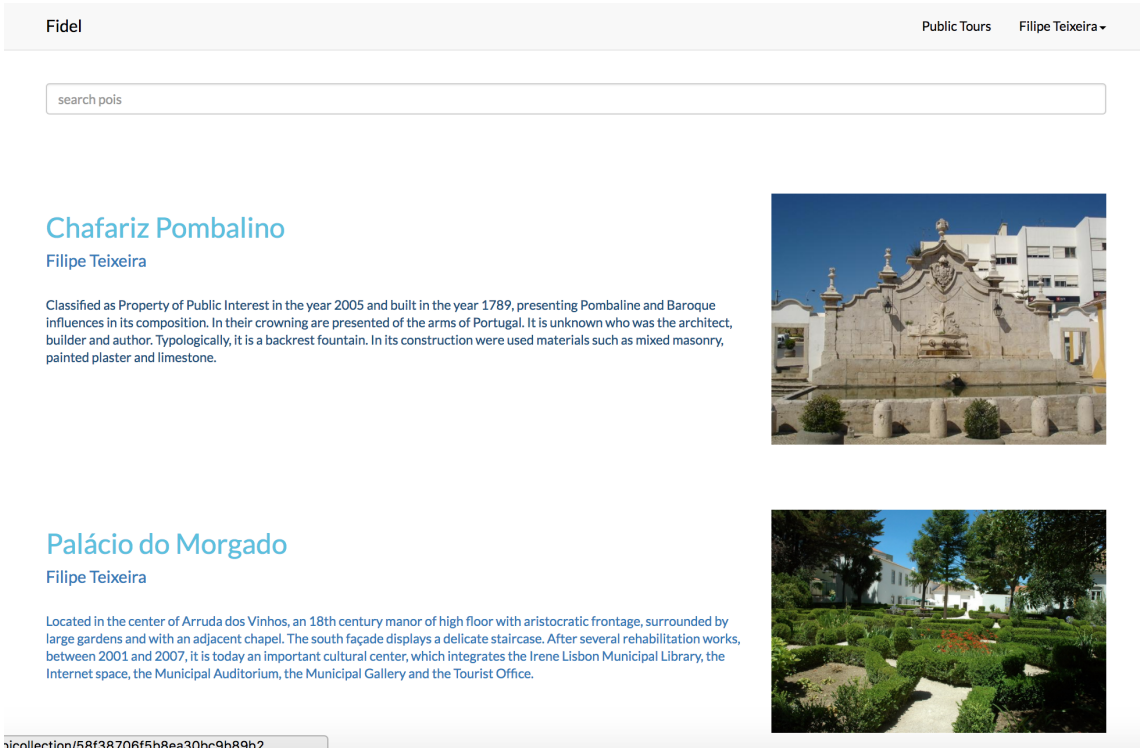

Figure 5.7: POI Collection.

API webservice

All the aforementioned operations were achieved by consuming the webservices provided by the back-end, which is accessible to external applications, namely the tour guide application. A full list of all

the methods, input and output parameters will be available under `http://146.193.41.162/fidel/api/documentation`.

5.1.2 Tour Guide Mobile Application Functional Evaluation

The experimental tour guide mobile application developed, not available on the Google App Store, is downloadable from the FIDEL home page. The main focus of this application was making sure it met the functional requirements provided. Consequently, the presentation layer that will be demonstrated still has some tweaking ahead to make up for a pleasant user experience.

After downloading, installing and starting the application, the user will be presented with four tabs:

1. Homepage tab;

   This tab presents the user with a greeting message, serving as a landing page.

2. Settings Tab;

   From this section, the user can enable the broadcasting service, seen in figure 5.8, that translates in starting the web server functionality of this application. Toggling it on will provide a popup message stating the IP address from which the service is reachable, to be provided to the tourists. To be reachable, the tour guide must either enable the hotspot functionality of Android/iOS operating system, connect to a LAN switch or the Internet.

