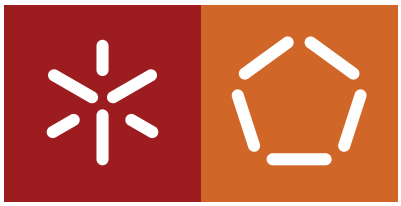
Universidade do Minho
Escola de Engenharia
Departamento de Informática

Miguel Ángel López Mamani

Web Portal for Touristic Social Communities

December 2015



Universidade do Minho
Escola de Engenharia
Departamento de Informática

Miguel Ángel López Mamani

Web Portal for Touristic Social Communities

Master dissertation
Master's Degree in Informatics

Dissertation supervised by
António Joaquim André Esteves

December 2015

DECLARAÇÃO

Nome: Miguel Ángel López Mamani

Endereço eletrónico: mlopez383@gmail.com

Telefone: +591 73609120

Número do Bilhete de Identidade: 5635032

Título da Tese:

Web Portal for Touristic Social Communities

Orientador:

Professor Doutor António Joaquim André Esteves

Departamento de Informática

Escola de Engenharia

Universidade do Minho

Ano de conclusão: 2015

Designação do mestrado: Mestrado em Informática

Universidade do Minho, 30/10/2015

ACKNOWLEDGEMENTS

First of all, I want to thank Professor António Joaquim André Esteves for the overall support, the research guidance and also for all the suggestions that improved this dissertation.

I am also grateful to Ricardo Ferrolho and Rui A. Costa from UBIWHERE Company, for giving me an insight about touristic social communities.

I also thank my parents, brother and sisters, for their love and support in every moment of my life.

Finally, I thank my friends.

ABSTRACT

The proposed dissertation uses techniques of Web Engineering to develop a web portal, where the users can plan and organize their trips, to define what touristic resources they would like to visit, and to keep track of the performed activities. The touristic resources are grouped in services and points of interest, where the services can be restaurants, hotels, hospitals, shops, ATMs, etc., and the points of interest can be monuments, parks, squares, or any famous free place.

The development of the application will take into consideration different types of potential tourists with different needs in different situations. The key to achieve this goal will be to carefully analyze the trip context. The web portal must be sufficiently flexible to be fed with different input types, such as texts, pictures, videos, and maps of geographic location.

The web portal also integrates Web 2.0 elements, such as communication and social networking, allowing the sharing of touristic resources, trips, and comments. On the other side, with the aim of building a large virtual touristic community, it was analyzed the current status of web portals for touristic social communities, the current status of web portals for a specific touristic place, and the interactions in a virtual community.

Additionally, a mobile native application was developed for the Android operating system using techniques of Mobile Native Application Engineering with emphasis in the trip and its landmarks, such as hotels, restaurants, health, attractions, and purchases. The mobile application contains GPS features that allow the user to navigate on the map of the trip and to see the touristic resources that are closer to him/her. The synchronization and coordination between the web portal and the mobile application is made by a secure REST API, which presents features of scalability and reliability in order to restrict accessing the trips to authorized users only.

Key Words: Web Engineering, Mobile Native Application Engineering, Trip planner, Travel organizer

RESUMO

A tese de dissertação aqui proposta utiliza técnicas de Engenharia Web para desenvolver um portal Web, onde o utilizador pode planear e organizar as suas viagens, definir quais os recursos turísticos que pretende visitar e manter um histórico dos recursos visitados. Os recursos turísticos estão agrupados em serviços e pontos de interesse, onde os serviços podem ser restaurantes, hotéis, lojas, hospitais, ATMs, etc., e os pontos de interesse podem ser monumentos, parques, praças ou um local famoso gratuito.

O desenvolvimento da aplicação terá em consideração tipos de turistas diferentes, com necessidades diferentes, em situações diferentes. Para se atingir este objetivo teremos que analisar o contexto da viagem de uma forma cuidada. O portal web deve ser suficientemente flexível para que possa ser alimentado com diferentes tipos de entrada, tais como fotografias, vídeos, e mapas de localização geográfica.

O portal Web deverá ainda integrar elementos Web 2.0, tais como comunicação, comunidades *online*, redes sociais, partilha de conteúdos e de experiências. Por outro lado, com o objetivo de construir uma grande comunidade turística virtual, foi analisado o estado atual dos portais web para comunidades turísticas sociais, o estado atual dos portais web para um local turístico específico, bem como as interações numa comunidade virtual.

Para complementar o portal Web, foi desenvolvida uma aplicação móvel para o sistema operativo Android, utilizando técnicas de Engenharia de Aplicações Móveis Nativas com ênfase na viagem e nos seus pontos de referência, tais como hotéis, restaurantes, unidades de saúde, atrações e locais para compras. A aplicação móvel contém funcionalidades GPS que permitem ao utilizador navegar no mapa da viagem e ver os recursos turísticos que estão mais próximos dele/dela. A sincronização e coordenação entre o portal Web e a aplicação móvel é feita por uma API REST desenvolvida, a qual apresenta características de escalabilidade e confiabilidade, a fim de restringir o acesso às viagens apenas aos utilizadores autorizados.

Palavras-chave: Engenharia Web, Engenharia de Aplicações Móveis Nativas, planificador de viagens, organizador da viagens

CONTENTS

LIST OF FIGURES

List of Figures

LIST OF TABLES

List of Tables

LIST OF SOURCE CODES

LIST OF ABBREVIATIONS

AJAX Asynchronous Javascript And XML

API Application Programming Interface

CMS Content Management System

CSS Cascading Style Sheets

DOM Document Object Model

GPS Global Positioning System

HTML HyperText Markup Language

HTTP HyperText Transfer Protocol

MVC Model-View-Controller

PAC Presentation-Abstraction-Control

REST Representational State Transfer

UML Unified Modeling Language

UC Use Case

XML Extensible Markup Language

1

INTRODUCTION

Summary

This chapter presents the theme of this dissertation. The issues related to the planning of the whole trip, and the time phases of a trip are explained. Afterwards, the motivation and main goals are detailed.

At the beginning of any touristic trip, the potential tourist thinks in a place of interest to visit or in an activity to perform. If the tourist does not know the location where the activity can be done, then the tourist searches for locals where the activity can be performed. The users increasingly use the Internet to prepare their trips, not only to locate the information in search engines and portals touristic, but also they go to communities of social networks consulting blogs, photos, videos, comments, geo-location systems, etc. (Plaza, 2008). Social networks are relevant because it is possible to see touristic experiences of people close to us.

Nowadays, many companies are using the Web to exhibit and sell products. In the tourism industry, it is possible to find travel agencies, hotels, transportation, open tours, and other services. In fact, it is possible to find very low prices if the user buys through the Web. It is possible to consult prices, timetables, and ticket sales (air tickets, train tickets, car rental, hotel reserves, etc.).

Evolution of the tourism industry has revolutionized the distribution and promotion of touristic resources, and destinations, forcing the focus of the strategy to be directed into individual tourists and niche markets.

Tourism Websites give support to 3 different phases of a trip: before, during and after the trip. Table 1 shows what a tourist does at each of these three phases.

Before the trip	The tourist plans the trip and can choose: • Places. • Time interval (hours, days or months). • Events (concerts, tours, etc.). • Accommodation (hostel, hotel, etc.). • Food (restaurants, pizzerias, etc.). • Others. Usually, the tourist consults tourism web sites, maps, photos, videos, textual description or any information in social networks, where he can see the experiences of other tourists.
During the trip	At this point the tourists are on a trip and can publish in social networks, where he is, who is he with, and also if he is participating in an event. It can be observed that during the trips the tourists usually carry mobile devices with them, such as smartphones or small laptops.
After the trip	At this point the tourist shares experiences, suggestions of places, hotels, restaurants, and other touristic resources. A report of the trip is created with his experiences.

Table 1.: Time phases of tourism in the Web.

1.1 IDENTIFICATION OF THE THESIS CENTRAL PROBLEM

Organizing a touristic trip can be a complicated task if the tourist travels to various places for a long time with other people and has a very restrictive budget. Therefore, the central problem is to plan and organize the whole trip on the web portal as best as possible, based on the tourist profile and with the goal of reducing costs and avoiding any trouble during the trip.

1.2 MOTIVATION

The interest in the area of web engineering results from the fact that a web application can reach millions of people in the world through the Internet. A web application can be developed in many ways. It is possible to choose different technologies, such as programming languages, frameworks, content management system, and multimedia.

This dissertation will be developed partially at the Ubiwhere company, with the objective of integrating the Web portal into an existent platform dedicated to tourism. Ubiwhere works in the field of software for Innovative Pervasive Services (IPS).

1.3 BACKGROUND

This section explains the importance of tourism in Portugal and the importance of the centralization, and availability of information online.

1.3.1 *Tourism in Portugal*

According to the World Tourism Organization, a specialized agency of the United Nations and the main international organization in the field of the tourism, Portugal is one of the top 20 destinations (Wikipedia, 2014b) of the world, Portugal is widely recognized in Europe by the sun, beaches, gastronomy and cultural heritage and property.

Despite the fact that some hotels offered discounts due to the recession in order to attract tourists, the prices of the hotels have risen on average about 4% compared to last year and reached a new record in tourism in Portugal. The number of foreign visitors in Portuguese hotels increased to 7,44 million, overcoming the record of 2008, which was 7,12 millions, and 9 % more that in 2010 according to the numbers provided by the National Institute of Statistics.

It is also important to refer that 25% of tourists that travel through Portugal, using the itinerary format for their trips and that the total income of the tourism is responsible for about 10% of the gross domestic product of Portugal.

Having regard to the numbers and the facts presented above, the existence of a platform where there is updated information about itineraries and that is available to everyone, presents great advantages for the Portuguese tourism.

1.3.2 *Use of the Internet for Tourism Information*

The use of the internet (Stats, 2014) has had a growth quite accentuated in all the world. Only in the last 10 years, it increased 444% reaching approximately 2 billion users.

According to the Website World Internet Usage Statistics, the percentage of users by continents is presented in the table below and as you can see Europe represents one of the highest percentages in the world.

With the growth of Internet use, also increased its popularity and its range of features. Nowadays, the internet is by far the most popular source of information and the preferred choice for news and information relevant, being considered the most reliable source of information.
The Internet is increasingly present in our day and is widely used for the search of all kinds of infor-

Continent	Percentage of users who use the Internet
Africa	8.6% (240,146,482 people)
North America	10.7% (300,287,577 people)
Latin America	10.8% (302,006,016 people)
Asia	45.1% (1,265,143,702 people)
Middle East	3.7% (103,829,614 people)
Europe	20.2% (566,261,317 people)
Oceania	0.9% (24,804,226 people)
Total	100.0% (2,802,478,934 people)

Table 2.: Percentage of Internet users by continent.

mation, leaving behind the traditional systems of information dissemination (radio, television, newspapers, etc.).

1.4 OBJECTIVES

The main goals of this thesis are:

- To structure and organize the sources of tourist information, especially those that are physically static as museums or churches. The information can include images, videos, descriptions, weather, events (tours, concerts, etc.), maps, resources (restaurants, hotels, museums, churches, car rental, etc.) of a particular place.
- To develop a module where the user can plan and organize his trips in a daily point of view, with suggestions based on his profile (interests, age, nationality, etc.) and on the time considered in the trip planning.
- To develop a back-office that manages: touristic resources, trips, and users.
- To allow the tourists to create their picture albums.
- To allow registration and login in the web portal through social networks like Facebook or Twitter.
- To share tourist information (pictures, messages, links) in social networks (Facebook, Twitter, Google+, etc.).
- To develop a module that allows a tourist to print his trip plans. This way he can access the information in places where he does not have access to the Internet.
- To develop a web template with an intuitive aspect, ensuring a usability and navigability that motivate visiting and navigating in the web portal.

1.5 OVERVIEW

This dissertation is divided into six chapters. The first chapter presents the theme of this dissertation, the issues related to the planning of the whole trip, the time phases of a trip, the motivation, and the main goals.

The second chapter compares the main web portals for touristic social communities and analyses the current status of touristic web portals for a specific touristic place. Afterwards, it is explained the foundation of building a virtual tourist community.

In the third chapter, we present and compare the technologies necessary to develop web portals and mobile applications. Agile and traditional methodologies are analyzed in order to be used in the software development process. Afterwards, it is explained the choice of a mobile operating system and the importance of social networks for the present thesis.

The fourth chapter presents the web portal design using the Scrum agile methodology. Afterwards, it is described the requirements gathering process, with emphasis on two crucial phases: elicitation and documentation. The elicitation phase relies on techniques such as personas, interviews and conversations with the Ubiwhere company staff, documentation reading, and analysis of similar web portals. The documentation of requirements was done with the Volere template. The stakeholders, the use cases, the domain model, the class diagram and the data model are also presented. Afterwards, the MVC and PAC design patterns are compared.

The fifth chapter presents the implementation of the web portal and mobile native application, and the achieved results. It will be described the main features and how they were implemented. Finally, it is presented a REST API that is responsible for synchronizing the web portal with the mobile application.

The last chapter we present the main conclusions and contributions of the dissertation. To conclude, some notes are given on how this work can be further developed.

2

RELATED WORK

Summary

This chapter compares the main web portals for touristic social communities and analyses the current status of touristic web portals for a specific touristic place. Afterwards, it is explained the foundation of building a virtual tourist community.

Tourism is the way in which people search for a psychological reward by a temporary experience in new places, and situations, where they could be free from work and normal daily life style at home (Ryan, 1991). In this section, the leading touristic web portals are analyzed. Also, the touristic web portals for a specific place are analyzed.

2.1 CURRENT STATUS OF THE WEB PORTALS FOR TOURISTIC SOCIAL COMMUNITIES

In this section are analyzed the web portals for touristic social communities, in order to compare similar solutions. The evaluated web portals, whose logo is shown in Figure 1, are TripAdvisor[1], Yahoo Trip Planner[2], TouristEye[3], and TripIt [4].

2.1.1 *TripAdvisor*

TripAdvisor is the most popular touristic web portal that provides reviews of travel-related content. It also includes interactive forums. The users of the web portal provide most of the content. The web portal is supported by an advertising business model.

1 http://www.tripadvisor.com/
2 http://travel.yahoo.com/trip
3 http://www.touristeye.com/
4 http://www.tripit.com/

Figure 1.: Main web portals for Tourism.

TripAdvisor is focused mainly in six parts:

- **Cities**
- **Hotels** grouped into:
  - **Price Range**.
  - **Hotel Class**, categorized from one to six stars.
  - **Style**, categorized into economic, luxury, quiet, charming, mid-range, romantic, classic, boutique, family-friendly, green, trendy, quaint, and resort hotel.
  - **Location**, categorized into popular, neighborhoods, and near to the attractions.
  - **Amenities**, categorized into room service, free breakfast, restaurant, wheelchair access, bar, lounge, free Internet, air conditioning, parking, suites, fitness center, pets allowed, airport transportation, pool, casino, shuttle service, and spa.
  - **Hotel brand**

  At the same time, the hotels are subdivided into bed and breakfast (B&B), Inns, specialty lodgings, vacation rentals, and special offers. It is possible to make reservations via external booking web sites, such as the official website hotel, Expedia[5], Booking[6], Despegar[7], and so on.
- **Restaurants** grouped into:
  - **Establishment Type**, categorized into restaurants, dessert, coffee & tea, bakeries, ice cream, waffles & crepes, and juice & smoothies.
  - **Reservations**, categorized into restaurant deals and available tonight.

5 http://www.expedia.com/
6 http://www.booking.com/
7 http://www.despegar.com/

  - **Cuisine**, categorized into typical national food, pizza, dessert, barbecue, steakhouse, delicatessen, café, vegetarian, seafood, bakery, fusion, eclectic, pub, and soups.
  - **Meals**, categorized into breakfast, brunch, lunch, and dinner.
  - **Price**, categorized into cheap, mid-range, and fine dining.
  - **Location**, categorized into neighbourhoods and airports.
  - **Restaurant Features**, categorized into buffet, delivery, free Internet, gluten free, outdoor seating, parking available, private dining, reservations, serves alcohol, takeout, television, and wheelchair accessible.
  - **Type of Meeting**, categorized into bar scene, business meetings, child-friendly, groups, local cuisine, romantic, special occasion dining, and view.

- **Things to do** grouped into:
  - **Attractions**, categorized into cultural, landmarks, museums, performances, amusement, outdoors, sports, zoos, and aquariums.
  - **Activities**, categorized into sightseeing tours, food/drink, wellness/spas, adventure, classes, gear rentals, ranch, and farms.
  - **Night-life**, categorized into bars and clubs
  - **Shopping**.

- **Flights**, categorized into round trip, one-way, and multi-city.
- **Vacation Rentals**, filtered by price, quantity of rooms, and amenities.

TripAdvisor also has a mobile application that works on iPhone, iPad, telephones with Android, Nokia, and Windows Phone with similar functionalities to the web portal such as:

- Browse reviews, opinions, videos, and photos made by travelers related to a touristic resource.
- Find hotels.
- Explore restaurants by food type, price range, and rating.
- Discover things to do in any destination.
- Compare airfares and find great deals.
- Discover places near to the user or any address.
- Get answers to your specific travel questions in the forums.
- Add your own reviews and photos.

- Download maps, reviews, cities into the phone for free.

TripAdvisor uses several external services such as The Weather Underground for weather information, Google Maps for geo-location maps. It also has accounts in the social networks Facebook and Twitter.

2.1.2 *Yahoo Trip Planner*

Yahoo Trip Planner is a web portal providing reviews of travel-related content. The web portal is grouped into travel guides, hotel bookings, and flights bookings. The web portal is supported by an advertising business model.

The web portal is focused mainly in four parts:

- **Cities**
- **Hotels** categorized by:
  - **Popularity**.
  - **Amenities**, which shows information about of free breakfast, room service, free Internet, air conditioning, parking, refrigerator, telephone, television, safety, kitchen, balcony/terrace, mini-bar, and so on.
  - **Hotel details**, which show information about quantity of rooms, quantity of floors, electronic key, laundry, multi-lingual staff, year of construction, car rental, photocopy service, administrative service, fax, ATM/Bank, room service, restaurant, casino, health club, game room, shopping mall, and so on.
  - **User reviews**, rated by cleanliness, location, value, and service.

  Is possible make reservations via external booking website such as Expedia, Booking, or Despegar using TripAdvisor.
- **Things to Do** categorized by:
  - **Popularity**.
  - **User reviews**, rated by atmosphere, and value.
- **Restaurants** categorized by:
  - **Popularity**.
  - **User reviews**, rated by service, food, ambiance, and value.

Yahoo Trip Planner uses several external services such as Flickr, and Instagram for Photo/Video hosting, The Weather Channel for weather information, Yahoo Maps for geo-location maps. It also has accounts in the three leading social networks: Facebook, Google plus, and Twitter. Additionally, it uses Tumblr, the microblogging platform.

2.1.3 *TouristEye*

TouristEye is a web portal providing tips about travel-related content. The web portal supports only two languages English and Spanish. TouristEye claims that 800,000 people have traveled with the platform around the world.

TouristEye is focused mainly in seven parts:

- **Cities**
- **Accommodation**, categorized into hotel, hostel, B&B, and camping.
- **Things to Do**, categorized into sights, museums, nature, sports, shopping, night-life, entertainments, activities, and events.
- **Experiences**, described by characteristics like sunsets, local dishes, secret gardens, etc.
- **Transports**, categorized into airport, train station, bus station, port, and car rental.
- **Rentals**
- **Restaurants**, categorized into food by country, coffee, and pizzas.

TouristEye uses several external services such as Pinterest for photo hosting, Google Maps for geolocation maps, MixPanel for web/mobile analytics. It also has accounts in the three leading social networks: Facebook, Google plus, and Twitter. The mobile application works on iPhone and smartphones with Android.

The TouristEye mobile application has the following features:

- To plan the trip
- To collect ideas for the trip and getaways in the format of a wish list.
- To show recommendations for nearby places while you are on your trip. This feature works offline.
- To create community and collaborative wish lists. For example, we can create a wish list about beaches for surfing or visiting a city with children.
- To show texts, photos and maps. This feature works offline.

2.1.4 *TripIt*

TripIt is a web portal that organizes a trip in one place. We can simply forward our travel confirmation emails to plans@tripit.com and TripIt automatically creates a detailed daily itinerary for every trip. TripIt belongs to Concur company, that has over 15 million people in the tourism industry.

The mobile application works on iPhone, iPad, smartphones with Android, Blackberry, and Windows Phone 7.

The TripIt web portal and the mobile application have the following features:

- To forward hotel, airline, car rental, and restaurant confirmation emails to plans@tripit.com to create an instant itinerary.
- To access itineraries any time, on any device, even offline.
- To get directions, maps and weather for each destination.
- To synchronize trip plans with Google Calendar or Outlook.
- To add or edit plans manually.
- To share trip plans via email or social networks.

TripIt is supported by an advertising business model. Also, has a payment version called TripIt Pro without advertising, for frequent travelers, with a price of $49 per year that provides more features, such as more real-time flight alerts, seat tracker, flight finder, refund notifications, point tracker, etc.

TripIt uses several external services such as Google Maps for geo-location maps. It also has accounts in the social networks Facebook, Twitter, and LinkedIn.

2.2 COMPARISON OF WEB PORTALS

It is possible to observe that Yahoo Trip Planner and TripAdvisor have many strong points such as a great quantity of user feedback and many details in the qualification and valuation of the touristic resources. On the other hand, TouristEye is the new alternative of trip planners. TouristEye's strong point is its simplicity and navigability, something that does not happen with TripIt, which allows us to enter more detailed information at the moment of creating a trip. Table 3 shows the differences among the evaluated web portals.

All the evaluated web portals for tourism have the following features in common, which can be considered as the main requirements of any trip planner:

- Register or login with Facebook account.
- Add touristic resources (Points of Interest or Services) to the trip with Maps.
- Add comments to the trip.
- Add comments to the touristic resources.
- Visualize touristic resources by popularity.
- Share the trip through Facebook and Twitter.
- Share touristic resources through Facebook.
- Add touristic resources to the trip using text.

	Yahoo T.	TouristEye	TripAdvisor	TripIt
Register/login with Google Account	✓		✓	✓
Register/login with Twitter Account		✓		
Web mapping service	Yahoo	Google	Google	Google
Categorization of touristic resources	✓		✓	✓
Add travelers to the trip		✓	✓	✓
Export trip to the calendar				✓
Add a photo album through Flickr	✓		✓	
Send SMS with the link of trip	✓	✓		
Print the trip	✓		✓	✓
Print details of touristic resource	✓		✓	
Add tags to the trip	✓			
Add tags to the touristic resources	✓		✓	
Applications for mobile phones		✓	✓	✓
Search the web	✓	✓	✓	

Table 3.: Comparison of trip planners.

2.3 CURRENT STATUS OF THE TOURISTIC WEB PORTALS FOR A SPECIFIC PLACE

In this section are analyzed touristic web portals for specific places, in order to get more details about a touristic resource because the accommodation, transport, and food are different from one place to another. The evaluated web portals for a specific place are ParisInfo[8], VisitPorto[9], and VisitDublin[10]. The logo of these portals are shown in Figure 2.