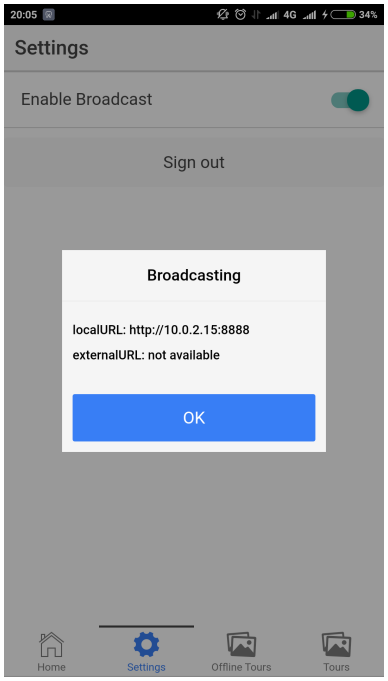

Figure 5.8: Turning on the broadcasting service.

3. Online Tours Tab;

   When navigating to this tab, the user will be prompted to login to the FIDEL platform if not already authenticated, requiring internet connection. When authenticated, the user will be presented with his/her created tours from the website, browsable and downloadable.

4. Offline Tours Tab.

   In this tab, the user will have access to all downloaded tours and all media associated with them and their corresponding POI that can be shared on demand after starting the broadcast service, demonstrated in figure 5.9. In addition, tours can be deleted from the local storage.

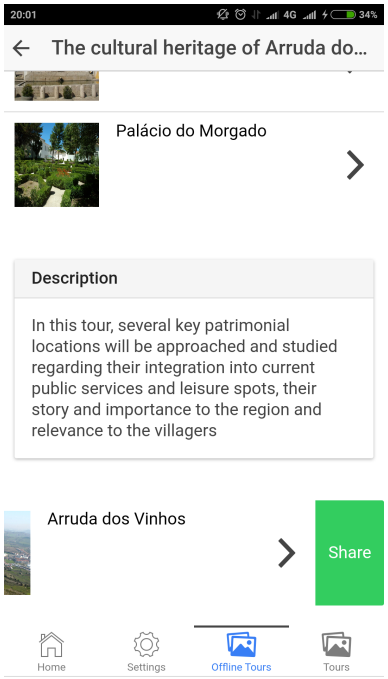

Figure 5.9: Sharing media.

These four tabs fulfill all the required functionalities proposed in chapter three.

5.1.3 Tourist Mobile Application Functional Evaluation

When the broadcast service is started by the previously mentioned solution, the tour guide is presented with the IP address of the server and share it with its audience. A tourist connects his/her device to the hotspot created by the tour guide or, if any of the other cases, Local Area Network or Internet, and navigates to the given IP address to be instantly provided with a web application designed to receive media from the *share* functionality of the tour guide application. Effectively, the tourist device is now actively polling the mobile web server for updated content and fetches it when available.

No further interaction is required at this point, even though the tourist can zoom in, inspect the received media and save it to the device like from any other Internet page.

5.2 Mobile application power consumption

When in a tour, either indoors or outdoors, the tour guide and the tourists are usually moving around different places. For a tour guide, the battery life of his/her device should be enough to cover the whole duration of the tour, at least, using the application developed. Therefore, studying the power

consumption of the application in different use cases is important to estimate how long can it be used without having to recharge the device.

To run the tests, it was used a Xiaomi Redmi Note 2 with Android 5.0.2, 2GB RAM, Mediatek MT6795 Helio X10 Octa-core 2.0 GHz Cortex-A53 CPU and a Removable Li-Po 3060 mAh battery.

This experience was done with the help of Trepn Profiler [64], an on-target power and performance profiling application for mobile applications developed by Qualcomm and positively reviewed by Pandikumar and Sumathi [65], who highlighted its capabilities to measure energy consumption from designated applications. All considered tests were run for approximately ten minutes each, with screen luminosity at a minimum, and all other applications and services disabled. Their results were validated through inspection of the output graphical data, that confirmed a steady average power consumption even if the run time was higher.

The test cases to evaluate power consumption and compare it to other applications presented in figure 5.10 were designed as following following:

1. Tour Guide Application with wifi turned on and no clients (no tourist applications);
2. Tour Guide Application with hotspot functionality turned on and no clients;
3. Tour Guide Application with wifi turned on and one client;
4. Tour Guide Application with hotspot functionality turned on and one client;
5. Tour Guide Application with wifi turned on and five clients;
6. Tour Guide Application with hotspot functionality turned on and five clients;
7. Google Chrome with wifi turned on;
8. Hearthstone, a popular mobile game developed by Blizzard, with wifi turned on.

The test results concluded that using the tour guide application consumes roughly the same in either scenario, and that consumption is much lower than Chrome and, especially Hearthstone, the mobile game that was introduced in the test for scaling purposes as it is a heavily taxing application. The values experimentally obtained prove that intensive usage of the tour guide application will not be a significant problem for the battery lifetime. Different devices are bound to have different battery autonomies, therefore this data only serves as an informed indicator.

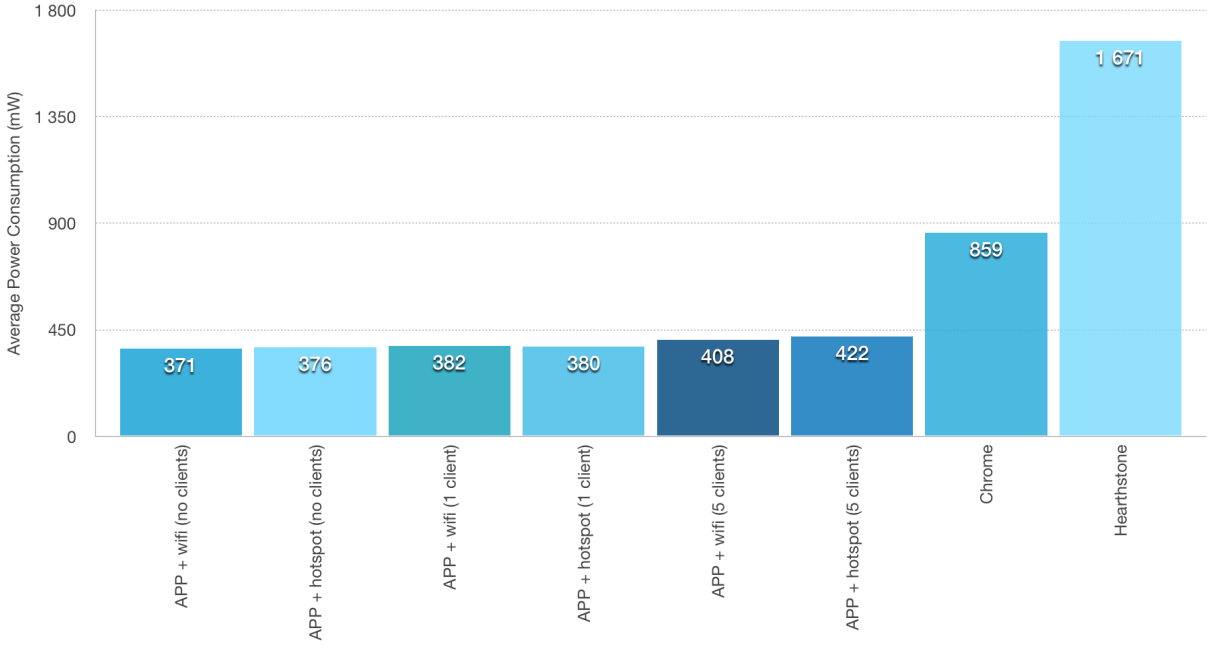

Figure 5.10: Application Data Model.