8 http://www.parisinfo.com/
9 http://www.visitporto.travel
10 http://www.visitdublin.com/

Figure 2.: Main touristic web portals for specific places.

2.3.1 *ParisInfo*

ParisInfo is a web portal that provides tourist information about the city of Paris, France. ParisInfo is supported by the Paris Convention and Visitors Bureau. ParisInfo has a high level of internationalization supporting up to eleven languages: French, English, Spanish, German, Italian, Portuguese, Dutch, Russian, Japanese, Chinese, and Korean.

ParisInfo is focused mainly in six parts:

- **Touristic Information** grouped into:
  - **Explore** major events, walks in Paris, the first time in Paris, travel agencies, images, sustainable tourism, themed guides, booking agencies.
  - **Where to Stay**, sub-categorized into hotels, aparthotels, bed and breakfasts, furnished rentals, camp sites, unusual hotels, accommodations for young people, and accommodations for family.
  - **Eating Out**, sub-categorized into gastronomic restaurants, French food, foreign specialties, on a budget, a meal in an unusual setting, cafés, bars, pubs, tea rooms, ice-cream sellers, fine food shops, caterers, and cooking schools.
  - **What to See**, sub-categorized into museums, monuments, guided tours, theme parks, parks, gardens, zoos, aquariums, and places of worship.
  - **Going Out**, sub-categorized into exhibitions, shows, concerts, celebrations, festival, sports, games, trips, outdoors, fairs, trade shows, and shopping.

  - **Transport**, sub-categorized into airport arrival, arrival by car, arrival by train, arrival by boat, arrival by bus, public transport, car rental, taxis, cycling tours, original transports, bus sightseeing, plans, and maps.
  - **Practical Paris**, sub-categorized into welcome centers, Paris city guides, practical facts sheets, on-line ticket sales, hotel booking, guides for people with some disabilities.

- **Booking**, sub-categorized into hotels, accommodations, museums, monuments, transports, seine cruises, cabarets, shows, open bus tours, trips, and tours.
- **Paris city passport**, which is used to visit places and use public transportation.
- **Tools**, sub-categorized into weather information, currency converter, surface area converter, and newsletter.
- **Event Organizers**.
- **Travel Trade**.

ParisInfo uses some external services such as Instagram for Photo/Video hosting, Google Maps for geo-location maps, MeteoVista for weather information. It also has accounts in the social networks Facebook and Twitter.

2.3.2 *VisitDublin*

VisitDublin is a web portal that provides tourist information about the city of Dublin, Ireland. VisitDublin supports only the English language. The web portal is supported by the National Tourism Development Authority of Ireland.

VisitDublin is focused mainly in nine parts:

- **Activities**, sub-categorized into nightclubs, shopping, sports, traditional Irish music and dance, cafés, pubs, and restaurants.
- **Upcoming events**, sub-categorized into festivals, free events, theater shows, music, dance, concerts, gigs, films, comedy events, and sport events.
- **Tours & attractions**, sub-categorized into Dublin's history, museums, galleries, visitor attractions, guide walking tours, city tours, literary, markets, parks, and gardens.
- **Accommodations**, sub-categorized into hotels, hostels, guest-houses, self-catering, country houses, farm houses, caravan, and camping.
- **Inside Dublin**, sub-categorized into top views, quirky things, events for saints, events for sinners, summer, nights, budget, fashion, and shopping.

- **Travel info**, sub-categorized into airlines, ferries, bus operators, car rental, motorbike rental, bicycle rental, campervan rental, light-rail transit service, and rail.
- **Special offers**, sub-categorized into food, drink, tours, and travel.
- **Competitions** where anyone can win a ticket for a tourist activity.
- **Download** Dublin maps, podcast audio guides, insider guides, rss feeds, mobile applications.

VisitDublin uses several external services as Flickr and Instagram for Photo/Video hosting, Pinterest for photo hosting, Youtube for video hosting, Google Maps for geo-location maps. It also has accounts in the three leading social networks: Facebook, Google+, and Twitter.

2.3.3 *VisitPorto*

VisitPorto is a web portal providing tourist information about the city of Porto, Portugal. VisitPorto supports ten languages: Portuguese, English, Spanish, German, Italian, French, Dutch, Russian, Japanese, and Chinese.

VisitPorto is focused mainly in five parts:

- **Visiting** grouped into:
  - **Discover**, sub-categorized into port wine, museums, heritage, historical center, sightseeing tours, architecture, culture, leisure, river & sea, outdoors, shopping, gastronomy, nightlife, and fashion.
  - **Your Trip**, sub-categorized into basic information on Porto, public transportation system, sightseeing tours, special offers, and flights.
  - **Community**, sub-categorized into experience, blogs, and stories.
  - **Porto Card**, which is used to visit places and use public transportation.
- **Living** grouped into:
  - **Why Porto**.
  - **Accommodation**.
  - **Education & research**.
  - **Working**.
  - **Useful Information**.
  - **Stories**.
- **Business** grouped into:

  - **Why Porto**.
  - **Event Planning**.
  - **Investing in Porto**.
  - **Research & Development in Porto**.
  - **Leisure in Business**.
  - **Trade**.

- **What's on** grouped into:
  - **Sports**.
  - **Special Events**.
  - **Exhibitions**.
  - **Music and Festivals**.
  - **Night-life**.
  - **Theater and Dance**.
  - **Workshops and Guided Tours**.
  - **Business**.

- **More Porto** grouped into:
  - **Porto's History**.
  - **Geography and Location**.
  - **Tradition**.
  - **Multimedia Gallery**.
  - **Events**.

VisitPorto uses several external services as Flickr for Photo/Video hosting, Pinterest for photo hosting, Youtube for videos, Google Maps for geo-location maps, Issuu for digital publishing. It also has accounts in the social networks Facebook and Twitter. The mobile application works on iPhone, iPod Touch, iPad, smartphones with Android, Blackberry, Windows Phone, and Nokia.

2.4 VIRTUAL COMMUNITY

A virtual community is a social network where enough people carry on contents long enough, with sufficient human feeling through specific content like images, texts, videos, maps, or any media information in order to form communities in the cyberspace. A virtual community can cross geographical and political boundaries where the people or places can be considered like tools (H., 1993).

2.4.1 *Touristic Virtual community*

There are several specialized information communities, that centralize a specific group of users who are interested in the same topic where the users feel that their contributions are important and that by contributing, they are forming meaningful connections with other users. Sharing and using the information require that users are able to add, modify, remove, upload, download, display, and share media content. A Touristic virtual community is a specialized information community where enough people, such as tourists or travelers, carry on contents long enough, with sufficient human feeling through photos, comments, videos, maps, or any information related to tourism.

A Touristic virtual community helps tourists to overcome risks regarding tourism destination choice, such as monetary risks (e.g. feeling of wasting money), functional risks (e.g. the standards of the place did not meet tourists goals), physical risks (e.g. getting injured or ill), social risks (e.g. visiting a fashionable place and getting a high status), and psychological risks (e.g. not damaging the self-esteem, by feeling guilty that there was not enough time spent looking for more information about the place) (L.V. et al., 2010).

2.4.2 *Building a virtual touristic community*

In order to build a large virtual touristic community, it is necessary for people to get information much more easily, maintain connections, and deepen relationships. It implies design a virtual community based on a comprehensive understanding of the consumers' functional, social, and psychological needs (Dimitris and Vasiliki, 2013). Building a virtual touristic community comprises two phases. The first phase consists in the adoption of affiliation strategies, which are based on attractive contents and positioning on a web search engine. The affiliation strategies have the following purposes: providing solving-problems services, yielding comprehensive information and adding decision-support software on content (Daniela Baglieri, 2008). The information gathered in the first phase, by tracking and generating a visitor profile, give us a deeper understanding about dominant online behavior models. The second phase consists in deploying a lock-in strategy by increasing customer stickiness and customer loyalty. This phase is based on three strategies: building an installed customers base, capturing customers through product design and promotion, leveraging customer commitment by selling complementary products (Daniela Baglieri, 2008). Therefore, the most important interactions in a virtual community are:

- **Identification:** This concept is divided in affective and cognitive components: the cognitive one is said to be the result of perception of similarities with other members and dissimilarities with people who are not in the group. The affective component is believed to be this type of identification experienced when a person is emotionally involved with a group (L.V. et al., 2010). Identification with a group impacts consumer motivation to communicate and interact

with the rest of group members. It leads to further participation in a group and recommending community.

- **Satisfaction:** We may define satisfaction as an affective condition that results from a global evaluation of all the aspects that make up the relationship. We can say that if a product is better than a customer assumed, he or she is satisfied. If it is not, the customer is dissatisfied (L.V. et al., 2010).

- **Continuance Participation:** It is very important as the future and success of the community rely heavily on it. When we participate in the life of the virtual community, we can guarantee its endurance and help achieve groups' goals (L.V. et al., 2010).

- **Community Promotion:** When a consumer wants that a certain virtual community to be known or recommended to non-members, he or she uses promotion. Community Promotion is also very important in a life of virtual community. It makes it more attractive to the potential future members (L.V. et al., 2010).

- **Loyalty:** The longer the tourists are in a virtual community, more loyal and committed remain with this organization because they feel that the amount of value received for participating in the community is greater than the value for not participating (L.V. et al., 2010).

3

TECHNOLOGIES

Summary

In this chapter we present and compare the technologies necessary to develop web portals and mobile applications. Agile and traditional methodologies are analyzed in order to be used in the software development process. Afterwards, the fundamental and complementary technologies are detailed. Finally, the choice of mobile operating system is explained and the importance of social networks for our applications.

3.1 THE SOFTWARE DEVELOPMENT PROCESS

Software development is a creative process leading to a software product. Traditionally, the process can be decomposed into the following basic activities (Casteleyn et al., 2009):

- *Requirements engineering*: aims at understanding the problem.
- *Design*: aims at planning a solution to the problem.
- *Implementation*: translates the plan into running application code.
- *Testing and evaluation*: aims at identifying coding errors or inconsistencies between the collected requirements and their implementation.
- *Deployment*: brings the solution to the customers.
- *Maintenance*: aims at monitoring a running system and keeping it healthy and running.
- *Evolution*: aims at improving the developed solution over time, providing new input to the development process in the form of new requirements.

As a result of studies in this area, it was defined a set of methods, models and techniques, which can be divided into 2 groups: agile methodologies, and web engineering methodologies. The difference

between them are explained in the next 2 sections.

3.1.1 *Agile Methodologies*

Agile methodologies can be characterized as it follows (Fowler, 2001):

- *Agile methods are adaptive rather than predictive*. Engineering methods tend to plan out a large part of the software process in great detail for a long span of time. This strategy works well until things change. So their nature is to resist change. The agile methods, however, welcome change. They try to be processes that adapt and thrive on change, even to the point of changing themselves.

- *Agile methods are people-oriented rather than process-oriented*. The goal of engineering methods is to define a process that will work well whoever happens to be using it. Agile methods assert that no process will ever make up the skill of the development team, so the role of a process is to support the development team in their work.

The most popular agile methodologies are:

- Extreme Programming (XP) has five values: communication, feedback, simplicity, respect, and courage. This minimal and consistent set of values allows to make life easier for the group, the management and the customers. Serves both for human and commercials ends. The practices of Extreme Programming are pair programming, an on-site customer, continuous integration, and test-first development (Holcombe, 2008).

- Scrum: It concentrates on the management aspects of software development, dividing development into short iterations called *sprints* (typically lasting for 3 weeks) and applying closer monitoring and control with daily scrum meetings. The main roles in Scrum are: (1) the Scrum Master, the person who maintains the processes, (2) the Product Owner, the person who represents the stakeholders and the business, and (3) the Team, a cross-functional group who does the actual analysis, design, implementation, testing, etc. (Rubin, 2013).

3.1.2 *Web Engineering Methodologies*

Web engineering methodologies can follow different paradigms: Data-oriented, Hypertext-oriented, Object-oriented, and Software-Oriented (Casteleyn et al., 2009). The most popular proposals in the area of Web Engineering are UML-based web engineering (UWE), Web Modelling Language (WebML), and Object-Oriented Hypermedia Design Method (OOHDM). Table 4 presents a comparison of these proposals. WebML is considered the best methodology for web applications development

among them (Wakil and Jawawi, 2014) (dos Santos Domingues et al., 2008) because it allows modelling with a visual language, and supports more design features than UWE (Wakil et al., 2015). On the other side, the hypertext design with WebML is richer than OOHDM (Casteleyn et al., 2009). This comparison helps designers and analysts in the selection of a web engineering methodology for developing web applications.

	UWE	WebML	OOHDM
Modeling paradigm	Object Oriented	Data Oriented	Object Oriented
Notation	UML	UML, Entity-Relationship (ER)	UML + own notation
Evolving	✓	✓	✓
Requirements Modeling	✓	✓	✓
Content Modeling	✓	✓	✓
Hypertext Modeling	✓	✓	✓
Presentation Modeling	✓	✓	✓
Customization Modeling	✓	✓	✓
Structure and Behavior	✓	✓	✓
User Modeling		✓	✓
Design reuse		✓	✓
Process/Approach	RUP	own	own
Strengths	UML-based method, model-driven development, aspect-oriented customization	well-elaborated notation, database integration, generation	powerful concepts for contextual navigation, personalization

Table 4.: Comparison of Web Engineering Methodologies.

3.2 WEB ENGINEERING

Web Engineering is defined as follows (Altmann and Arora, 2006):

1. Web Engineering is the application of systematic and quantifiable approaches (concepts, methods, techniques, tools) to cost-effective requirements analysis, design, implementation, testing, operation, and maintenance of high-quality Web applications.

2. Web Engineering is also the scientific discipline concerned with the study of these approaches.

3.2.1 *Fundamental technologies*

The fundamental technologies when developing a web portal are HTML, CSS and Javascript. Although currently jQuery is replacing Javascript like a mask, because jQuery is simpler to use than Javascript. On the other hand HTML5 is prevailing in current applications, but is not yet widely supported by all browsers, so we chose to follow the XHTML standard.

HyperText Markup Language

HyperText Markup Language (HTML) is the standard markup language used to create web pages. HTML allows images, videos, texts, and other objects to be embedded and can be used to create interactive forms. In turn, the eXtensible HyperText Markup Language (XHTML) is a family of XML markup languages that mirrors or extends HTML. XHTML adds extensibility and portability to the web pages.

Javascript

Javascript (JS) is a dynamic programming language used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the web page content that is displayed.

Cascading Style Sheets

Cascading Style Sheets(CSS) is a style sheet language used for describing the look and formatting of a document written in a mark-up language such as HTML, XHTML, or HTML5.

3.2.2 *Complementary Technologies*

Before start developing a web portal, several technologies were compared in order to select those that best fit our needs.

Programming Languages

Today, a wide variety of server-side scripting languages is available. The factors that influence the choice of the most appropriate language include the availability of the technology on the desired operating system, the support for the scripting language by the Web Server, the price, and the complexity of the required functionality. In the following, we summarize three popular scripting languages.

JAVA SERVER PAGES

Java Server Pages (JSP)[1] is a technology that extends Java servlets. In its basic form, a JSP page

1 http://java.sun.com/products/jsp

is simply an HTML web page that contains additional bits of code that execute application logic to generate dynamic content. This application logic may involve JavaBeans, JDBC objects, Enterprise Java Beans (EJB), and Remote Method Invocation (RMI) objects, all of which can be easily accessed from a JSP page.

HYPERTEXT PREPROCESSOR

Hypertext Preprocessor (PHP)[2] is one of the most popular server-side scripting language that is especially used for web development and can be embedded into HTML. This open source scripting language has a huge supporting community, available on all major platforms, and is supported by all popular Web servers. It is in essence an imperative dynamically typed language, enhanced with object-oriented programming constructs.

PYTHON

Python[3] is a general-purpose, high-level programming language whose design philosophy emphasizes code readability. Its syntax is said to be clear and expressive. Python supports multiple programming paradigms, primarily but not limited to object-oriented, imperative and, to a lesser extent, functional programming styles. Python is often used as a scripting language for web applications. Web application frameworks like Django, Pylons, or TurboGears support developers in the design and maintenance of complex applications. Actually, the most popular Content Management System (CMS) for Python is Plone.

JavaScript libraries

A JavaScript library is a library of pre-written JavaScript which allows for easier development of JavaScript-based applications, especially for AJAX and other web technologies.

JQUERY

jQuery[4] is the most popular cross-platform JavaScript library designed to simplify the client-side scripting of HTML. The syntax of jQuery is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. Also, jQuery is free, and open source software.

PROTOTYPE

Prototype[5] is a cross-platform JavaScript library that provides a features range from programming shortcuts to major functions for dealing with XMLHttpRequest. Also, Prototype provides library functions to support classes and class-based objects.

2 http://www.php.net
3 http://www.python.org
4 http://jquery.com
5 http://prototypejs.org

Web Content Management System

A Web Content Management System (WCMS) is a computer program that provides tools designed to allow users with little knowledge of web programming languages, or markup languages, to create and manage websites content with relative facility. In the following, we summarize three popular web content management systems.

DRUPAL

Drupal[6] is an open source web content management framework with a hook-based architecture. It is highly modular and extensible. In Drupal, the modules are plugins that extend, build or enhance Drupal core functionality. Drupal requires some expertise and experience to operate. Drupal can be used to build an Internet portal, an e-commerce site, an online newspaper, a social networking site, and virtually any other type of website that one can imagine. Drupal is known for its powerful taxonomy and ability to tag, categorize, and organize complex content (Tomlinson, 2010).

JOOMLA

Joomla[7] offers a middle ground between the developer-oriented extensive capabilities of Drupal and the user-friendly, but more complex site development options, of Wordpress.

WORDPRESS

WordPress[8] is the most popular blogging platform. This WCMS is very user-friendly with an ever-increasing repertoire of themes, plugins and widgets. As a WCMS used in other ways than blogging, it can be a viable solution, but is not nearly as potent as Joomla or Drupal.

Database Management System

A database management system (DBMS) consists of collection of interrelated data and a set of programs to access that data. It is a software that is helpful in maintaining and utilizing a database (Sumathi and Esakkirajan, 2007).

MYSQL

MySQL[9], the world's most used open source relational database management system has emerged as an attempt of connection between the manager mSQL and the own tables of MySQL AB, using low-level routines. After initial tests, they found that mSQL was not flexible enough, so they had to develop new functions. This resulted in an SQL interface to the database with a fully compatible interface to mSQL. The main features are:

6 http://drupal.org
7 http://www.joomla.org
8 http://www.wordpress.org
9 http://www.mysql.com

- It takes advantage of multiprocessor, thanks to its multi-thread implementation.
- It supports many data types in columns.
- It is supported in many programming languages, such as C, C++, Java, or PHP.
- It presents high portability between systems.
- It supports ODBC and JDBC connections.
- The tables in MySQL are composed by three files with extensions ISD, ISM, and FRM.

POSTGRESQL

PostgreSQL[10], is an object-relational database management system (ORDBMS) available for many platforms. PostgreSQL is considered the open source database most advanced of the world. The main features are:

- It implements the SQL92/SQL99 standard.
- It allows the creation of own data types.
- It incorporates a data structure array.
- It incorporates functions of diverse nature, i.e. orientated to operations with networks.
- It allows the declaration of own functions and the definition of triggers.
- It includes inheritance between tables.
- It supports ODBC and JDBC connections.
- The architecture follows the client/server model.

PostGIS is an extension to PostgreSQL that allows using Geographic information systems (GIS) objects (Ramsey, 2005). PostGIS was created by Refractions Research Inc. as a research project of technologies for spatial databases. It is released under GNU license. With PostGIS it can be used all the objects in the OpenGIS specification, such as points, lines, polygons, multi-line, multi-point, and geometric collections.

10 http://www.postgresql.org

Web Mapping Service

A web mapping service provides at least the next two mandatory functions (Peterson, 2014):

1. *GetCapabilities* that defines the capabilities of the server, such as the supported operations, the supported file formats, the available map layers, and the spatial reference systems (SRS) available to deliver map data.

2. *GetMap* requests a map from the server. The parameters specify the layers and styles to appear on the map, a bounding box for the map extent, an SRS, and a width, height, and format for the output. The information needed to call *GetMap* can be obtained with a *GetCapabilities*.

Most web mapping services support a handful of other functions. For example, *GetFeatureInfo* sends specific information about locations on the map, such as the name of the road or the height of a location. *GetLegendGraphic* function deals with the symbols used on the map. Below, we summarize three popular Web mapping services.

GOOGLE MAPS

Google Maps[11] is a web mapping service application and technology provided by Google, that powers many map-based services, including the Google Maps website, Google Ride Finder, Google Transit, and maps embedded on third-party websites via the Google Maps API.

OPEN STREET MAPS

OpenStreetMap (OSM)[12] is a collaborative project to create a free editable map of the world. This crowdsourced data is then made available under the Open Database License. The web mapping service is supported by the OpenStreetMap Foundation, a non-profit organization.

BING MAPS

Bing Maps[13] is a web mapping service provided as a part of Microsoft suite of search engines and powered by the Bing Maps for Enterprise framework.

3.2.3 *Selection of Technologies*

The selection of the technologies for the present work is focused on the choice of Web Content Management System because it is important to have the common functionalities of any web portal. Therefore, we selected Drupal because it is a framework highly extensible and customizable. The fact that we have chosen Drupal involves the use of PHP programming language.

11 http://maps.google.com
12 http://www.openstreetmap.org
13 http://www.bing.com/maps

It is difficult to choose a methodology and be sufficiently satisfied with it, because there are a variety of different approaches and objectives, therefore we will use the Scrum agile methodology, because the Web Engineering Methodologies are very rigid in the design process and the time is limited. This does not mean we can not use the UML notation. For example it is possible to rely on a use case diagram for perceiving the actors and the main functionalities of the web portal.

As part of the development methodology, we will use a revision control system (SVN) for the management of the diverse changes of the product, and a project management and bug-tracking tool called Redmine[14].

PostgreSQL was selected as the Database Management System since it is a mature open source system that has an advanced extension, called PostGIS, to handle geographic data.

The selected Web Mapping Service Application was Google Maps because it has a bigger coverage when compared with others. Coverage represents the amount of services or points of interest on the map, such as restaurants, hotels, museums, avenues, or stores.

3.3 MOBILE APPLICATIONS

Mobile applications became more popular in the last few years thanks to a huge increment in the use of mobile devices. From a developer's point of view, mobile applications became an important source of income thanks to the different mobile application repositories, such as App Store, Google Play, Windows Phone Store, and BlackBerry App World.

With an increase in the number of mobile applications available, users demand more features from the applications they are going to use. A solid testing of the mobile application and its security aspects are the key factors in the pursuit of success for a mobile application (Belén and nirola, 2014).

Mobile application is defined as follows (Janssen, 2014):

> A mobile application, most commonly referred to as an app, is a type of application software designed to run on a mobile device, such as a smartphone or tablet computer. Mobile applications frequently serve to provide users with similar services to those accessed on PCs. Mobile applications are generally small, individual software units with limited functionality.

Mobile applications usually help users by making easier to use the Internet on their portable devices. Therefore, the users now expect to access and act upon information wherever and whenever necessary and use common device features such as multimedia, maps, camera, and contacts. In addition, a mobile device offers new capabilities that users cannot access easily through web applications, such as,

14 http://www.redmine.org

touchscreen or GPS navigation.

While web applications offer complex functionality to support a wide variety of business processes, mobile applications are interested in completing specific tasks easily with only a few taps on the device. At the same time, mobile applications can be designed with a focus on the user's specific role, because commonly the mobile phone is used by the same person, which may be the employee or manager.

3.3.1 *Mobile Operating System*

A mobile operating system, also referred to as a mobile OS, is an operating system that operates a smartphone, tablet, PDA, or other mobile device. Modern mobile operating systems combine the features of a personal computer operating system with other features, including a touchscreen, cellular, Bluetooth, Wi-Fi, GPS mobile navigation, camera, video camera, speech recognition, voice recorder, music player, near field communication, infrared blaster, and other biometric sensors (Wikipedia, 2014a).
In the following subsection, we summarize three popular mobile operating systems.

Android

Android, from Google Inc., is based on the Linux Kernel and has the largest installed base worldwide on smartphones. As of September 2014, Android reached 85% of the global smartphone market share (Group, 2014). Most of Android is free and open source, but a large amount of software on Android devices, such as Play Store, Google Search, Google Play Services, Google Music, etc. are proprietary and licensed. Most major mobile service providers carry an Android device. Application programming is primarily done in Java. The Android specific Java SDK is required for development although any Java IDE may be used. Performance critical code can be written in C, C++ or other native code language using the Android Native Development Kit (NDK) (Meier, 2010).

iOS

iOS, from Apple Inc., has the second largest installed base worldwide on smartphones behind Android. As of September 2014, iOS reached 11% of the global smartphone market share (Group, 2014). The iOS mobile operating system is closed source, proprietary, and built on open source Darwin core OS. It is used by the Apple iPhone, iPod Touch, iPad, and second-generation Apple TV, which is derived from Mac OS X. Currently all iOS dev Selection of ices are developed by Apple.

Windows Phone

Windows Phone, from Microsoft, has the third largest installed base on smartphones behind Android and iOS. As of September 2014, Windows Phone reached 3% of the global smartphone market share (Group, 2014). The Windows Phone mobile operating system is closed source and proprietary. Windows Phone includes a completely over-hauled UI inspired by Microsoft's *Metro Design Language*. It includes full integration of Microsoft services, such as OneDrive, Office, Xbox Music, Xbox Video, Xbox Live games, and Bing, but also integrates with many other non-Microsoft services such as Facebook and Google accounts. Windows Phone devices are made primarily by Microsoft/Nokia, along with HTC, and Samsung.

3.3.2 *Selection of a Mobile Operating System*

The mobile operating system selected in the present work was Android because it is the most mature open source OS and it is well documented.

3.4 SOCIAL NETWORKS

Nowadays, the social networks are very effective as diffusers of information. The increasing success of social networks causes the diffusion of a lot of content, such as text messages, pictures, videos, maps, etc. Therefore, it is important that the interactions in these social networks can be replicated in our web portal, or vice versa, in order to increase the level of permanence of its users and to stimulate a feeling of community.

3.4.1 *Twitter*

Twitter[15] is a social network that enables users to send and read short messages called `tweets` where you can follow other users and read `tweets` posted by those users. These `tweets` can be stories, ideas, opinions, or any short text. The integration of user context, such as activities or locations, in a post, has very good results in terms of Information Diffusion (Choudhury et al., 2010), for example, with a `tweet` it is possible to promote a point of interest with thousands or millions of followers.

3.4.2 *Facebook*

Facebook[16] is the social network with the biggest number of active users, where the users can see different types of content from other users and they can share this content. Due to this complex

15 http://www.twitter.com
16 http://www.facebook.com

network, it was analyzed a data set from Facebook and concluded that it is very difficult to determine the influence of a given action (Bakshy et al., 2012). Nevertheless, there are three possibilities: a user shares a content because his friend also shared it, two or more friends share the same content because they visited the same website independently and, third scenario, a friend shares a content within and external to Facebook and other friends shares the same content through the external influence. The analysis of the first and third cases, indicates that most information diffusion on Facebook is driven by simple contagion (Bakshy et al., 2012).

4

WEB PORTAL DESIGN

Summary

In this chapter we present the web portal design. The design began with the application of the Scrum agile methodology during 6 sprints. The result of these sprints were the main user stories, user interface mockups, prototypes and some requirements. The requirements gathering process was further developed, with emphasis on two crucial phases: elicitation and documentation. The elicitation phase relies on techniques such as personas, interviews and conversations with the Ubiwhere company staff, documentation reading, and analysis of similar web portals. The documentation of requirements was done with the Volere template. The stakeholders, the use cases, the domain model, the class diagram and the data model are also presented. Afterwards, the MVC and PAC design patterns are compared.

4.1 AGILE DEVELOPMENT

In the first 6 months of the development of the web portal, it was used the Scrum agile methodology at the Ubiwhere Company. Therefore, 6 `sprints` were performed with each `sprint` lasting for approximately 30 days and always beginning with a planning meeting. During the meeting, the `Scrum Master` (adviser company) and the author of the thesis agreed on which work to be done during the 30 days. Every `sprint` had around 4 `daily scrums`, where we planned the tasks for the week and reviewed the old tasks.

During the first `sprint` were identified the initial requirements of the web portal.

During the second `sprint` it was presented a brief summary about similar web portals and the main user stories. This `sprint` included the following subtasks:

- Reporting the main user stories.
- Comparing the main Web Portals for Touristic Social Communities.

During the third `sprint` it was designed the web portal using mockups. This `sprint` contained the following subtasks:

- Designing a mockup for the homepage.
- Designing mockups to register touristic resources and trips.
- Designing mockups to share touristic resources or trips using social networks.
- Designing mockups to organize a trip.

During the forth `sprint` it was developed a functional prototype of the web portal where it was possible to register touristic resources. This `sprint` included the following subtasks:

- Installation and configuration of Drupal CMS.
- Development of a template to view touristic places.
- Development of a module to register touristic places.

During the fifth `sprint` it was developed a functional prototype of the web portal where it was possible to register trips. This `sprint` contained the following subtasks:

- Development of a template to view trips.
- Development of a module to register trips.

During the sixth `sprint` it was developed a functional prototype of the web portal where it was possible to organize a trip. This `sprint` included the following subtasks:

- Development of a template to view the phases of a trip.
- Development of a module to organize trips.

4.1.1 *User Stories*

The user stories reported during sprint 2 are presented next.

US001 - As a generic user, I can

US001.001: access online information about touristic resources.
US001.002: access suggestions of tours by popularity.
US001.003: share information online through e-mail.
US001.004: create an account.
US001.005: have access to the login or registration through a social network.
US001.006: select a preferred language.

US002 - As a authenticated user, I can

US002.001: change my password.
US002.002: create my trip.
US002.003: record my routes (routes data).
US002.004: add touristic resources (points of iterest or services) to the trip.
US002.005: add multimedia (photos or videos) to a touristic resource or to a trip, at the time of creation or later.
US002.006: add travelers to the trip.
US002.007: print my trip or touristic resource.
US002.008: comment a trip.
US002.009: comment a the touristic resources.
US002.010: share my activity associated with social networks.
US002.011: visualize my trips.
US002.012: associate my web portal account to social networks, such as Facebook, Twitter, or Google+.

US003 - Complete way to organize a trip

US003.001: The user registers into the portal.
US003.002: The user logs in the portal.
US003.003: The user plans his trip, introducing the location and the data.
US003.004: The user invites friends to his trip.
US003.005: The user shares his trip with other users.
US003.006: The user sees a trip made by other users and uses it as the base for creating his own trip.
US003.007: The user associates a touristic resource with photos and/or videos taken during his trip.
US003.008: The user associates his Facebook/Twitter account with the portal account.
US003.009: The visitor, by contributing to the community with the comments or votes accumulates points. The accumulated points will be converted to vouchers or promotions.

After the six months spent at Ubiwhere Company, the design phase continued finalizing the requirements elicitation and performing some UML modeling.

4.2 REQUIREMENTS ELICITATION

This section documents the requirements elicitation process and the techniques used for its realization. Requirements capture, which includes elicitation, negociation/priorization, documentation and validation phases, is a critical part of the software development process, because at this stage are identified and documented the customer needs. We will focus on the two most crucial phases only: elicitation and documentation. To capture all the needs of the web portal, different elicitation technique were applied, namely writing personas, carrying interviews and conversations with the Ubiwhere company staff, analyzing similar web portals, and consulting tourist offices.

4.2.1 *Personas*

Personas is one, of many, existing techniques for requirements elicitation. It is very useful when the system users are not present or available. This technique consists in creating an imaginary personality of a user for which it will collect requirements. The character created has features very similar to the target audience. Therefore, stereotypes of system users are created (Cooper and Reimann, 2003).

The web portal has a very wide audience. Therefore, it is important to satisfy the needs of different users. To achieve this goal through personas, three stereotypes with different nationalities, tastes, and interests were created. The first persona can be described as the typical English tourist who travels with family to discover new cultures. The second persona can be described as the typical Portuguese young man who likes outdoor activities and spends a lot of time with friends. Finally, we attempted to describe the business man who constantly finds in business.

Persona 1

Name: Ayleen

Marital Status: Married

Nationality: English

Age: 45 years.

Occupation: Professor of history and geography

Biography: Ayleen is married to a successful businessman, with whom she had a son, who now is 15 years. It is common since her son was born, a family trip.

She enjoys discovering new cultures, visiting museums and historical areas, so in their travels always carries a camera. She loves to take a good photo and capture the best moments of her trip.

Persona 2

Name: Rodrigo
Marital Status: Single
Nationality: Portuguese
Age: 23 years.
Occupation: Student
Biography: Rodrigo is a typical Portuguese guy that enjoys outdoor activities. He has an adventurous spirit and never refuses a stay amidst nature. He practices sport regularly, like any other young man of his age. He enjoys and appreciates a good game of football. He is currently taking a course in Sport. In college, the people know him for his good disposition. He is a very sociable person.

Attaches great importance to their friends and with them has the most of its activities. He spends most of vacation time.

Persona 3

Name: Jean
Marital Status: Married
Nationality: French
Age: 49 years.
Occupation: Business Manager
Biography: Jean is a business manager, very devoted to business. He is married and has the full support of his wife.

Because of his profession is a man who is constantly in travel and enjoy getting installed comfortably.

He is a very demanding professional man and very informed. He, constantly checks his email, reads many newspapers and uses social networks.

4.2.2 *Interviews and Conversations with the Ubiwhere Company Staff*

This dissertation was developed partially at the Ubiwhere Company, with the objective of integrating the web portal into an existent platform dedicated to tourism.

Ubiwhere Company

The Ubiwhere is a R&D company with a special focus on Networks and Telecommunications. Its main offices are located in the city of Aveiro, maintaining a close relationship with universities by hosting Masters' students and projects development in partnership with the major national universities. Figure 3 shows the areas and business units of Ubiwhere and figure 4 summarizes the products and services they provide.

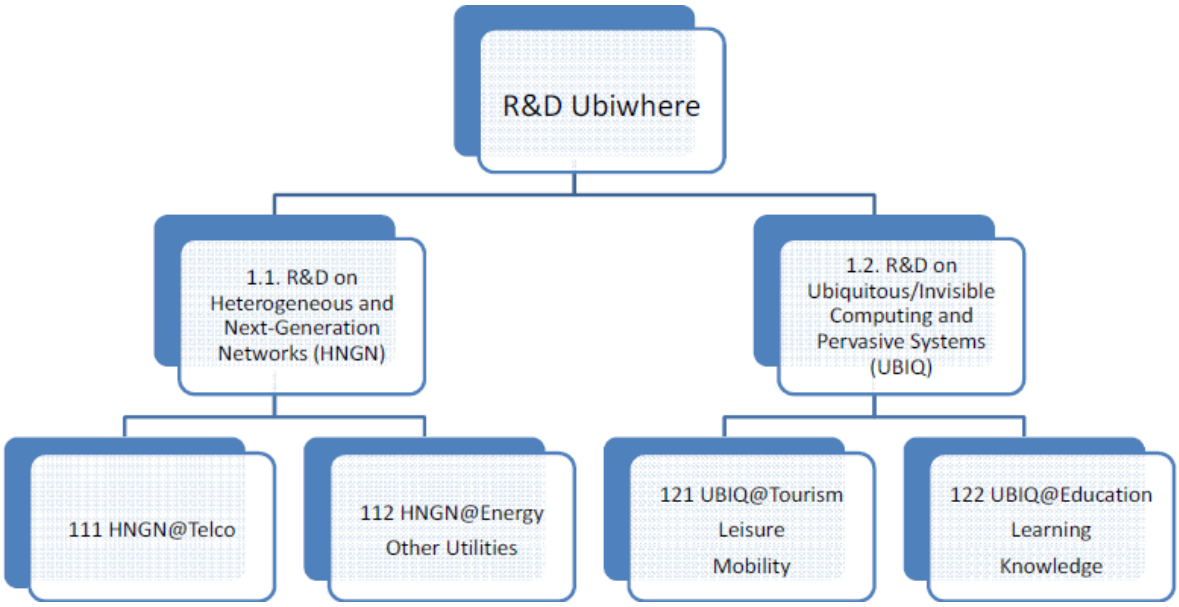


Figure 3.: Areas and Business Units of Ubiwhere Company.

Figure 4.: Products/Services of Ubiwhere Company.

The company's management is composed by three people: André Oliveira, Nuno Ribeiro, and Rui A. Costa (Costa, 2011). There are approximately ten project managers at the company, where my project manager at Ubiwhere is Ricardo Ferrolho, with whom I had several conversations.

Figure 5 shows the organizational structure of Ubiwhere.

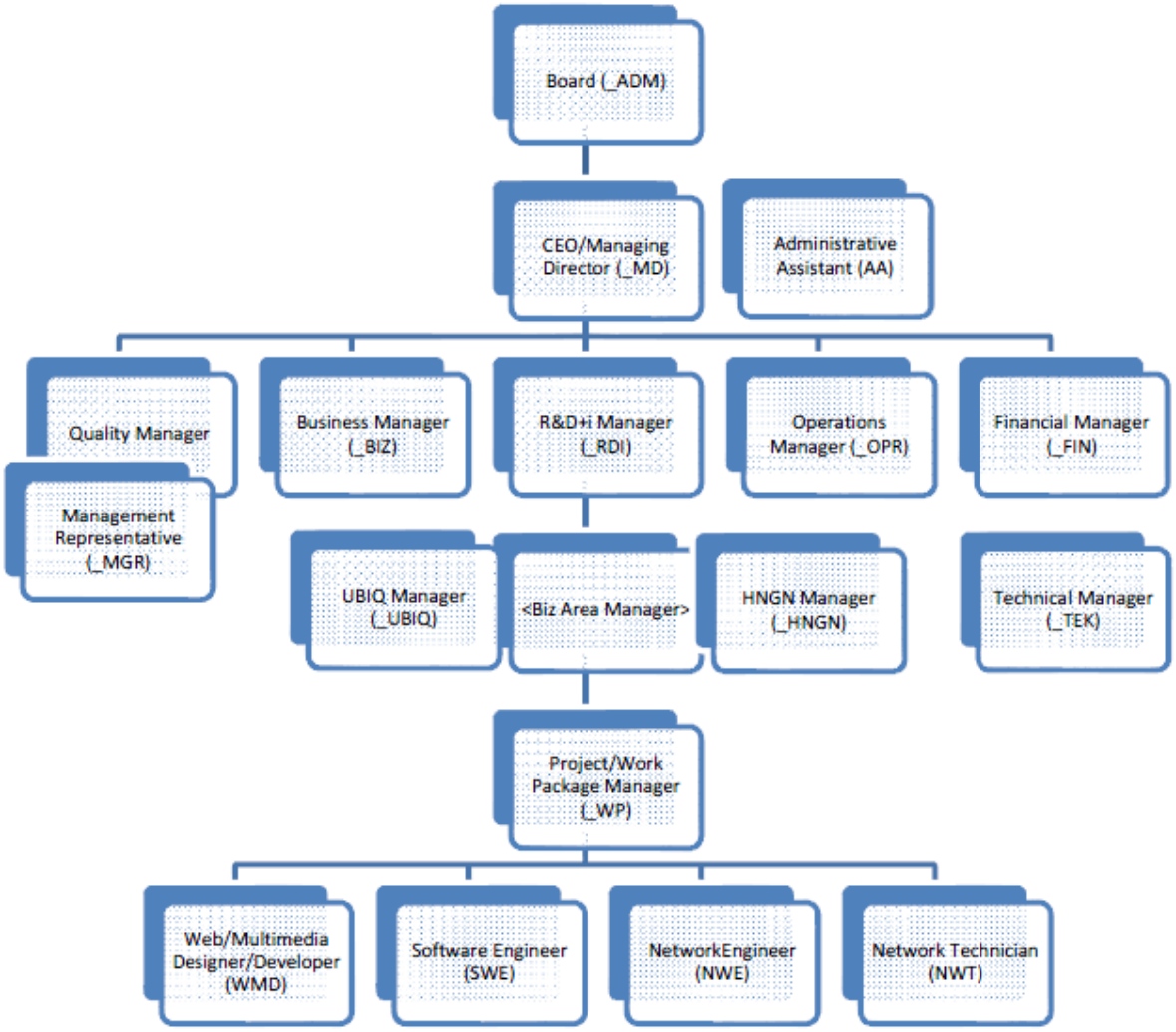


Figure 5.: Organizational Structure of the Ubiwhere Company.

Interviews and Conversations

The meetings are divided into 6 monthly meetings and 18 weekly meetings. There was also a meeting on February 16^{th}, 2012 between the project manager from Ubiwhere and the thesis tutor from the University of Minho.

The first monthly meeting was at December 11^{th}, 2011. At that day the author presented:

- His proposal to develop a touristic web portal where the tourist can plan their trips and define what tourist resources they would like to experiment.
- His service to develop a Front-office where the tourist can organize their trips and develop a Back-office where the administrators can manage the trips and touristic resources.

The second monthly meeting was the day January 30^{th}, 2012. That day the author presented a brief summary of the state-of-the-art speaking about similar web portals.

The third monthly meeting was the day February 29^{th}, 2012. That day it was presented a design about the web portal.

The forth monthly meeting was the day March 26^{th}, 2012. That day it was presented a prototype of the web portal where it was possible to register touristic resources.

The fifth monthly meeting was the day April 23^{rd}, 2012. That day the author presented a prototype of the web portal where it was possible to register trips.

The sixth monthly meeting was the day June 4^{th}, 2012. That day it was presented a functional demonstration of the web portal where it was possible to organize a trip.

4.2.3 *Documentation Reading*

Ubiwhere has a platform in development dedicated to tourism, where the main products/services are Mobile Travel Guide(MTG), Museum Mobile Guide (MMG), Interactive User Experience (IUE), Tourist Tracking System (TTS), Virtual Tourist Card (VTC), and Marketing one-to-one (1to1 MKT). Figures 6 and 7 illustrate the Client-Server and Back Office-Front Office views of the Ubiwhere platform dedicated to tourism, respectively.

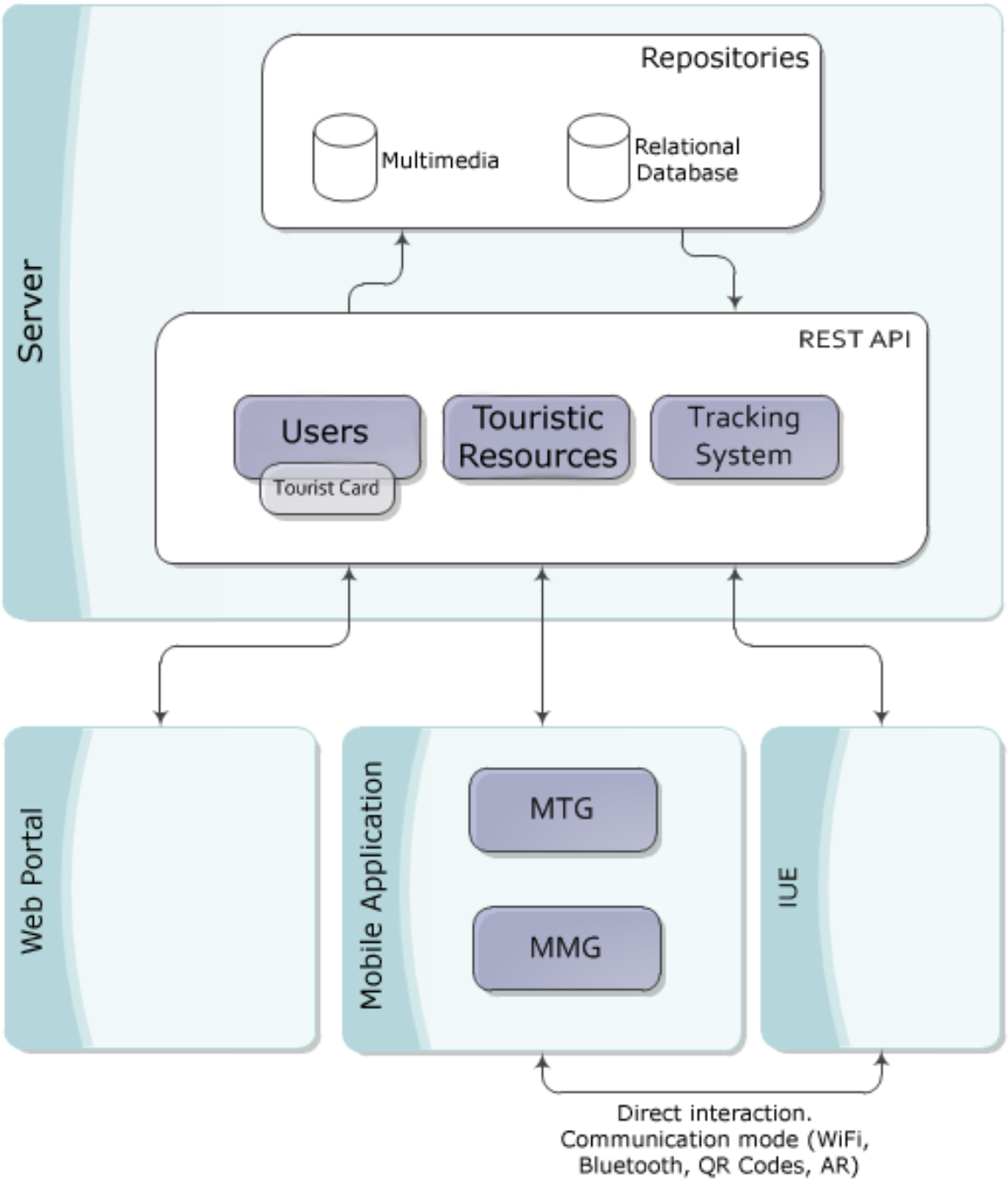


Figure 6.: Platform dedicated to tourism (View Client-Server).