Chapter 6

Conclusions

This document started with the description of the current state of tourism, the tourism industry and its relevance and the issues felt by tour guides and their customers. The goal of this thesis was to produce a data model and a system architecture that supports the development of a platform that would address ill-resolved issues and properly implement it. In order to fulfill this mission, research has been made regarding existing applications and services, gathering the requirements needed to be met. Based on this step, a data model was defined that would serve the intended purpose and a generic and flexible architecture was presented that would serve as a platform for the creation of custom, curated tours and POI, with support for sharing media content on demand with a mobile application without requiring Internet connectivity. The platform was then developed and met the requirements analysis defined, resulting in a robust web application and a lightweight mobile application.

This work takes novel approach into solving a necessity in the delivery of technologically-enhanced experiences by tour guides to provide the best possible experiences to their customers.

6.1 Achievements

The project developed provides a centralized platform for creation of tours and POI, with aggregation tools from external providers and delivering this content to third party applications via webservices which was urgently needed by tour guides and developers for tourism applications. The mobile applications created as a result of having this service not only demonstrate its power, but also implement the major feature of sharing content without requiring an Internet connection, a novel approach to technologically enhance walking tours experience. In addition to all the functionalities provided, the greatest achievement of this ambitious work is setting the foundation for a lot of derivative products and modular extensibility to either the web application and the mobile applications developed.

6.2 Future Work

This project sets a base for numerous possible extensions. As such, these are the ones chosen after careful consideration:

1. Improve the interactivity between the tour guide application and the tourist application;

   The functionalities provided by the mobile applications fulfill the established requirements. That being said, they have an enormous potential to be further developed, such as peer-to-peer streaming, audio guide for tourists or itinerary tracker.

2. Integration with other related platforms;

   These applications, although focused with tour guides necessities in mind, are easily expandable to related sectors, such as museum expositions. As such, and considering the work developed by Marta Lima [66], an integration of both platforms would achieve a quality proposition for both platforms.

3. Designing a business plan;

   This work was developed with few regards to monetization for academic purposes and as an open software mentality. It was only considered a tiered offer mentioned in chapter four (personal, community or enterprise). However, to be economically viable, it needs a well defined business plan that can make it a sustainable platform, a concern left for future consideration.

4. Adopt the media sharing technology to different sectors;

   This work was based upon tour guides necessities as the top priority, but the technology developed has a far reaching potential that is not limited to that sector. Therefore, the features provided by the mobile applications can be applied in different scenarios, such as presentations and lectures.

Bibliography

[1] M. Y. Çetinkaya and Z. Öter. Role of tour guides on tourist satisfaction level in guided tours and impact on re-visiting intention: a research in istanbul. *European Journal of Tourism, Hospitality and Recreation 7.1*, 7(1):40–54, 2016.

[2] W. T. Organization. Unwto anual report 2015. Technical report, UNWTO, Madrid, 2015.

[3] D. Bowie and J. C. Chang. Tourist satisfaction: A view from a mixed international guided package tour. *Journal of Vacation Marketing*, 11(4):303–322, 2005.

[4] Points of interest swg. `http://www.opengeospatial.org/projects/groups/poiswg`, 2016. [Online; accessed 26-November-2016].

[5] C. Grün, H. Werthner, B. Pröll, W. Retschitzegger, and W. Schwinger. Assisting tourists on the move-an evaluation of mobile tourist guides. In *Mobile Business, 2008. ICMB'08. 7th International Conference on*, pages 171–180. IEEE, 2008.

[6] B. Neuhofer, D. Buhalis, and A. Ladkin. A typology of technology-enhanced tourism experiences. *International Journal of Tourism Research*, 16(4):340–350, 2014.

[7] M. Teixeira. Point-of-interest based navigation system. Master's thesis, Instituto Superior Técnico, 2015.

[8] U. N. S. Commission et al. International recommendations for tourism statistics. *New York/Madrid: Author*, 2007.

[9] C. Cooper. *Tourism: Principles and practice*. Pearson education, 2008.

[10] Portugal2020. Programa operacional regional de lisboa 2014-2020. Technical report, Portugal2020, Lisbon, 2014.

[11] TravelBI. Turismo em números — janeiro 2017. Technical report, Turismo de Portugal, 2017.

[12] TravelBI. Análise regional 2016. Technical report, Turismo de Portugal, 2017.

[13] T. de Portugal. Animação turística em portugal 2014. Technical report, Turismo de Portugal, 2015.

[14] T. de Portugal. Inquérito a turistas. Technical report, Turismo de Portugal, 2015.

[15] A. M. Nunes. Turismo 2020, 5 principios para uma ambição. Technical report, Portugal2020, 2014.

[16] B. J. Pine, J. H. Gilmore, et al. Welcome to the experience economy. *Harvard business review*, 76: 97–105, 1998.

[17] T. de Portugal. Estratégia turismo 2027. Technical report, Turismo de Portugal, 2017.

[18] D. Wang, Z. Xiang, and D. R. Fesenmaier. Adapting to the mobile world: A model of smartphone use. *Annals of Tourism Research*, 48:11–26, 2014.

[19] D. Wang and Z. Xiang. The new landscape of travel: A comprehensive analysis of smartphone apps. In *Information and communication technologies in tourism 2012*, pages 308–319. Springer, 2012.

[20] J. E. Dickinson, K. Ghali, T. Cherrett, C. Speed, N. Davies, and S. Norgate. Tourism and the smartphone app: Capabilities, emerging practice and scope in the travel domain. *Current Issues in Tourism*, 17(1):84–101, 2014.

[21] B. Brown and M. Chalmers. Tourism and mobile technology. In *ECSCW 2003*, pages 335–354. Springer, 2003.

[22] I. International. Itb world travel trends report 2016/17. Technical report, ITB Berlin, 2017.

[23] Vayable. `http://www.vayable.com`, 2016. [Online; accessed 26-November-2016].

[24] Meetrip. `http://www.meetrip.com`, 2016. [Online; accessed 26-November-2016].

[25] Anyroad. `http://www.anyroad.com`, 2016. [Online; accessed 26-November-2016].

[26] Eurotrip planner. `http://www.eurotripplanner.eu`, 2016. [Online; accessed 26-November-2016].

[27] G. Inc. Google tour builder. `https://tourbuilder.withgoogle.com`, . [Online, accessed 26-November-2016].

[28] mtrip. `http://www.mtrip.com`, 2017. [Online; accessed 5-April-2017].

[29] G. Inc. Google maps. `http://developers.google.com/maps`, 2016. [Online; accessed 26-November-2016].

[30] S. Coast. Openstreetmap. `http://www.openstreetmap.org`, 2016. [Online; accessed 26-November-2016].

[31] Microsoft. Bing maps. `http://www.bingmapsportal.com`, 2016. [Online; accessed 26-November-2016].

[32] A. DuVander. Map apis report. Technical report, EverydayDeveloper, Portland, Oregon, 2017.

[33] S. Coast. Nominatim openstreetmap wiki. `http://wiki.openstreetmap.org/wiki/Nominatim`, 2017. [Online; accessed 26-March-2017].

[34] Google. Google places. `http://developers.google.com/places`, 2016. [Online; accessed 26-November-2016].

[35] D. C. Naveen Selvadurai. Foursquare. http://developer.foursquare.com, 2016. [Online; accessed 26-November-2016].

[36] Facebook. Facebook. https://www.facebook.com, 2017. [Online; accessed 13-April-2016].

[37] D. G. do Património Cultural. Informação geográfica. http://www.igeo.pt/DadosAbertos/Listagem.aspx. [Online, accessed 26-November-2016].

[38] C. M. de Lisboa. Lisboa aberta. http://dadosabertos.cm-lisboa.pt. [Online, accessed 26-November-2016].

[39] J. Wilbanks. Creative commons. https://creativecommons.org. [Online, accessed 12-April-2017].

[40] M. Janssen, Y. Charalabidis, and A. Zuiderwijk. Benefits, adoption barriers and myths of open data and open government. *Information systems management*, 29(4):258–268, 2012.