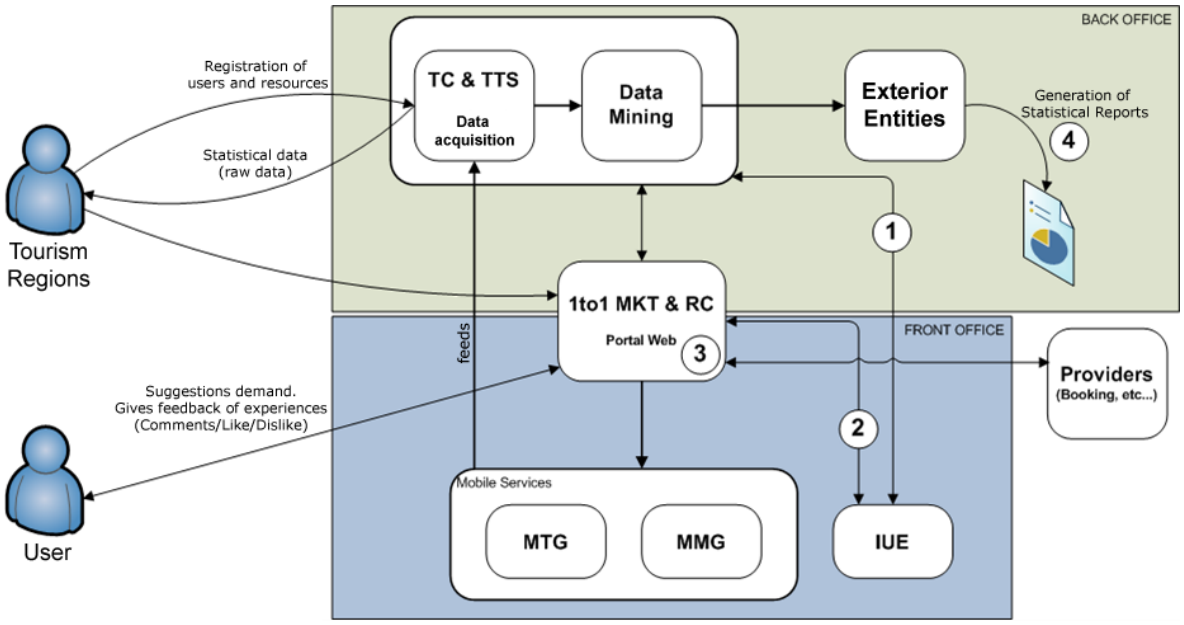


Figure 7.: Platform dedicated to tourism (Divided in Back Office and Front Office).

4.2.4 *Analysis of Similar Web Portals*

The Internet is an essential source of information when someone wants to create a web portal. It is hard to develop a new web portal that has not already been partially implemented or documented and is not available online. Therefore, a relevant requirements elicitation technique consists in analyzing similar web portals with features that could be integrated into our web portal. During the analysis of similar web portals we also found sites, not connected to tourism, but with interesting features for our system. A detailed analysis of web portals was documented in chapter 2. Table 5 just summarizes the main objective of the most relevant portals analyzed.

Web Portal	Description
TripAdvisor	City Guides Catalog that provides reviews of travel-related content.
Yahoo Trip Planner	City Guides Catalog that provides reviews of travel-related content.
TouristEye	Travel guide that provides tips of travel-related content.
TripIt	Trip planner that organizes travel plans into an itinerary that has all of your trip details in one place.
ParisInfo	Touristic web portal of the city of Paris
VisitPorto	Touristic web portal of the city of Porto
VisitDublin	Touristic web portal of the city of Dublin
BuscoUnViaje	Travel Guide generated from trip features.
bedandbreakfast	Travel Guide with accommodation and inclusive breakfast.

Table 5.: Tourism Web Portals.

4.3 REQUIREMENTS DOCUMENTATION

Information gathered during the elicitation phase, through the use of techniques such as personas, interviews, conversations, documentation reading, and analysis of similar web portals, led to the requirements documentation.

The requirements are the desirable needs for a software system, which is mainly divided into functional and non-functional requirements(Young, 2004). The functional requirements are the needs of the customer in the system and non-functional requirements can be defined as restrictions on the solution. In this thesis we used the Volere template (James and Robertson, 2006) to document the requirements. The complete requirements documentation, with Volere template, can be found in Appendix A. Next we present only the description of the requirements.

- **Functional requirements of the web portal**
  - **Requirement 1:** Manage trips, travels, tours, places, and point of interest.
  - **Requirement 2:** Display the details of the trip and navigate on the trip.
  - **Requirement 3:** Display on the map the trip, the point of interest, the attractions, the tours and the events.
  - **Requirement 4:** Search trips and places of interest.
  - **Requirement 5:** Export the trips to PDF format.
  - **Requirement 6:** Allow the user registration.
  - **Requirement 7:** Allow the user management.
  - **Requirement 8:** Allow users to log in and log out.
  - **Requirement 9:** Modify the profile data of the registered user.
  - **Requirement 10:** Rate or comment trips, places, attractions, events, and tours.
  - **Requirement 11:** Display photos and videos of trips, places, events, and tours.
  - **Requirement 12:** Provide statistics of evaluations made by users.
  - **Requirement 13:** Display the most rated trips by users.
  - **Requirement 14:** Consult and manage useful schedules.
  - **Requirement 15:** Consult the weather.
  - **Requirement 16:** Consult traffic bulletin.
  - **Requirement 17:** Allow to download the mobile application.
  - **Requirement 18:** Share trips in social networks.
  - **Requirement 19:** Send a trip by email.

  - **Requirement 20:** Export the trip for GPS.
  - **Requirement 21:** Add trip to favourites.

- **Non-functional requirements of the web portal**
  - **Requirement 22:** Support multiple languages.
  - **Requirement 23:** Provide compatibility of the portal to several web browsers and mobile devices.
  - **Requirement 24:** Be considered attractive and easy to use by 90% of the users.

- **Functional requirements of the mobile application**
  - **Requirement 25:** Download trips.
  - **Requirement 26:** List attractions, events, tours, and useful places.
  - **Requirement 27:** Show the distance and duration of a trip.
  - **Requirement 28:** Post a trip or point of interest on Social Networks.
  - **Requirement 29:** Delete the downloaded trips.
  - **Requirement 30:** Allow the user to login/logout.

- **Non-functional requirements of the mobile application**
  - **Requirement 31:** Performance - Request has response times of less than 2 seconds.
  - **Requirement 32:** Be considered attractive and easy to use by 90% of the users.

4.4 ACTORS

The actors of the system can be divided in:

- **Anonymous user:** This user is an individual who visits the portal web and does not log in using a user ID and password. Therefore, he only has permissions to query the contents of the web portal.

- **Authenticated user:** This is a visitor of the web portal who logs in using a unique user ID and password. This type of visitor has certain permissions, such as organizing a trip and changing his profile. The authenticated user has all the permissions of anonymous user.

- **Administrator:** This actor has access to all the system's functionalities. So, he can modify, add and remove any content of the system. He owns all the permissions of the authenticated user.

The hierarchy of the web portal users is presented in figure 8.

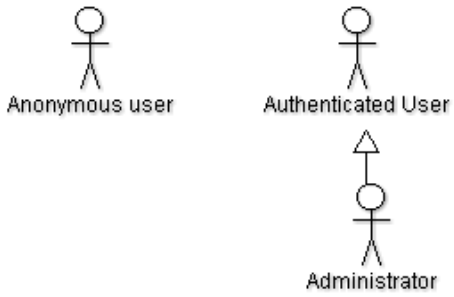


Figure 8.: Hierarchy of the web portal users.

4.5 USE CASES

The use cases can be used to model the system requirements. In the web portal system, use cases were grouped into four packages: `User Administration`, `Trip Management`, `Social Networks`, and `Searching`. The use cases are detailed, including their textual description, in the Appendix B.3.2. The identified use cases are listed next:

- Package `User Administration`:
  - Register via Web Portal.
  - Login via Web Portal.
  - Change the Password.
  - Recover the Password.
  - Logout.
  - Select the Preferred Language.
  - Add User.
  - Edit User.
  - View User Information.
  - Remove User.
  - Login via Mobile Application.
  - View statistics of evaluations.
- Package `Trip Management`:
  - Create Trip.
  - Edit Trip.
  - View Trip.

  - Remove Trip.
  - Add Comment to Trip.
  - Add Comment to Touristic Resource.
  - Add Touristic Resource to Trip.
  - Remove Touristic Resource from Trip.
  - Add Touristic Resource to System.
  - Edit Touristic Resource from System.
  - View Touristic Resource from System.
  - Remove Touristic Resource from System.
  - Add Photo.
  - Add Video.
  - View Video.
  - View Photo.
  - Remove Video.
  - Remove Photo.
  - Export trip to PDF format.
  - Consult useful schedules.
  - Consult the weather.
  - Consult traffic bulletin.
  - Receive the trip by email.
  - Download the trip to Mobile Application.
  - Delete the downloaded trip from Mobile Application.

- Package `Social Networks`:
  - Login via Social Network.
  - Share Trip.
  - Share Touristic Resource.
  - Associate a Social Network.
  - Dissociate a Social Network.
  - Share Automatically.

- Package `Searching`:
  - Advanced Search.
  - Search in all.
  - Search by Category.
  - Search by Places.
  - Display the most rated trips.
  - Add trip to favourites.

The use cases diagram of the package `User Administration` is included in figure 9 and the individual use cases description is presented next.

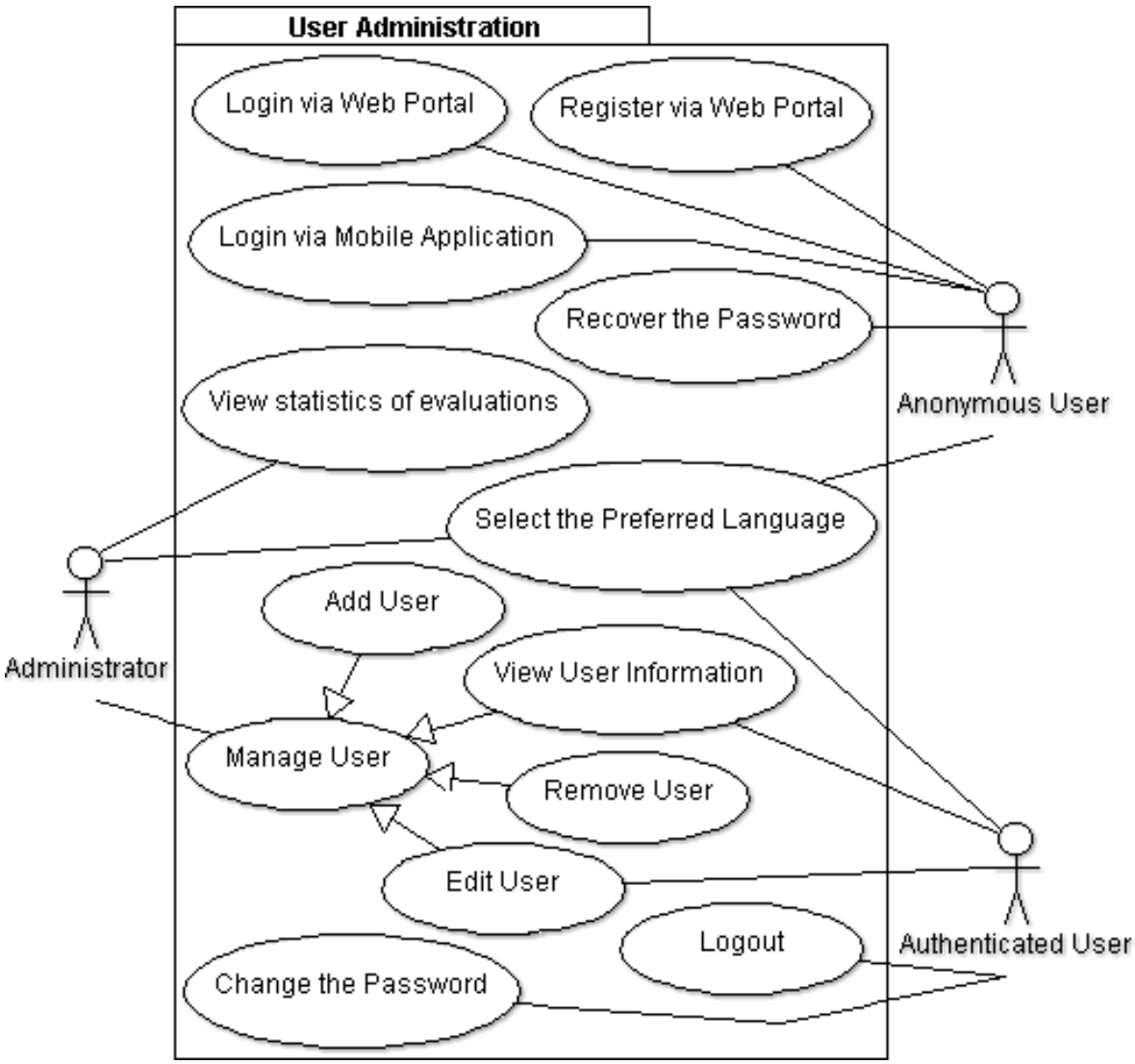


Figure 9.: Use case diagram for package `User Administration`.

Register via Web Portal: This use case allows a user to create an account in the web portal.

Login via Web Portal: This use case allows a user to login in the web portal.

Login via Mobile Application: This use case allows a user to login via mobile application.

Recover the Password: This use case allows an authenticated user to recover his password.

View statistics of evaluations: This use case allows the administrator to visualize the statistics about trips or touristic resources.

Select the Preferred Language: This use case allows a user to change the language of the web portal.

Add User: This use case allows the administrator or an anonymous user to add an account to the web portal.

View User Information: This use case allows an authenticated user to view his profile information.

Remove User: This use case allows an authenticated user to remove his account.

Edit User: This use case allows an authenticated user to edit his account.

Change the Password: This use case allows an authenticated user to change his password.

Logout: This use case allows an authenticated user to logout from the web portal.

The use cases diagram of the package `Trip Management` is included in figure 10 and the individual use cases description is presented next.

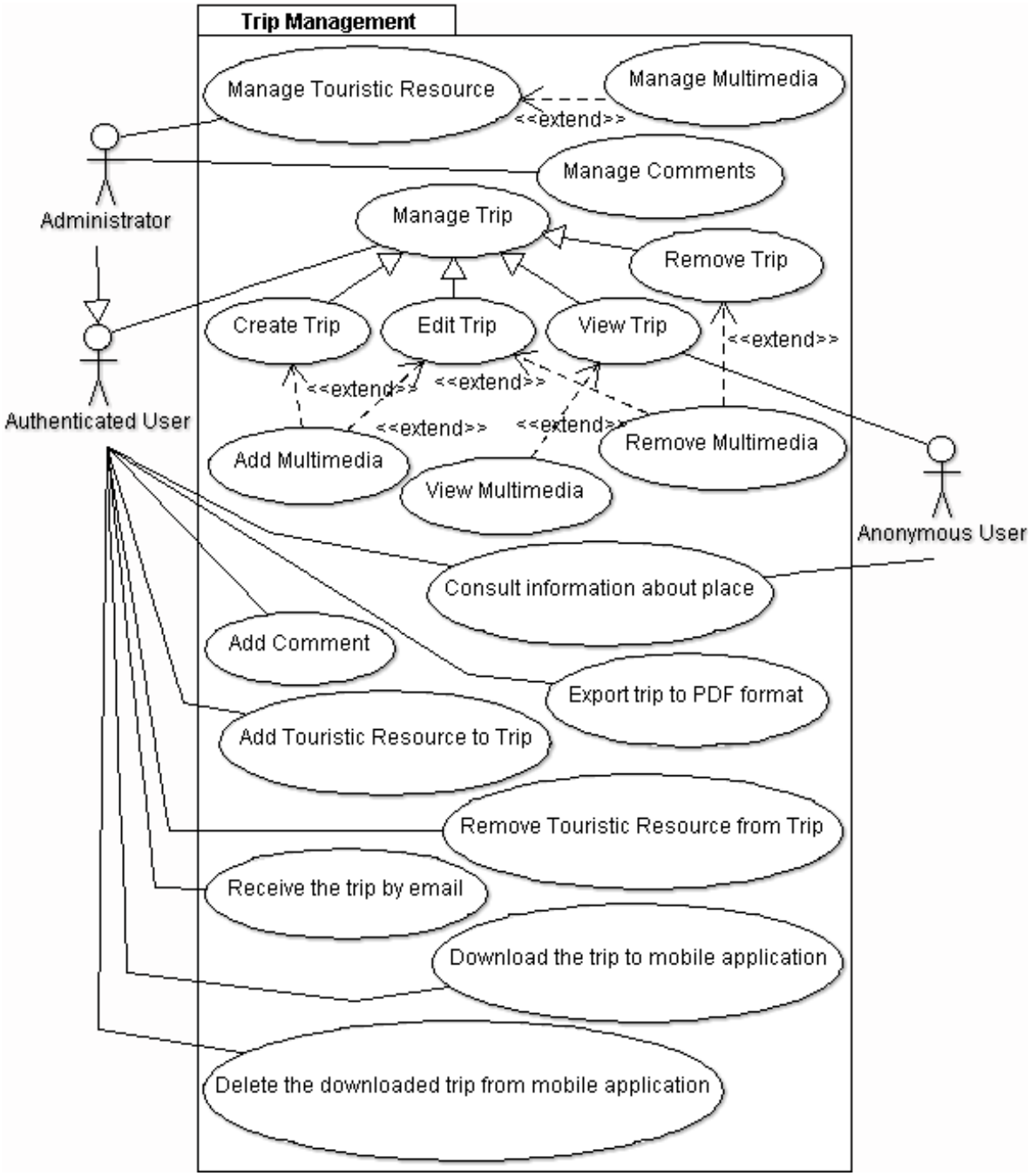


Figure 10.: Use case diagram for package `Trip Management`.

Manage Touristic Resource: This use case allows the administrator to create, edit, view, or remove touristic resources.

Manage Multimedia: This use case allows an authenticated user to create, edit, view, or remove videos or photos.

Manage Comments: This use case allows an authenticated user to create, edit, view, or remove comments.

Create Trip: This use case allows an authenticated user to create a trip.

Edit Trip: This use case allows an authenticated user to edit a trip.

View Trip: This use case allows a user to visualize a trip.

Remove Trip: This use case allows an authenticated user to remove a trip from the portal.

Add Multimedia: This use case allows an authenticated user to add videos or photos to a trip.

View Multimedia: This use case allows a user to view photos or videos about a touristic resource.

Remove Multimedia: This use case allows an authenticated user to delete videos or photos from a trip.

Consult information about place: This use case allows a user to consult information about a place such as schedules, weather, or traffic bulletin.

Add Comment: This use case allows a user to add comments to a trip or touristic resource.

Export trip to PDF format: This use case allows an authenticated user to export a trip to PDF format.

Add Touristic Resource to Trip: This use case allows an authenticated user to add touristic resources to a trip.

Remove Touristic Resource from Trip: This use case allows an authenticated user to remove touristic resources from a trip.

Receive the trip by email: This use case allows an authenticated user to receive a trip by email.

Download the trip to mobile application: This use case allows a user to download a trip to his

mobile application.

Delete the downloaded trip from mobile application: This use case allows an authenticated user to delete a downloaded trip.

The use cases diagram of the package `Social Networks` is included in figure 11 and the individual use cases description is presented next.

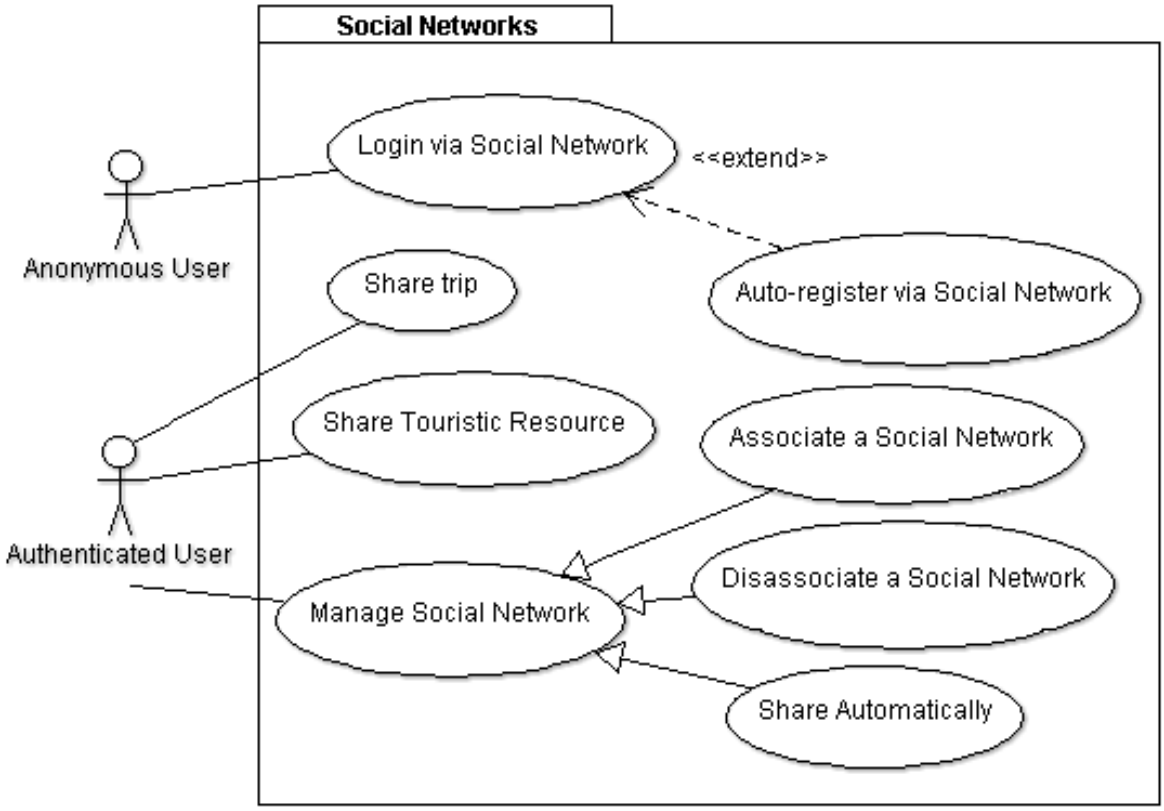


Figure 11.: Use case diagram for package `Social Networks`.

Login via Social Network: This use case allows an anonymous user to login in the web portal via a Social network such as Facebook or Twitter.

Auto-register via Social Network: This use case allows an anonymous user to create an account in the web portal using a social network such as Facebook or Twitter.

Share trip: This use case allows an authenticated user to share a trip on social networks.

Share Touristic Resource: The use case allows a user to share the touristic resource on social net-

works.

Associate a Social Network: This use case allows an authenticated user to associate his web portal account with his social network account.

Disassociate a Social Network: This use case allows an authenticated user to disassociate his social network account from his web portal account.

Share Automatically: This use case allows an authenticated user to share his trips automatically via social networks.