[41] G. Alonso, F. Casati, H. Kuno, and V. Machiraju. Web services. In *Web Services*, pages 123–149. Springer, 2004.

[42] L. Richardson and S. Ruby. *RESTful web services*. " O'Reilly Media, Inc.", 2008.

[43] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau. Extensible markup language (xml). *World Wide Web Journal*, 2(4):27–66, 1997.

[44] N. Nurseitov, M. Paulson, R. Reynolds, and C. Izurieta. Comparison of json and xml data interchange formats: A case study. *Caine*, 2009:157–162, 2009.

[45] CUBICRACE. How to convert a given xml snippet to an equivalent json format and vice-versa. https://http://www.cubicrace.com/2015/06/How-to-convert-XML-to-JSON-format.html. [Online, accessed 6-April-2017].

[46] Z.-R. Peng and C. Zhang. The roles of geography markup language (gml), scalable vector graphics (svg), and web feature service (wfs) specifications in the development of internet geographic information systems (gis). *Journal of Geographical Systems*, 6(2):95–116, 2004.

[47] Open gis consortium. http://opengeospacial.org, 2016. [Online; accessed 26-November-2016].

[48] O. Smartphone. Market share, q3 2016. *IDC.[Online] IDC Corporate USA*, 2016.

[49] M. D. Network. Web apis. https://developer.mozilla.org/en/docs/Web/API. [Online, accessed 13-April-2017].

[50] Google. Progressive web app. https://developers.google.com/web/progressive-web-apps/. [Online, accessed 13-April-2017].

[51] M. Palmieri, I. Singh, and A. Cicchetti. Comparison of cross-platform mobile development tools. In *Intelligence in Next Generation Networks (ICIN), 2012 16th International Conference on*, pages 179–186. IEEE, 2012.

[52] J. M. Wargo. *PhoneGap essentials: Building cross-platform mobile apps*. Addison-Wesley, 2012.

[53] B. Curtis. Ibm, phonegap and the enterprise. *PhoneGap Day 2011*, 1960.

[54] A. Sánchez Blanco et al. Development of hybrid mobile apps: Using ionic framework. 2016.

[55] X. Xamarin. Xamarin. 2014.

[56] S. Xanthopoulos and S. Xinogalos. A comparative analysis of cross-platform development approaches for mobile applications. In *Proceedings of the 6th Balkan Conference in Informatics*, pages 213–220. ACM, 2013.

[57] K. R. Lakhani and E. Von Hippel. How open source software works:"free" user-to-user assistance. *Research policy*, 32(6):923–943, 2003.

[58] I. K. Chaniotis, K.-I. D. Kyriakou, and N. D. Tselikas. Is node. js a viable option for building modern web applications? a performance evaluation study. *Computing*, 97(10):1023–1044, 2015.

[59] K. Lei, Y. Ma, and Z. Tan. Performance comparison and evaluation of web development technologies in php, python, and node. js. In *Computational Science and Engineering (CSE), 2014 IEEE 17th International Conference on*, pages 661–668. IEEE, 2014.

[60] N. Foundation. Nodejs 2016 user survey report. `https://nodejs.org/static/documents/2016-survey-report.pdf`. [Online, accessed 26-November-2016].

[61] C. Győrödi, R. Győrödi, G. Pecherle, and A. Olah. A comparative study: Mongodb vs. mysql. In *Engineering of Modern Electric Systems (EMES), 2015 13th International Conference on*, pages 1–6. IEEE, 2015.

[62] M. Inc. Introduction to mongodb. `https://docs.mongodb.com/manual/introduction/`, . [Online, accessed 26-November-2016].

[63] Automattic. Mongoosejs. `https://mongoosejs.com`. [Online, accessed 26-November-2016].

[64] T. Profiler. Qualcomm.

[65] S. Pandikumar and M. Sumathi. Analysis of energy profilers in smartphone environment.

[66] M. Lima. Mlime. Master's thesis, Instituto Superior Técnico, 2017.