The use cases diagram of the package `Searching` is included in figure 12 and the individual use cases description is presented next.

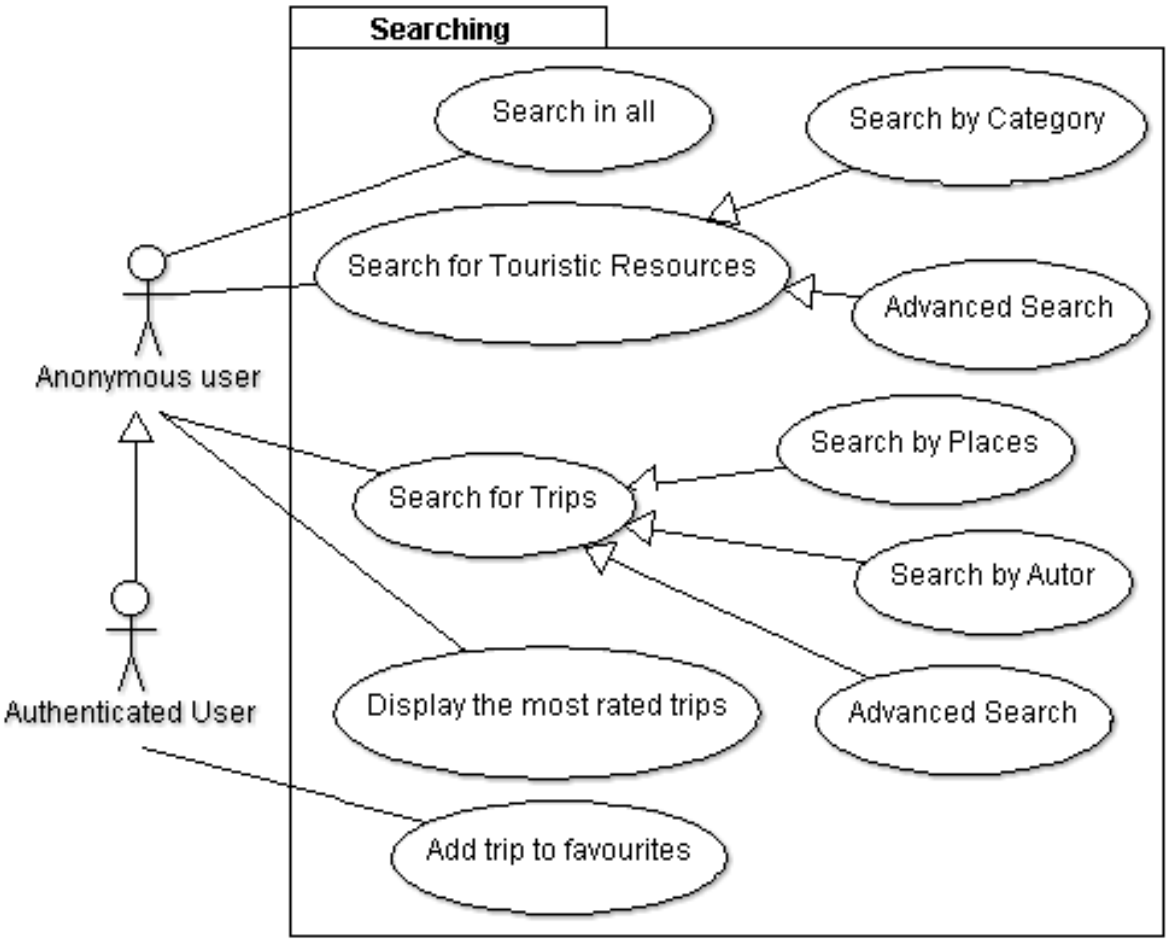


Figure 12.: Use case diagram for package `Searching`.

Search in all: This use case allows a user searching for touristic resources or trips.

Search for Touristic Resources: This use case allows a user to search for touristic resources.

Search for Trips: This use case allows a user to search for trips.
Display the most rated trips: This use case allows a user to visualize the most rated trips.

Add trip to favourites: This use case allows an authenticated user to add a trip to his favourites.

4.6 DOMAIN MODEL

To draw a global picture of the main entities involved in the Web Portal for Touristic Social Communities, and to understand their relationships, it was elaborated the domain model shown in Figure 13. The model follows the recommendations from (Rosenberg and Stephens, 2007) and includes the entities that are candidates to become classes of the implementation.

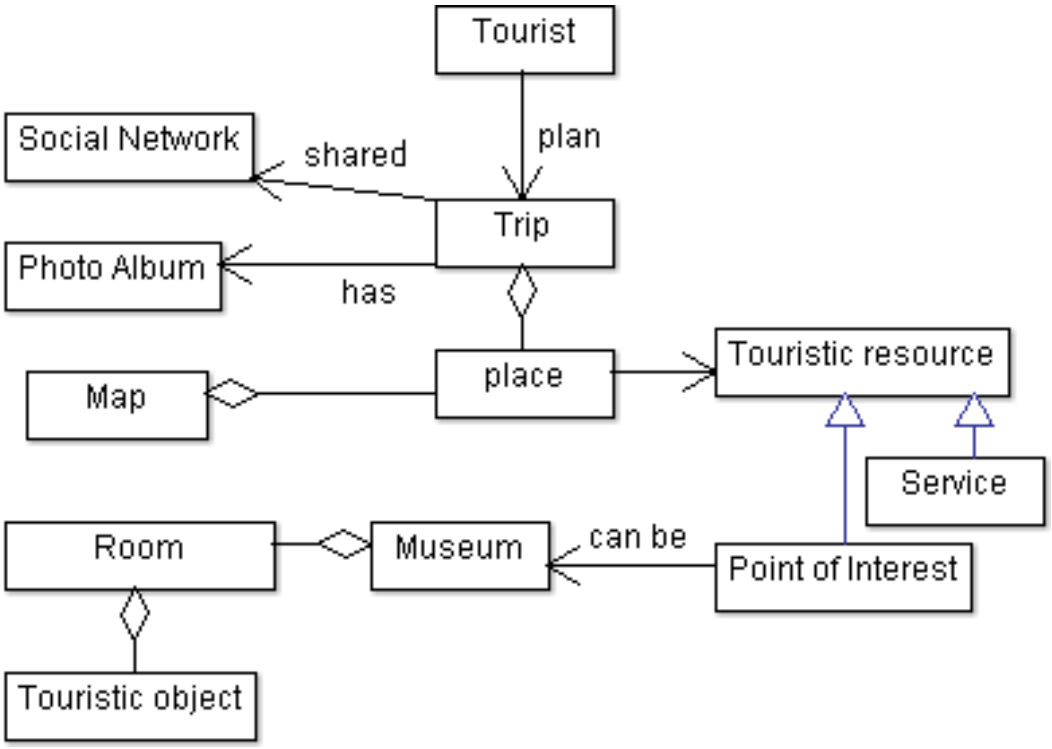


Figure 13.: Domain model of the Web Portal for Touristic Social Communities.

4.7 CLASS DIAGRAM

The next step in the web portal design was to elaborate its class diagram. The class diagram models the structure of the system. We can draw conceptual and implementation class diagrams, with more or less implementation details. The implementation details come from the amount of attributes and methods included in the classes. To evolve from a conceptual to an implementation class diagram, we can drop some conceptual classes and merge some classes into one. In the design phase, it was drawn a conceptual class diagram, where classes include attributes only. The result diagram is shown in figures 14 and 15.

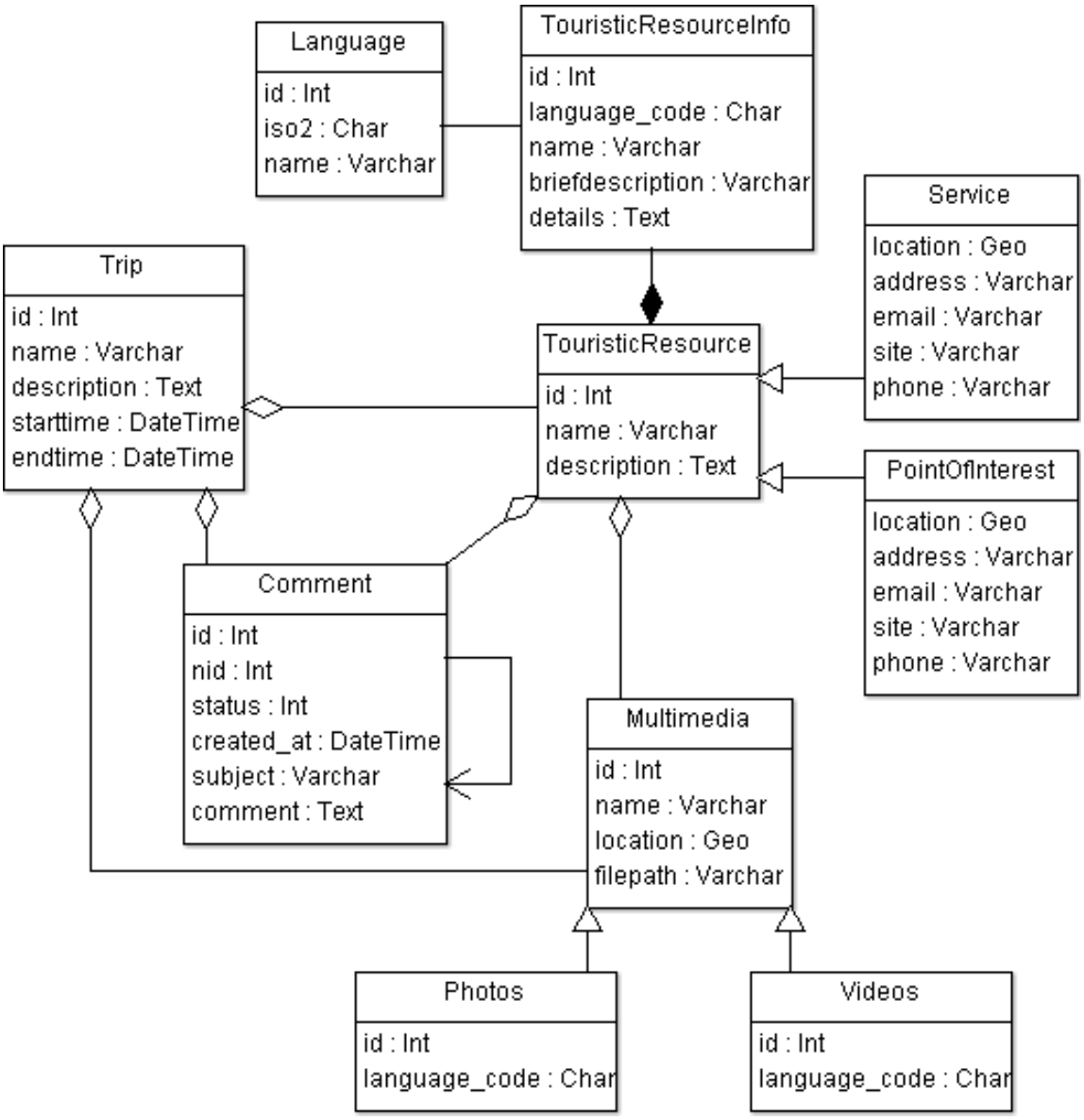


Figure 14.: Web Portal class diagram (part 1).

The main class is Trip, which can have one o more Touristic Resources. A Trip also has a name, a start-date, and a end-date. In turn, to structure and organize the sources of tourist information, a Touristic Resource can be specialized into Point of Interest or Service. A Touristic Resource can have geo-location information, address, e-mail, website, telephone, and multimedia contents, such as photos or videos. Both, the Trips and the Touristic Resources can have comments. The comments have features, such as the subject, the creation date, the status, and the name of the author. The Touristic Resource information can be displayed in one or multiple languages, and it is identified by two-letter country codes defined in the ISO 3166-1.

A User is characterized by an username, password, and e-mail. A user can have different types of Role, such as anonymous user, authenticated user, and administrator. When a user is registered in a Session, it is not necessary to authenticate again while the Session is active.

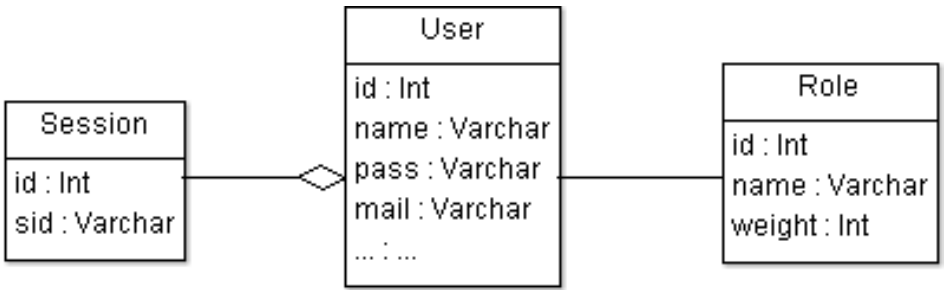


Figure 15.: Web Portal class diagram (part 2).

4.8 DATA MODEL

In this section we will present the tables of the data model derived from the class diagram. The total number of tables is one hundred and six, from which eleven tables were developed. Few tables were created because the Drupal database has many tables that can be reused. For example, the tables related to user management from Drupal were reused to register potential tourists. The node table, the main feature in Drupal, was reused to register tourist content such as trips and touristic resources. Finally, the Drupal tables related to menus and blocks management were reused to place the tourist resources in the web portal. Figures 16 until 21 illustrate the data model tables of the web portal. Figure 16 shows the tables related to the trip management, where a service can be categorized into restaurants, entertainments, accommodations, hospitals or other service. In turn, a restaurant has specific features such as the specialities and prices. An entertainment has features such as the schedule and prices. An accommodation has features such as the number of rooms and the prices. Finally, a hospital has features such as the schedule and the medical specializations.

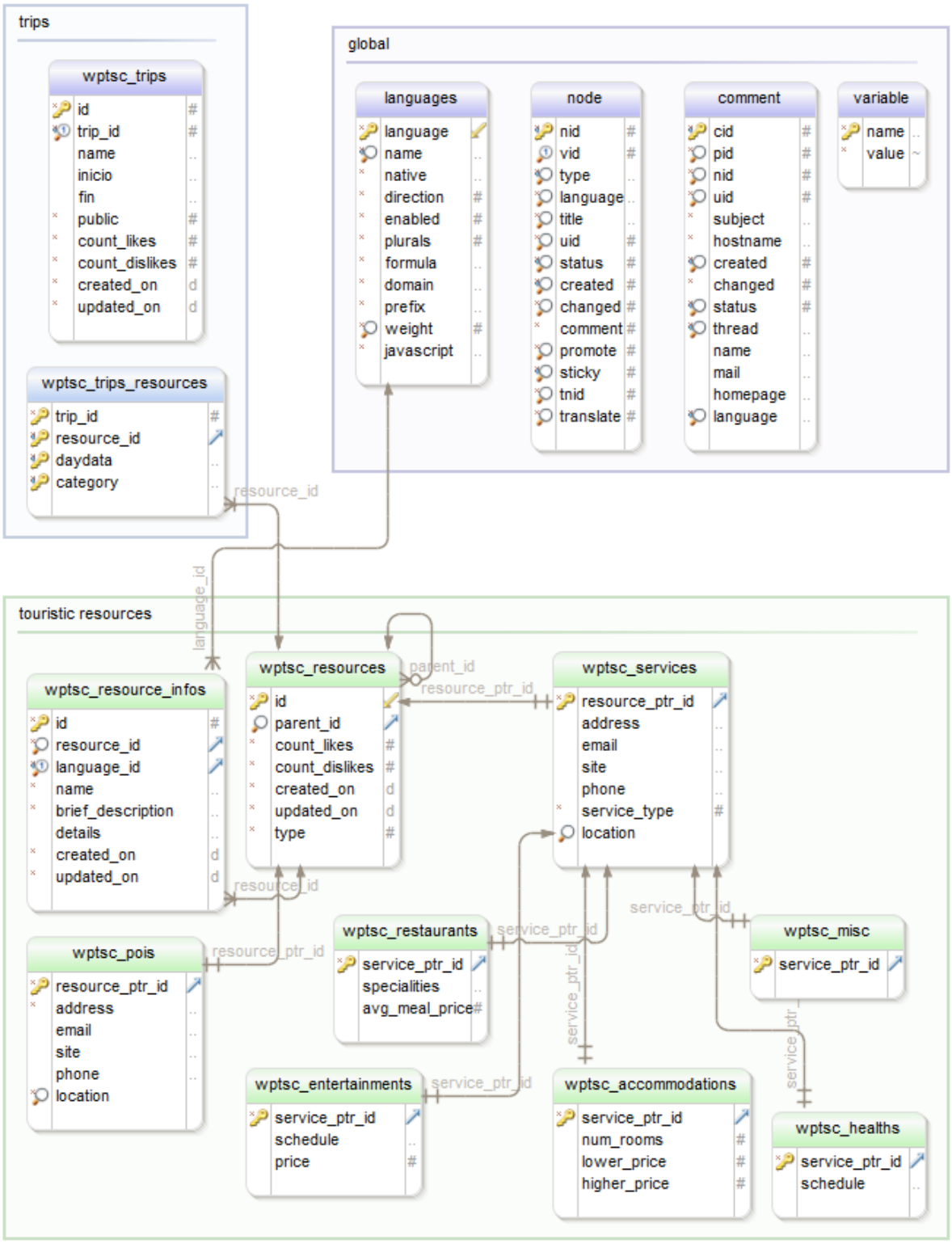


Figure 16.: Tables related to the trip management.

The tables of user administration allow to manage users, roles, and permissions. The basic user roles are anonymous user, authenticated user, and administrator. Is possible to assign permissions to a role and also to add or remove a user from a role. In turn, the user has features such as the name, password, e-mail, creation date, status, last access date, and language.

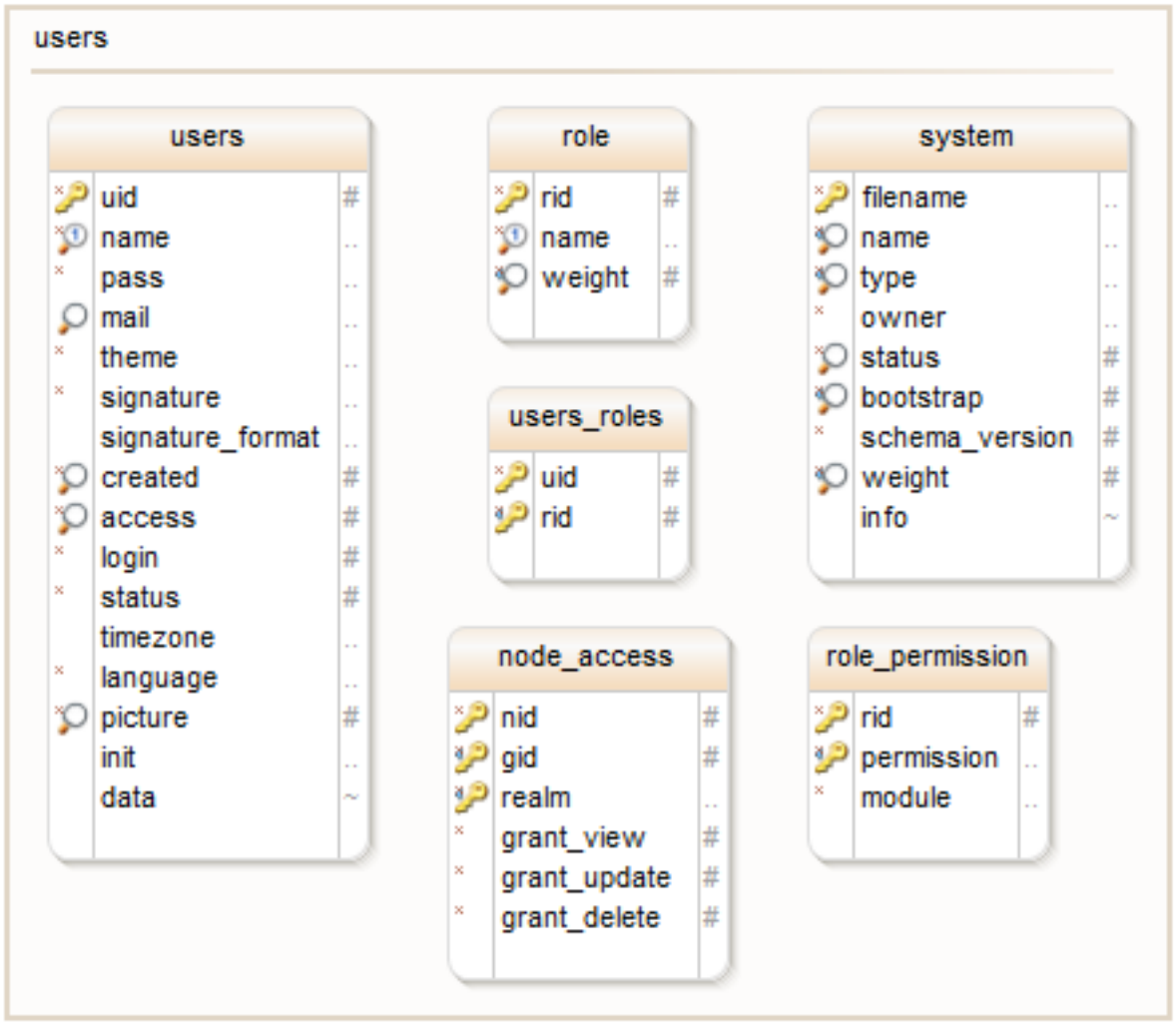


Figure 17.: Tables related to the user administration.

The system menus allows callback mapping, access control, and menu customization. The callback mapping allows converting URLs to functions. It is possible to add, remove and rename menus and items from menus. It is also possible to specify the content of the menu items and configure a special block for a menu. In turn, the menu has features such as the title, description, type, link path, depth, visibility, and weight.

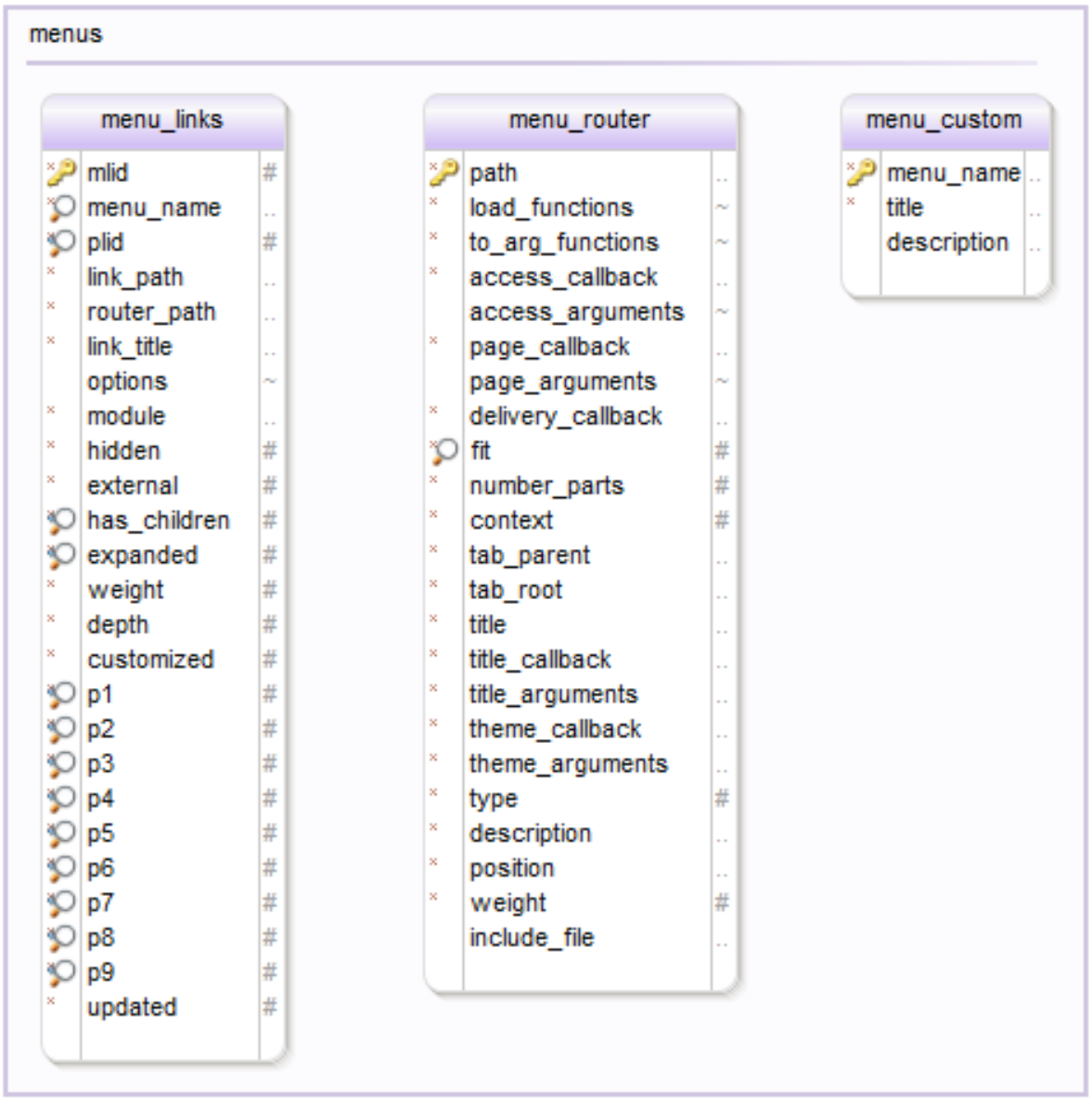


Figure 18.: Tables related to the system menus.

Blocks are the boxes of content that can be placed in different regions of the web portal, such as the footer, the header, the search region, the main content, the content top, the content bottom, the sidebar left or sidebar right. Therefore, the regions of the block are defined by the Drupal theme template. Once created, a block can be modified to adjust its type, title, visibility, appearance, shape, size, and position.

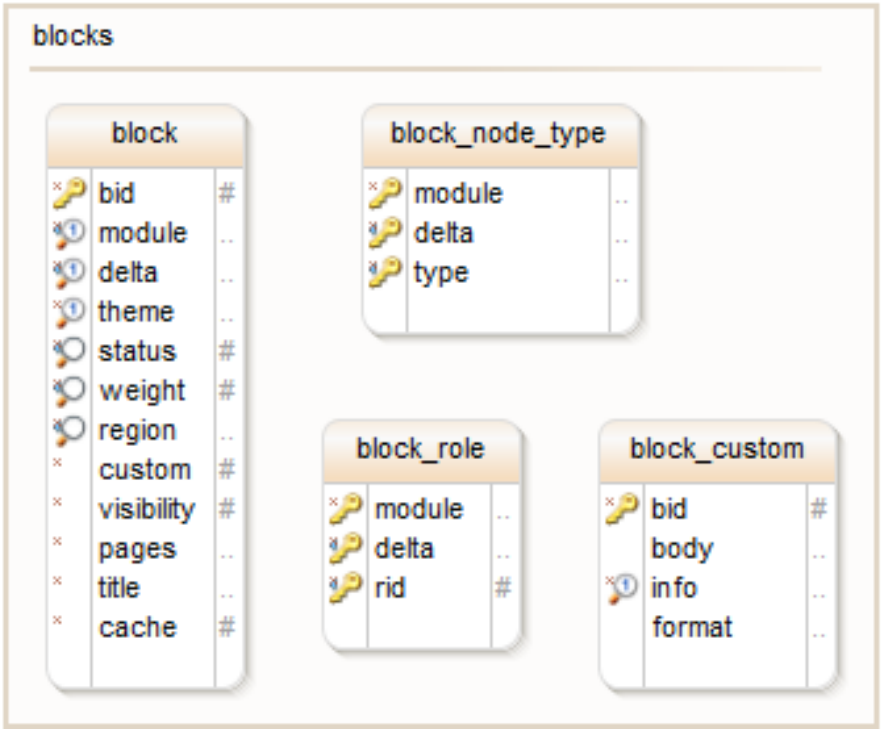


Figure 19.: Tables related to the blocks content.

System cache allows to retrieve information about the saved content, the site settings, and the current user, without numerous trips to the database, avoiding the processing of the retrieved information from the server.

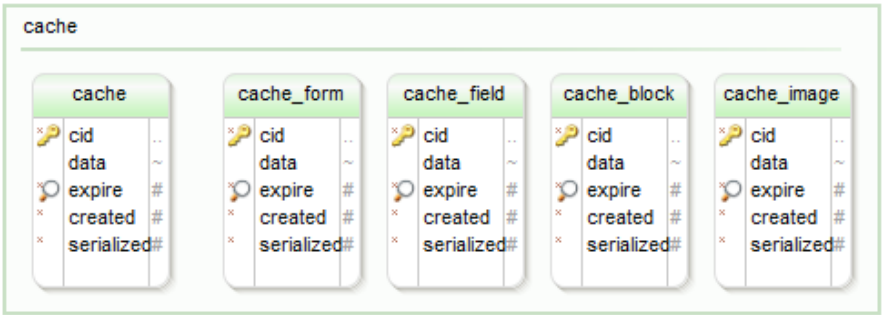


Figure 20.: Tables related to the system cache.

The Drupal data fields allow to create content types using fields either programmatically, by creating a module, or through the Drupal administrative interface, by creating a new content type and assigning fields through the user interface. There are different types of fields, such as input, text area, checkbox, calendar, image, video, etc.

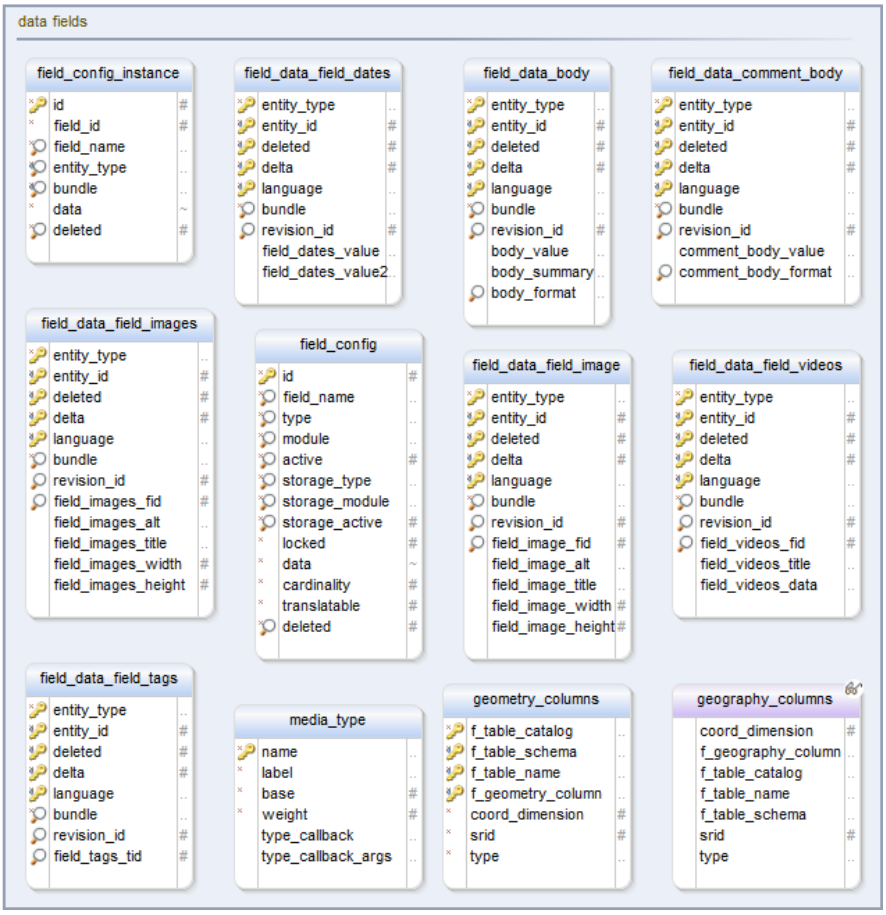


Figure 21.: Tables related to the data fields.

4.9 DESIGN PATTERNS

A design pattern is a formal way of building a solution to a design problem in a particular field. The idea was introduced by the architect Christopher Alexander in the field of architecture (Christopher, 1977). There are two important architectures in web developing, the Model-View-Controller (MVC) architecture and the Presentation-Abstraction-Control (PAC) architecture.

4.9.1 *Model-View-Controller*

The most commonly-known design pattern is MVC (Rosenberg and Stephens, 2007). Most desktop applications with professional aspect use MVC or a variant of it. MVC is a design pattern for computer user interfaces that separates data from an application, the user interface and business logic into three distinct components:

- **Model** is a object representation of the data and its functionality, usually it is used to connect with the database, executing CRUD operations (Create, Read, Update, Delete).
- **View** observes the state of the system and generates the output (presentation) to the users. The view commonly refers to a template.
- **Controller** translates user input commands into operations on the model. Controllers are commonly a connection between the view and the model with the responsibilities of communication and coordination. When a request is received, the controller manages (inserts, gets, or updates) the data from the model, it decides which view to show to the user, and hands the requisited data to the view.

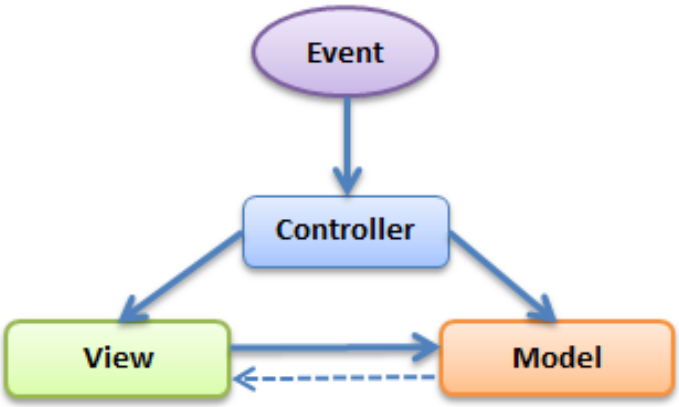


Figure 22.: MVC diagram.

Figure 22 shows the relation between the model, the view and the controller. Note, that the solid lines indicate a strong association and the dotted lines a weak association. It is possible to observe that:

1. Model has a weak association pointer (dotted line) to the View, which means the Model should not know the exact type of View.
2. View has a strong association pointer (solid line) to the Model, which means the View must know the Model.
3. Controller has strong association pointers (solid lines) to both the Model and View, which means it must know the type of Model and View instances.

The pointer analogy can be considered as access level. An object with strong association pointer to another object has full control over it while one with weak association pointer has access to only base class members and not members of object. One Model can have many Views or Controllers.

Usually, in the MVC of web applications (i) the View is composed by HTML and CSS files, (ii)

the Controller is composed by dynamic web pages, such as PHP, JSP, or ASP files, and (iii) the model is usually stored in database or XML files.

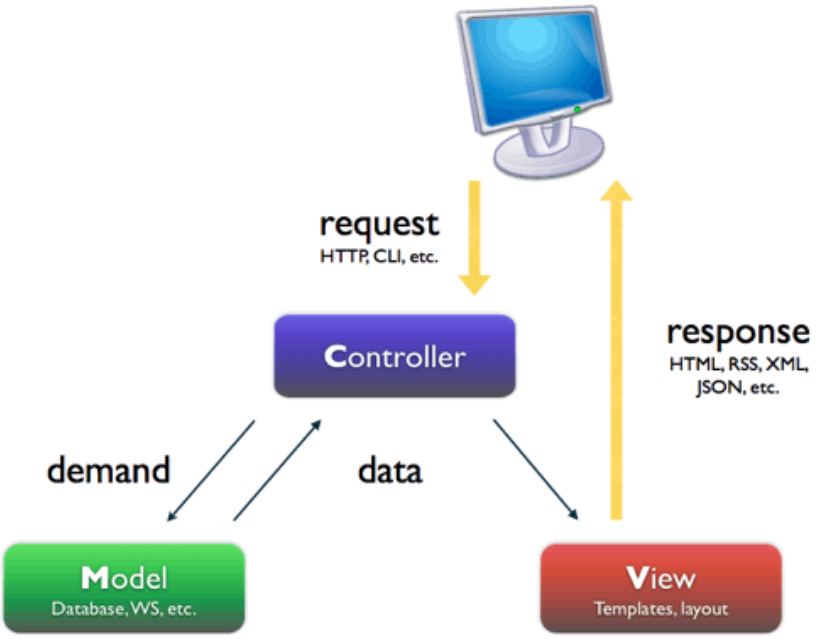


Figure 23.: MVC behavior.

In figure 23 it is possible to see the behaviour of MVC, where the computer sends a request to the controller. Then, the controller interacts with the model, making a demand. Next, the model returns some data to the controller. Then, the controller analyses the results, and sends the data to the view, which is interpreted by the web-server, and it is displayed in the browser.

4.9.2 *Presentation-Abstraction-Control*

PAC is a design pattern similar to MVC. However, PAC is used as a hierarchical structure of agents, or triads, that react to the changes in their environment. Each agent has following components (Buschmann, 1996):

- **Presentation** observes the system state and generates the output (view) to the users. The presentation commonly refers to a template.
- **Abstraction** contains the data and its functionality, as in MVC. However, it may be only part of the entire data structure of the application.
- **Control** connects the presentation and abstraction components. The Control provides the functionality that allows the agent to communicate with other PAC agents. The Control of PAC is

different from the control of MVC, in a way that the child PAC component sends its occurred changes to its parent PAC component.

Levels of PAC

Top-level PAC agent provides the core functionality of the system. For example, it includes the parts of the user interface that cannot be assigned to any particular subtask, such as the menu bars or a dialogue box displaying information about the application. Some PAC agents do not implement a presentation component or an abstraction component.

Bottom-level PAC agents represent a specific semantic concept of the application domain, on which users of the system can act, such as spreadsheets and charts. The bottom-level agents present these concepts to the user and support all the operations that users can perform on these agents, such as zooming or moving a chart. Users of the system only interact with bottom-level agents.

Intermediate-level PAC agents represent either combinations of lower-level agents, or relationships between lower-level agents. For this reason, the intermediate level is optional. For example, an intermediate-level agent may maintain several views of the same data, such as a bar chart and pie chart of a dataset. In figure 24 we can observe the 3 levels of the PAC hierarchy.

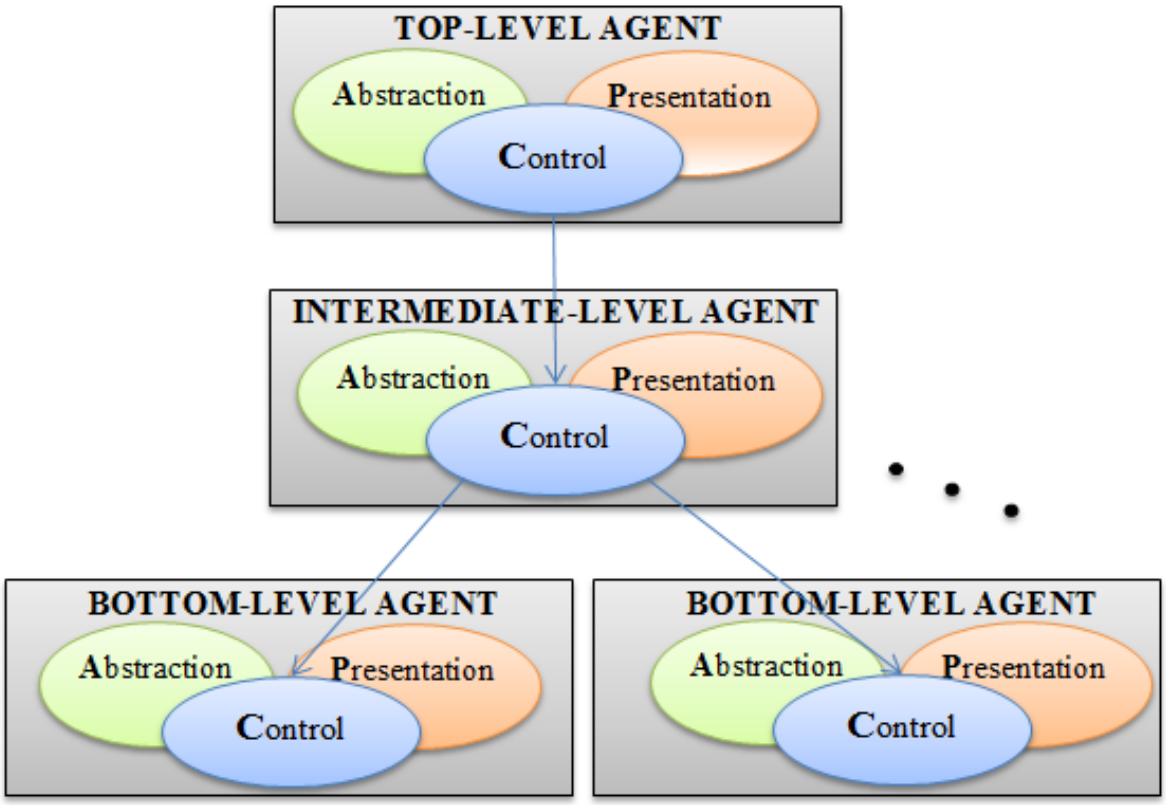


Figure 24.: The three levels of PAC.

PAC Working

Figure 25 gives an example of how PAC works. The Control of the top-level agent creates its child PAC agents, either at program startup or dynamically at run-time. When the control of a PAC agent receives an event (1), it may update its Presentation (2a) and/or its Abstraction (2b). Then it sends a change event to its parent (3). The parent updates its children, except the child agent where the change was originated (5). Thus, all children update their Presentation (6a) and/or Abstraction (6b). After the children have been updated, the parent is updated (7). The activity ends when all necessary PAC agents have been updated.

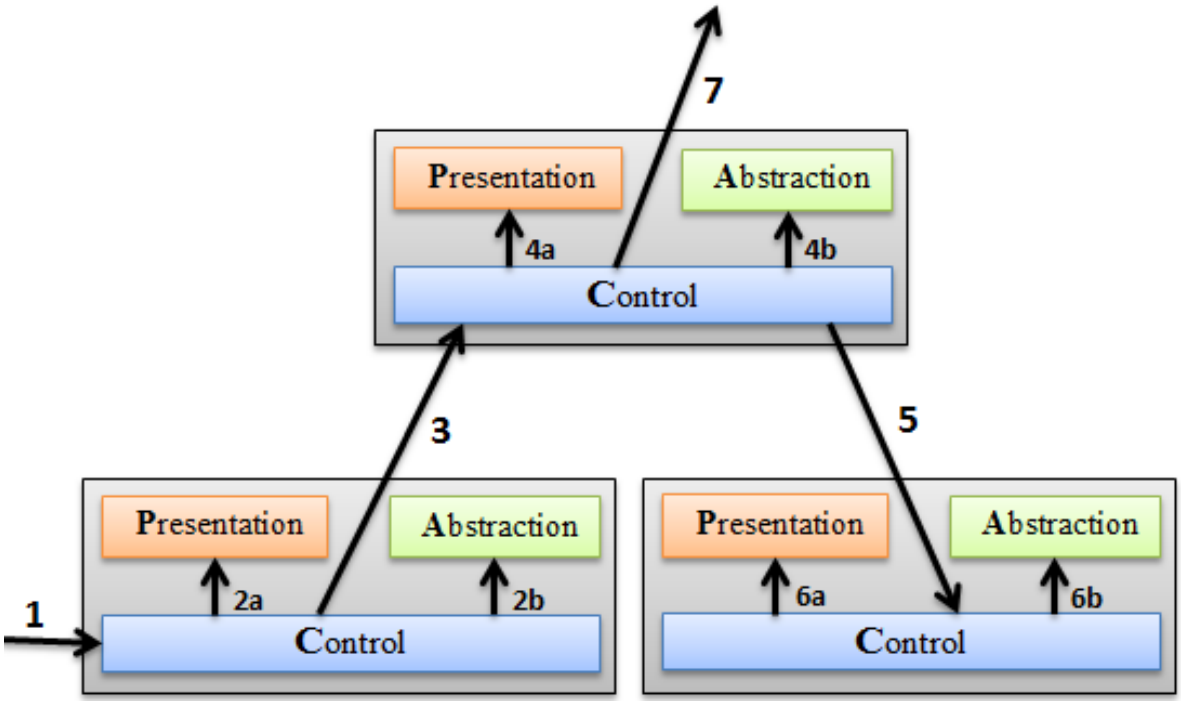


Figure 25.: PAC Working.

4.9.3 *Comparison between PAC and MVC*

The MVC is commonly restricted to a simple graphical user interface (GUI), with one or more views layer operating on the same model layer. The PAC architecture does not have the model layer as its main component, but a hierarchical structure of PAC agents. The abstraction component of a PAC agent contains the data, in a similar way to the model layer of MVC. However, it may be only part of the data structure of the complete application, and does not play an active role in the notification of changes. The separation between the view (or presentation) and the data model in PAC is greater than in MVC.

PAC agents communicate with each other only through the control of each triad. PAC also differs from MVC within each triad, since PAC completely isolates the presentation (view in MVC) from the abstraction (model in MVC). This separates the model and the view, which gives the user a faster user experience, because the interface (presentation) can be displayed before the abstraction is completely initialized.

The common purpose of the MVC and PAC is to separate the data (business logic) from its presentation (the view). This allows changes to the view without requiring changes to the model.

4.9.4 *PAC in Drupal*

Drupal has an architecture where the `blocks` are top-level PAC agents, the `nodes` are intermediate-level PAC agents, and our own modules as `trips` and `touristic resources` are bottom-level PAC agents (see Figure 26).

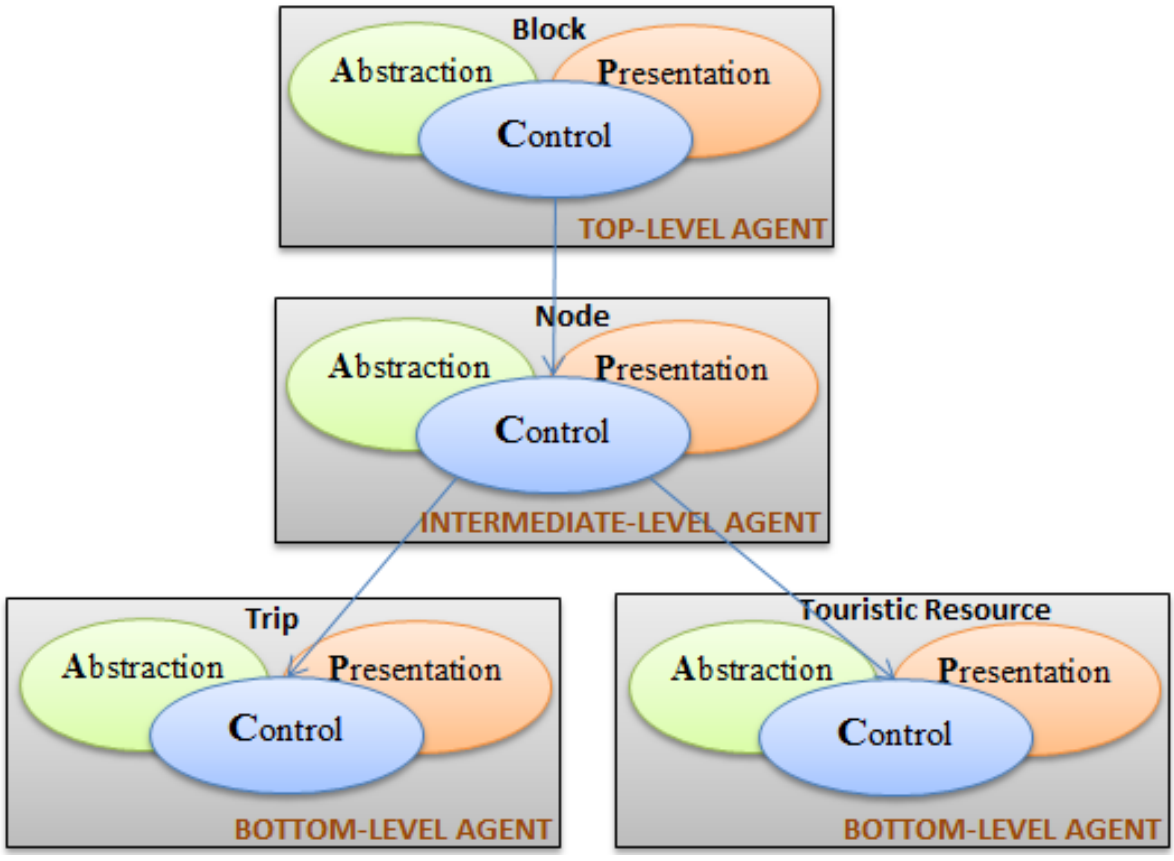


Figure 26.: An example of PAC in Drupal.

The way PAC works in Drupal can be illustrated by an example where a touristic resource is displayed on the web portal (see Figure 27). First, the Control of a `node` receives an event (1), such as

a HTTP Request, and it updates its Abstraction (2). Then the Control of the `node` sends the change event to its child (3). The child `touristic resource` updates its Abstraction (4) and Presentation (5). After the child has been updated, the Control returns the change event to its parent (6). The parent `node` updates its Presentation (7). Then Control of the `node` sends the change event to its parent (8). The parent `block` updates its Abstraction (9) and Presentation (10). The activity ends when all necessary PAC elements have been updated.

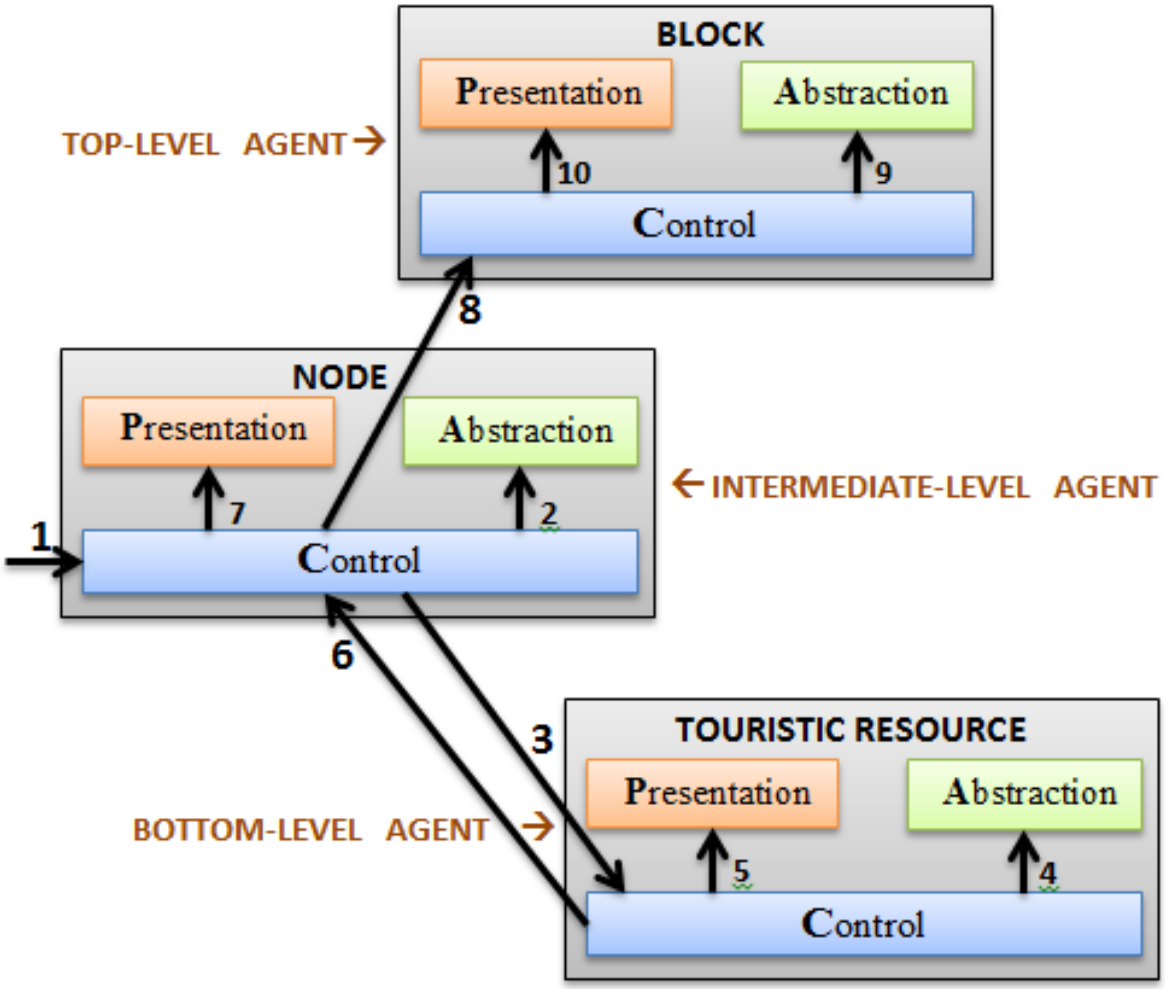


Figure 27.: An example of how PAC works in Drupal.

5

WEB PORTAL AND MOBILE APPLICATION IMPLEMENTATION

Summary

This chapter presents the implementation of the web portal and mobile native application, and the achieved results. It will be described the main features and how they were implemented. Finally, it is presented a REST API that is responsible for synchronizing the web portal with the mobile application.

5.1 OVERVIEW

Both the web portal and mobile application make available the same content to the users, which is provided by a web server (figure 28). In case of the web portal, it is used the HTTP protocol and the server will respond with HTML content. On the other hand, the mobile application makes the request via the developed REST API and receives the response in JSON format. In order to understand the architecture of the web portal implemented in Drupal, it was stratified in two sides, the client and server, presented in the figure 29.

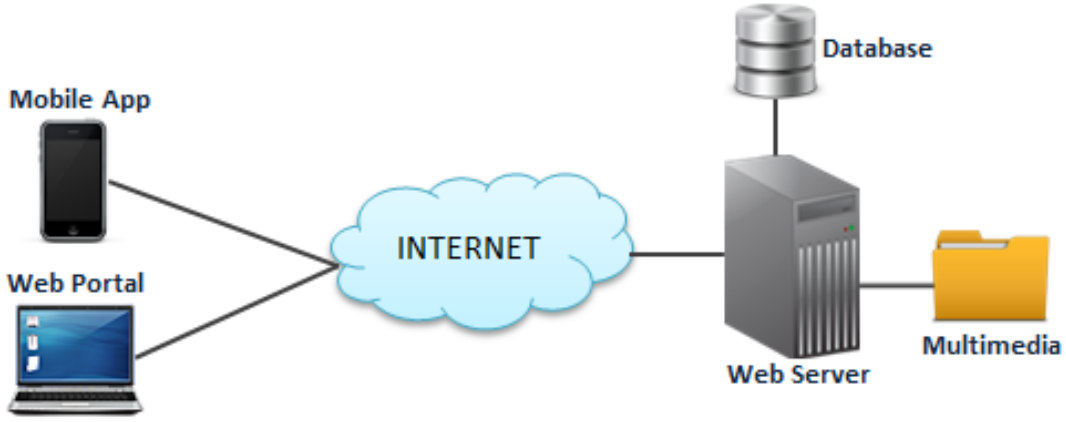


Figure 28.: Overall system architecture.

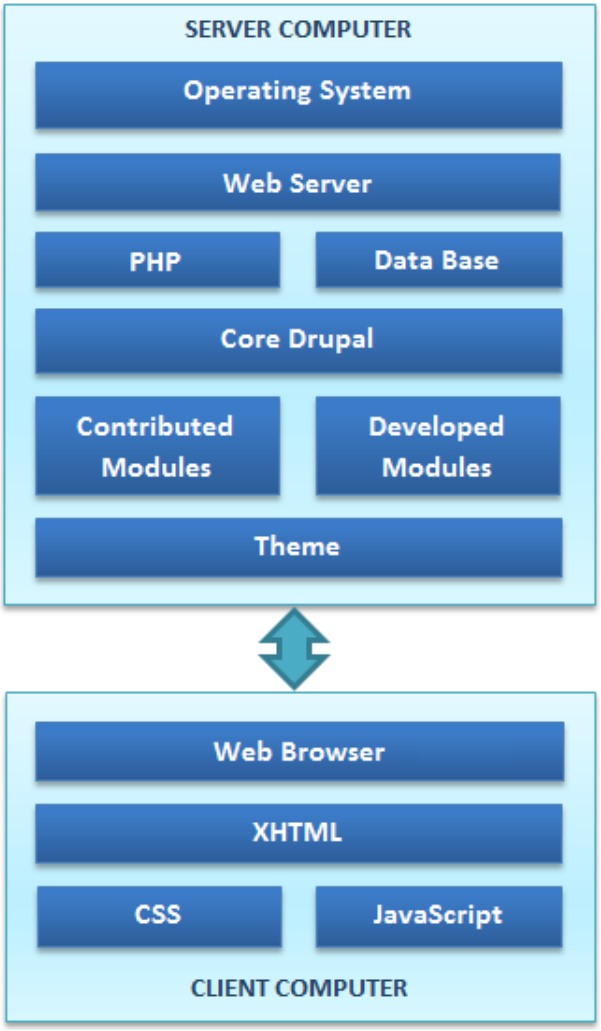


Figure 29.: Drupal stratification of the system architecture.

5.2 WEB PORTAL

This section describes the modules of the web portal, which include modules provided by the Drupal CMS and modules developed in this thesis. It is also presented the Drupal Database Abstraction Layer, and the web portal front-end and back-end.

5.2.1 *Web Portal Modules*

The web portal was developed based on Drupal CMS. Therefore, the web portal is designed to be customized, but this customization is done by overriding the core modules or by adding new modules, not by modifying the core code.

Drupal Core Modules

Drupal provides several core modules that cannot be disabled because their capabilities are required to the standard functioning of Drupal. The core modules are listed below:

- **Block:** It controls the visual building blocks of a page. Blocks are boxes of content rendered into an area, or region, of a web page.
- **Comment:** It allows users to comment on and discuss published content.
- **Content translation:** It allows content to be translated into different languages.
- **Contextual links:** They provide contextual links to perform actions related to the elements on a page.
- **Dashboard:** It provides a dashboard page in the administrative interface for organizing administrative tasks and tracking information within our site.
- **Date and Calendar:** These modules provide the ability to create and manipulate date fields, as well as rendering a list of events on a calendar.
- **Field:** A field API to add fields to entities, such as nodes and users.
- **File:** It defines a file field type.
- **Filter:** It filters content in preparation for display.
- **Image:** It provides image manipulation tools.
- **List:** It defines list field types. When used with Options, it allows the creation of selection lists.
- **Locale:** It adds language handling functionality and enables the translation of the user interface to languages other than English.
- **Menu:** It allows administrators to customize the site navigation menu.
- **Node:** It allows content to be submitted to the site and displayed on pages.
- **Number:** It defines numeric field types.
- **Path:** It allows users to rename URLs.
- **Pathauto:** This module creates search engine-friendly URLs by automatically generating a “pretty" URL that is based on the pages title.
- **System:** It handles general site configuration for administrators.
- **Taxonomy:** It enables the categorization of content.
- **Text:** It defines simple text field types.
- **User:** It manages the user registration and login system.

Drupal Community Contributed Modules

The Drupal Community contributes with many modules where most of them are free software. The contributed modules used in the web portal are listed next:

- **Chaos tools:** A library of helpful tools.
- **Date:** It makes date/time fields available and enables jQuery popup calendars and time entry widgets for selecting dates and times.
- **SMTP Authentication Support:** It allows to send emails through an SMTP server.
- **Media:** It provides the core Media API.
- **Automatic Nodetitles:** It allows hiding of the content title field and automatic title creation.
- **Colorbox:** It provides a customizable lightbox plugin for jQuery.
- **Facebook Comments:** It adds Facebook comments to specific node types and handles comment counts with views.
- **Pathauto:** It provides a mechanism for modules to automatically generate aliases for the content they manage.
- **Twitter:** It adds integration with the Twitter microblogging service.
- **Gallery Formatter:** It provides a jQuery gallery Content Construction Kit (CCK) formatter for image-field.

Developed Modules

The developed modules implement the functionality described in the captured functional requirements, and which is not present in the contributed modules neither in the core modules. The developed modules are listed next:

- **Touristic Resources:** It allows authenticated users to manage touristic resources.
- **Trips:** It allows authenticated users to manage trips.
- **Plan Trip:** It allows authenticated users to organize their trips.

5.2.2 *Database Abstraction Layer*

Drupal contains a database abstraction layer based on the PHP Data Objects (PDO) library, which introduces an object-oriented syntax to queries (Shreves and Dunwoodie, 2011). This layer gives developers one way of dealing with more complex functionality. It also provides a structured interface for dynamically constructing queries, and enforcing security checks and similar good practices.

The Drupal architectural pattern is called `Node`. This pattern makes the mapping between the classes and the relational database, so each instance of the class is linked to a row in the table. The `Node` pattern has methods for data accessing. An example of using the pattern in the web portal is presented in figure 30.

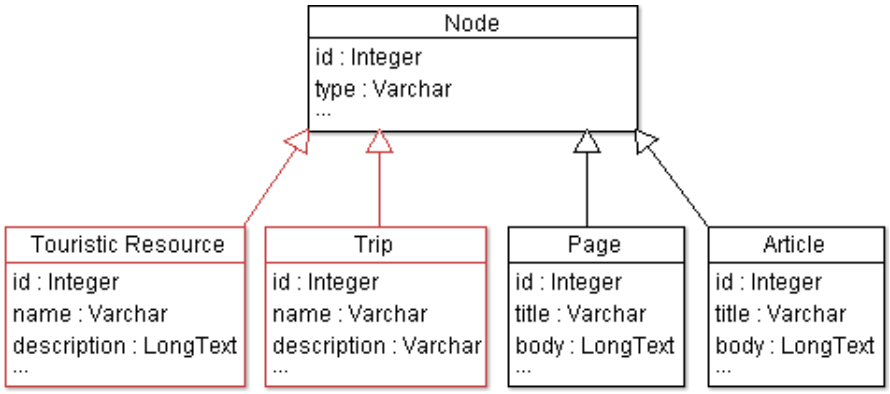


Figure 30.: An example of the Drupal architectural pattern `Node`.

In the example presented in figure 30 it is possible to see that the class `Node` is responsible for data access, and it has five main methods: `db_query()`, `db_select()`, `db_insert()`, `db_update()`, and `db_delete()`. We can also note that there are other business objects that derive from `Node`: `Touristic Resource`, `Trip`, `Page`, and `Article`.

The operations supported by the `Node` are:

- CRUD operations (Create, Read, Update, and Delete data);
- Find methods to make queries;
- Validation methods.

The Drupal database does not store just application data, but also stores configuration, caches, metadata, structural information, and even PHP code (Butcher et al., 2010).

5.2.3 *Web Portal Front-End*

The web portal is where tourists can view the tourist information they seek. In terms of presentation, it is necessary to have a simple and friendly interface for most people. The website contains many features to satisfy the users expectations, as can be seen in this section.

Home Page

Figure 31 shows the homepage of the web portal, which displays links to the most popular touristic resources and a big button for planning a new trip. There are also elements for navigation, such as the main menus, search input, a trip list, and user information. This page contains a header that will be used in other pages to bring the user back to the homepage of the web portal. Additionally, the homepage can be viewed in three languages: English, Portuguese, and Spanish.

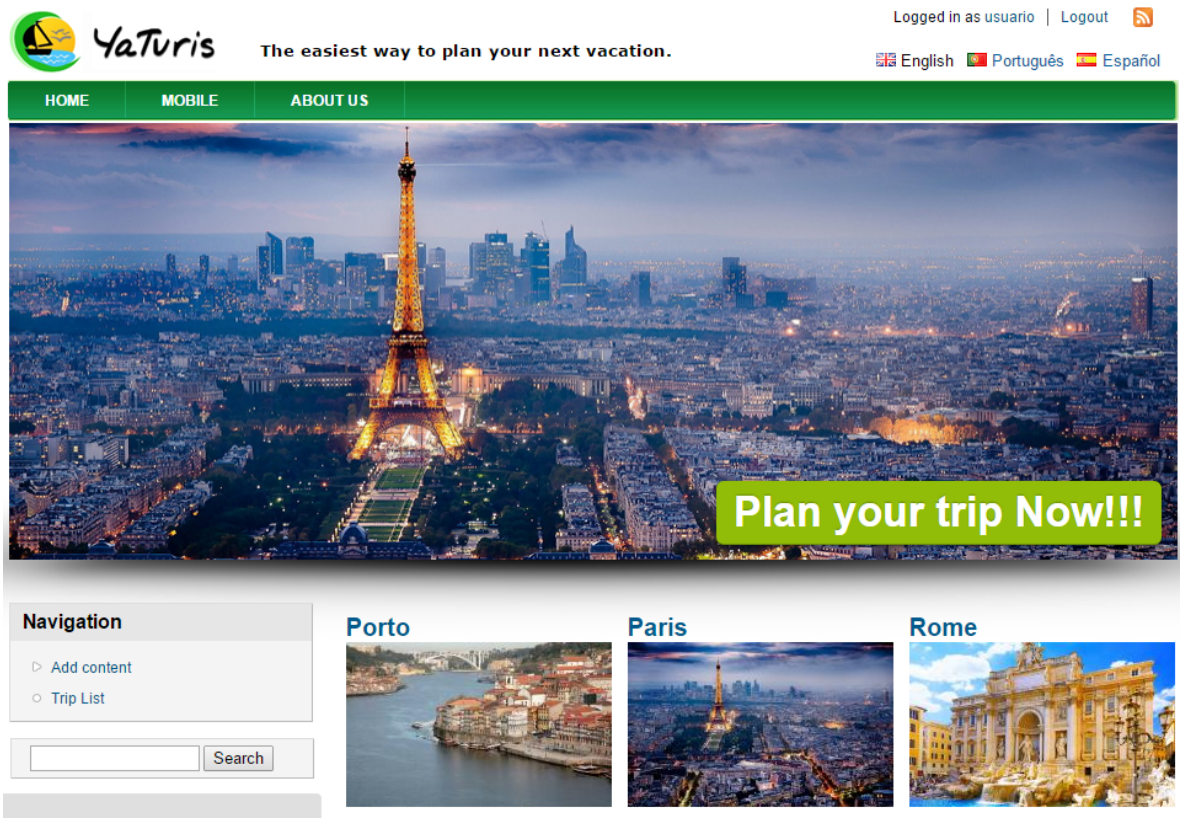


Figure 31.: Web portal home page.

Create a New Trip

Figure 32 shows an overlay page where the users can create a new trip. The overlay has several inputs, such as the name of the trip, the list of places, the start-date, the end-date, and the type of privacy. The list of places is read from the database using a tokenizing autocomplete text entry, which allows users to select multiple places. The selection of dates is accomplished by a datepicker popup. The PHP source code used to create a trip is included in appendix C.1.

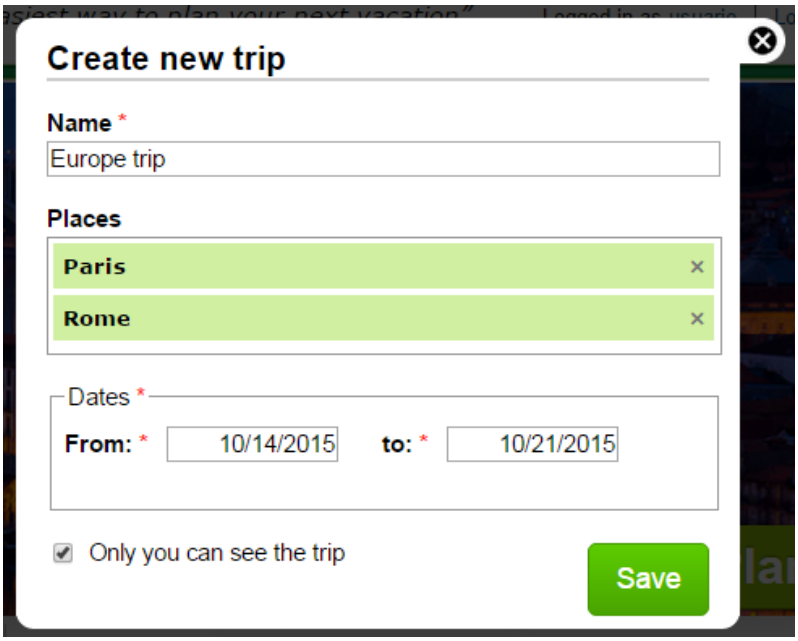


Figure 32.: User interface to create new trip.

Planning a Trip

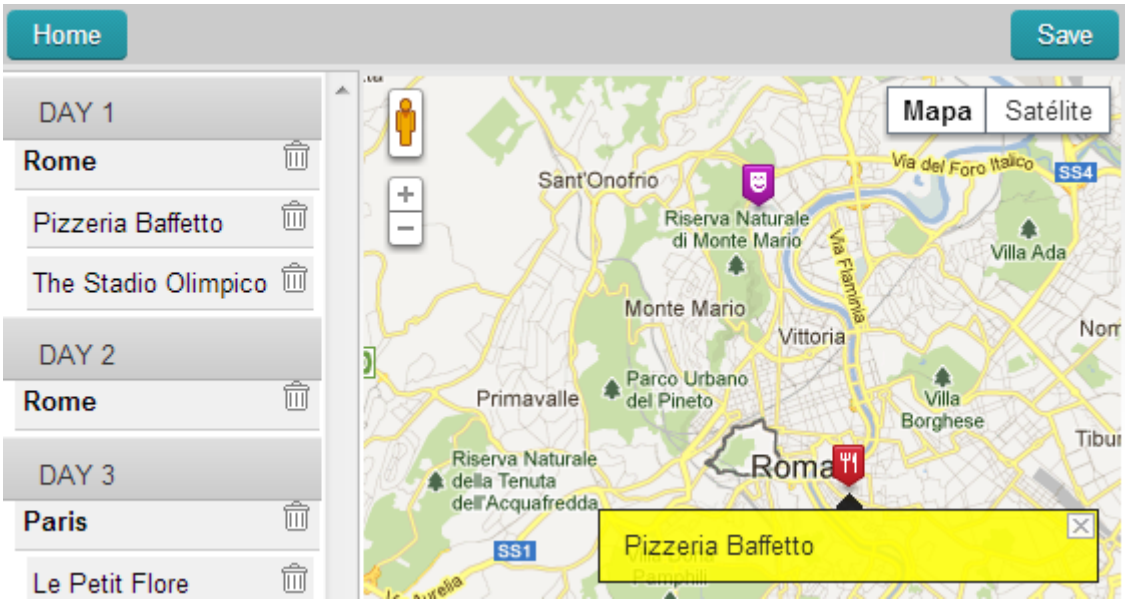


Figure 33.: User interface to plan a trip.

Figure 33 shows the page where the users can plan a trip. In the left sidebar are located the cities with their touristic resources, and the main page content is a map with marks pointing to the touristic

resources for the selected day. Finally, the header has two buttons: `Home` that returns the user to the homepage and `Save` that saves the whole trip.

View a Touristic Resource

Figure 34 shows a touristic resource, in this case the city of Paris, via an image gallery, the textual description, and comments where is possible to expand the image size using a zoom tool. It is also possible to share the touristic place on the main social networks such as Facebook and Twitter and to explore the map.

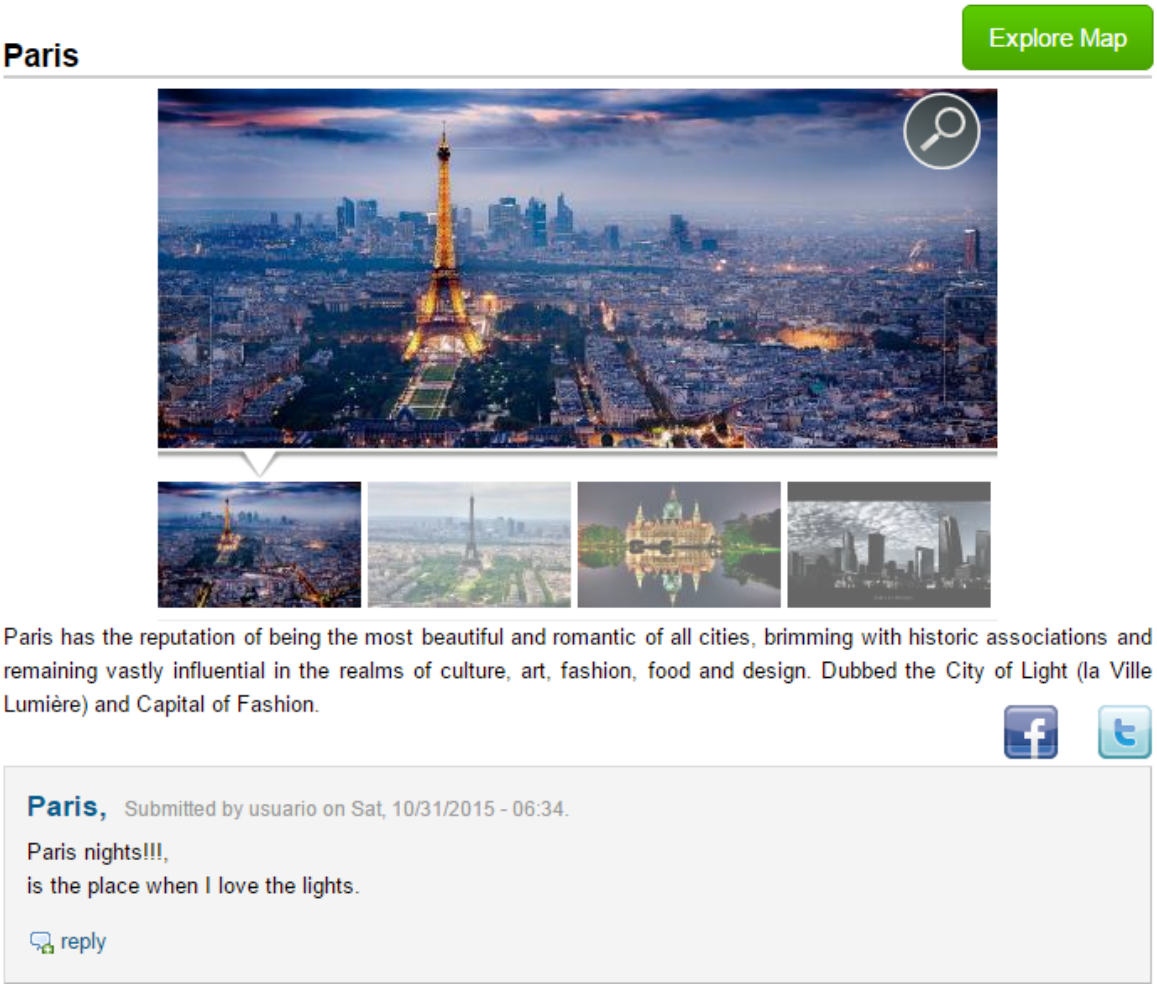


Figure 34.: Touristic resource view.

List of Trips

Figure 35 shows the authenticated user's list of trips. For each trip it is displayed the name of the trip, the start-date, the end-date, and places to visit. The places of the trip are connected by a blue line. The PHP source code used to print the list of trips is included in appendix C.4.

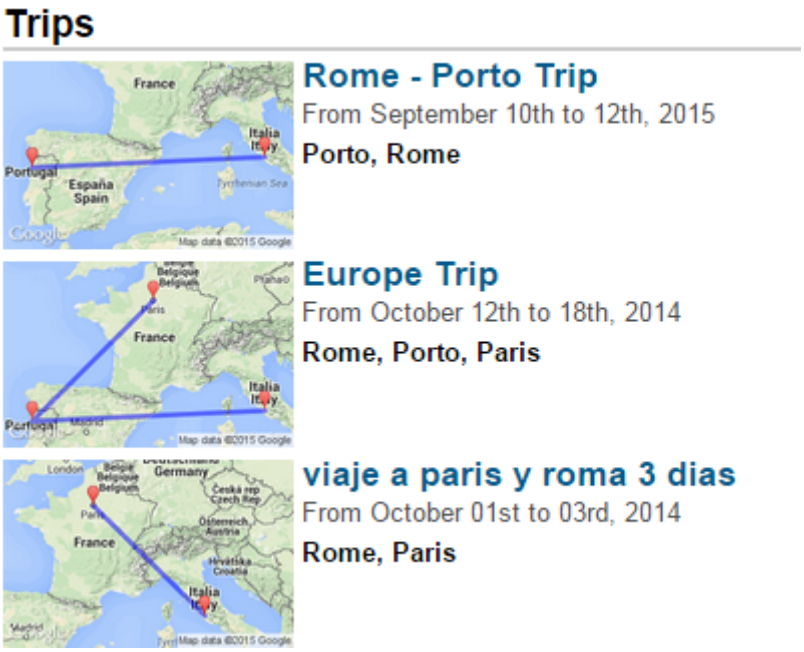


Figure 35.: User list of trips.

Trip Details

Figure 36 shows the interface where the user can view and edit the general information about the trip. It is also possible to add comments, to share the trip, and to rearrange the trip.

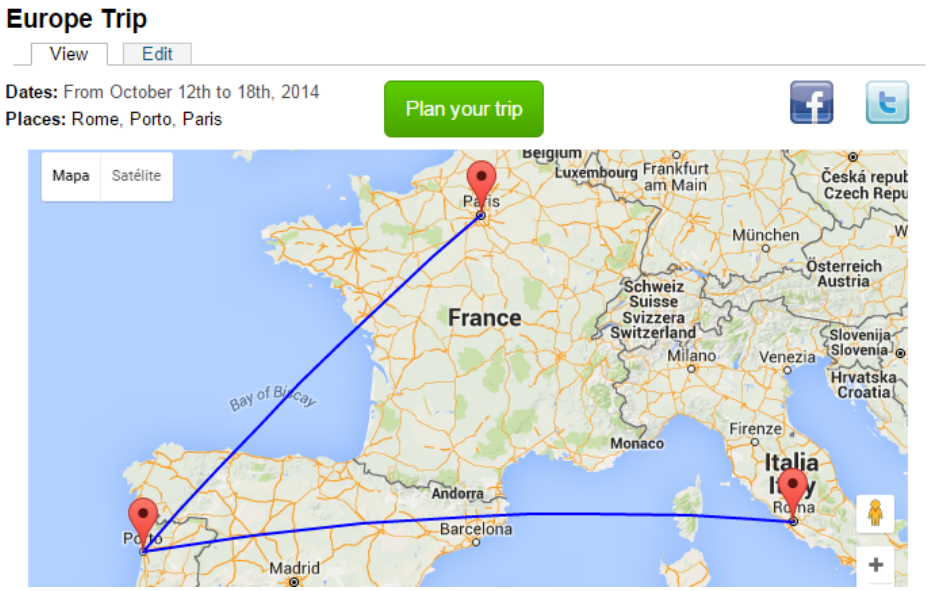


Figure 36.: Trip details.

Create a New Account

Figure 37 shows an overlay page where the user can create a new account. The required inputs are the user name, a valid e-mail address, and a password. Below each input field it is displayed its description. Security recommendations about passwords are shown in order to enter a strong password. The password must be confirmed.

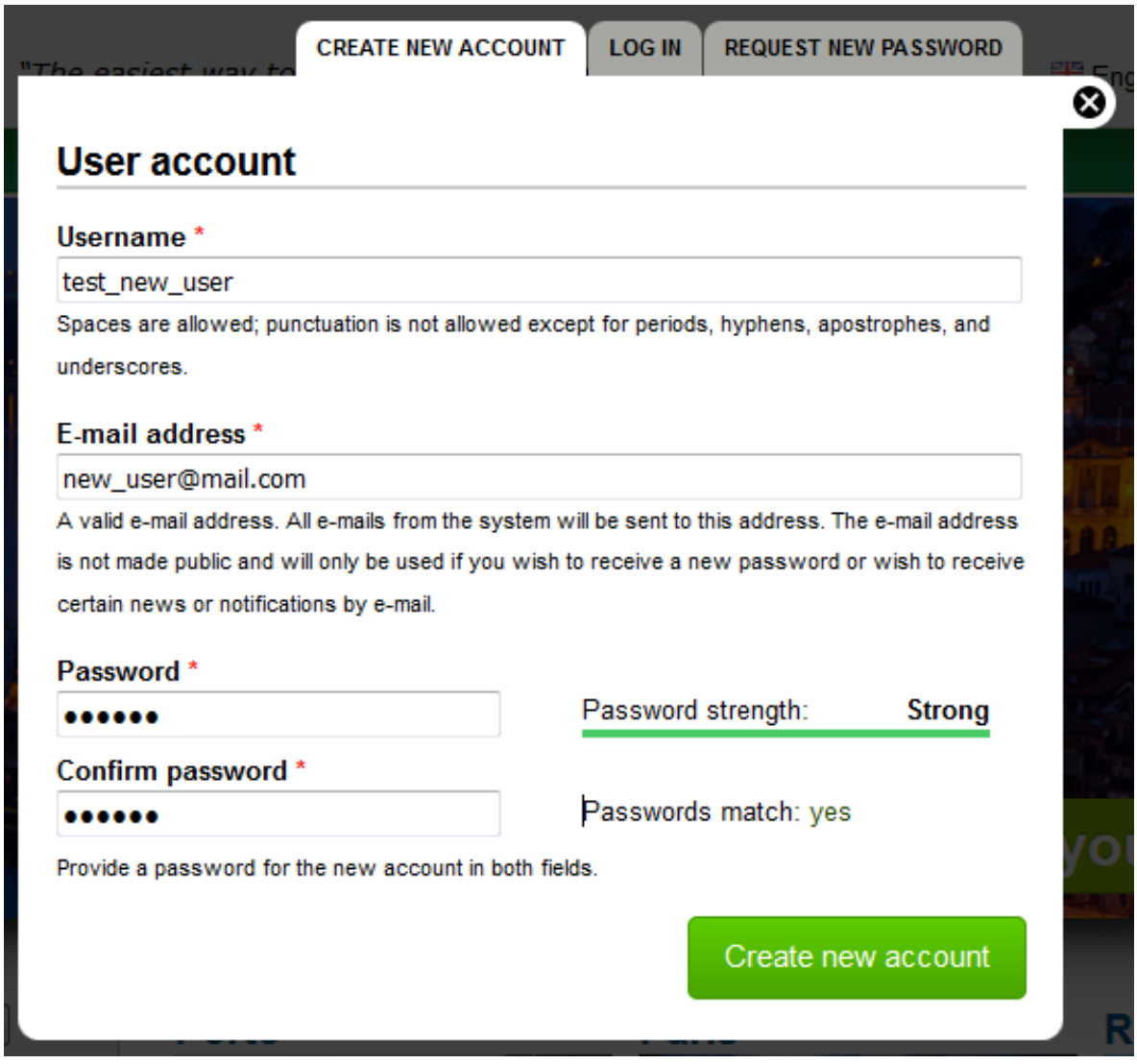


Figure 37.: Create a new account.

Request a New Password

In figure 38 it is presented a small overlay page used by the user to request a new password. To successfully complete this request it is necessary to enter the user name or the e-mail address.

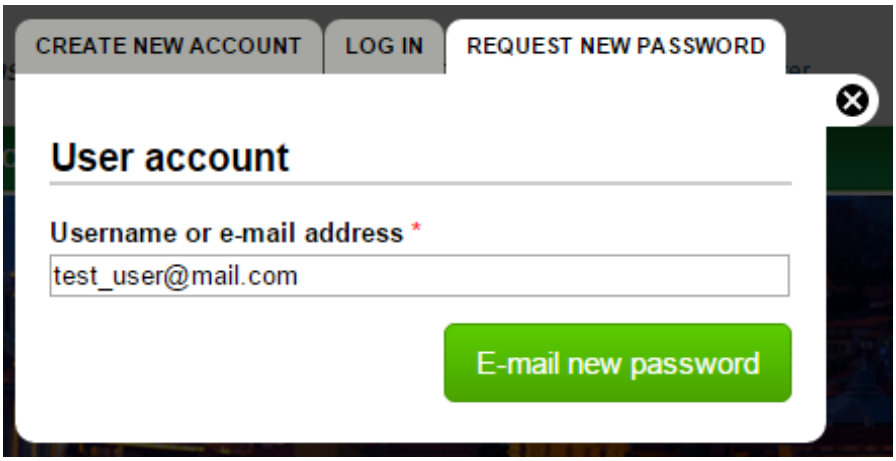


Figure 38.: Request a new password.

Log in

Figure 39 shows the `log in` overlay page where the user introduces his login and password. Also, users can login with their social network accounts such as Facebook or Twitter.

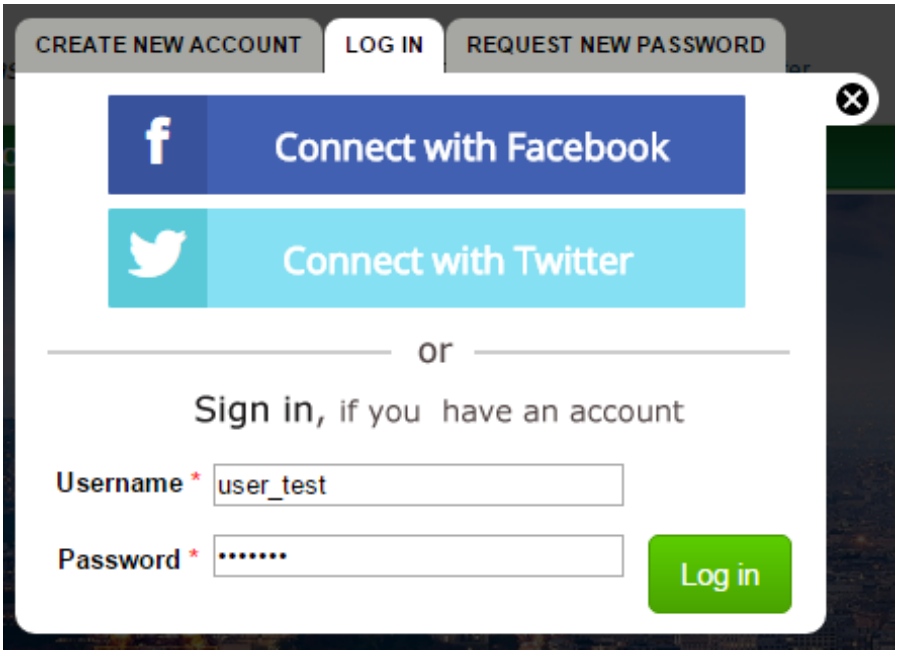


Figure 39.: Log in to the web portal.

5.2.4 *Web Portal Back-End*

This section describes the web portal back-end, which includes the administrator pages used to configure and manage the system content. This configuration allows the administrator to enable/disable contents or to change the content order, among other functionalities.

System Modules

Figure 40 shows the administration interface, where the administrator can activate, disable or configure the system modules. The system modules are grouped in nine sections:

- The `Core` section allows managing users, blocks, contents, comments, content translation, fields, images, menus, files, and statics.
- The `Chaos Tool Suite` provides a set of APIs and tools to improve the module development.
- The `Date/Time` section allows managing date, time, and popup calendars.
- The `Mail` section allows the web portal to send e-mails through an SMTP server.
- The `Media` section provides a framework for managing files and multimedia assets.
- The `OAuth` section allows managing social networks.
- The `Other` section allows managing tokens, automatic node titles, and image lightboxes.
- The `User Interface` provides a gallery formatter for images.
- The `Touristic Social Communities` allows managing trips and touristic resources.

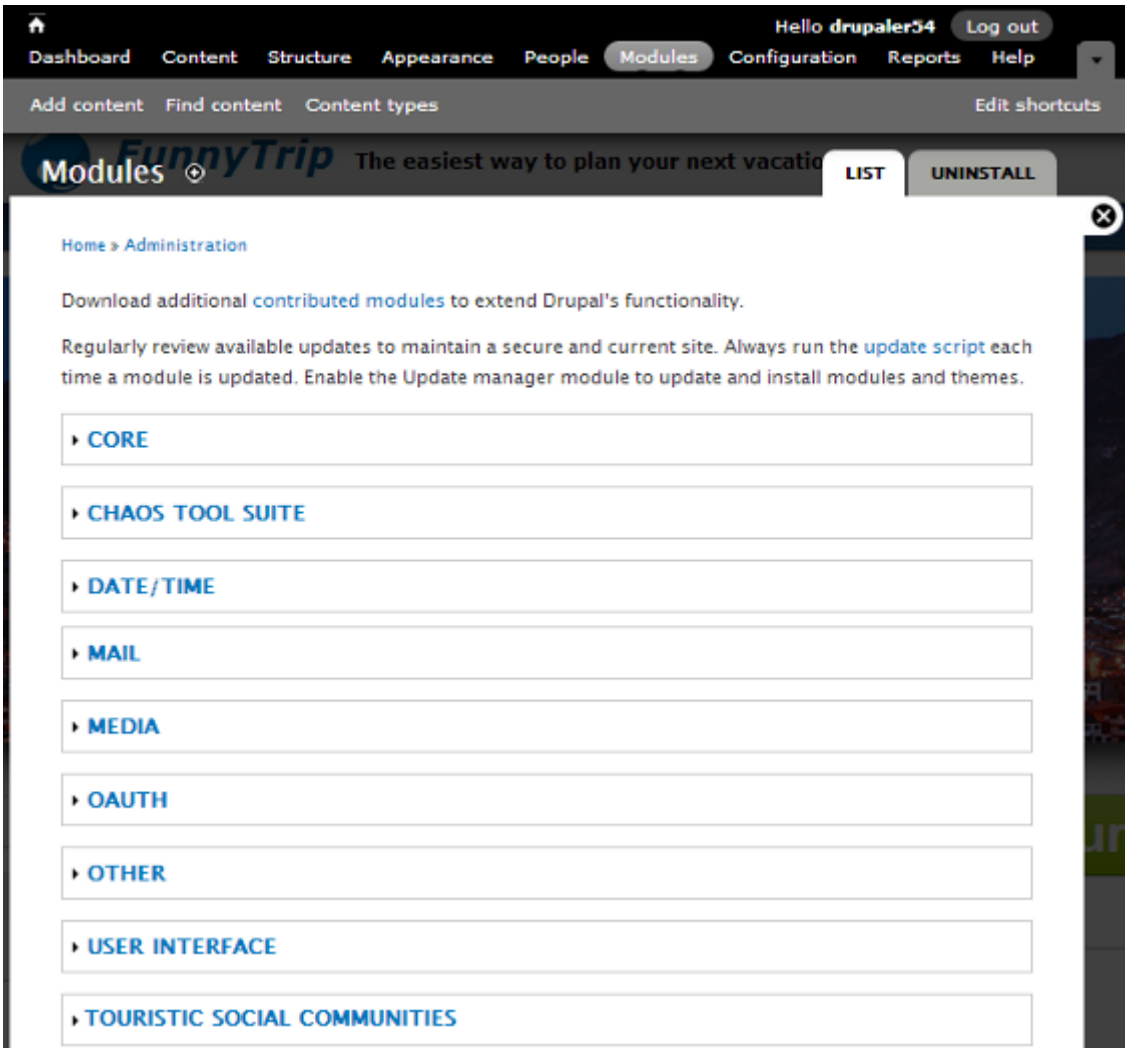


Figure 40.: Administration page.

Add or Edit a Touristic Resource

Figure 41 presents the form used to edit a touristic resource. Here the administrator can introduce the data that characterizes the resource, namely the name, country, city, address, zip code, location in the map, category, e-mail, web site, telephones, brief description, details, images, and videos. The PHP source code used to create and update a touristic resource is included in appendices C.2 and C.3, respectively.

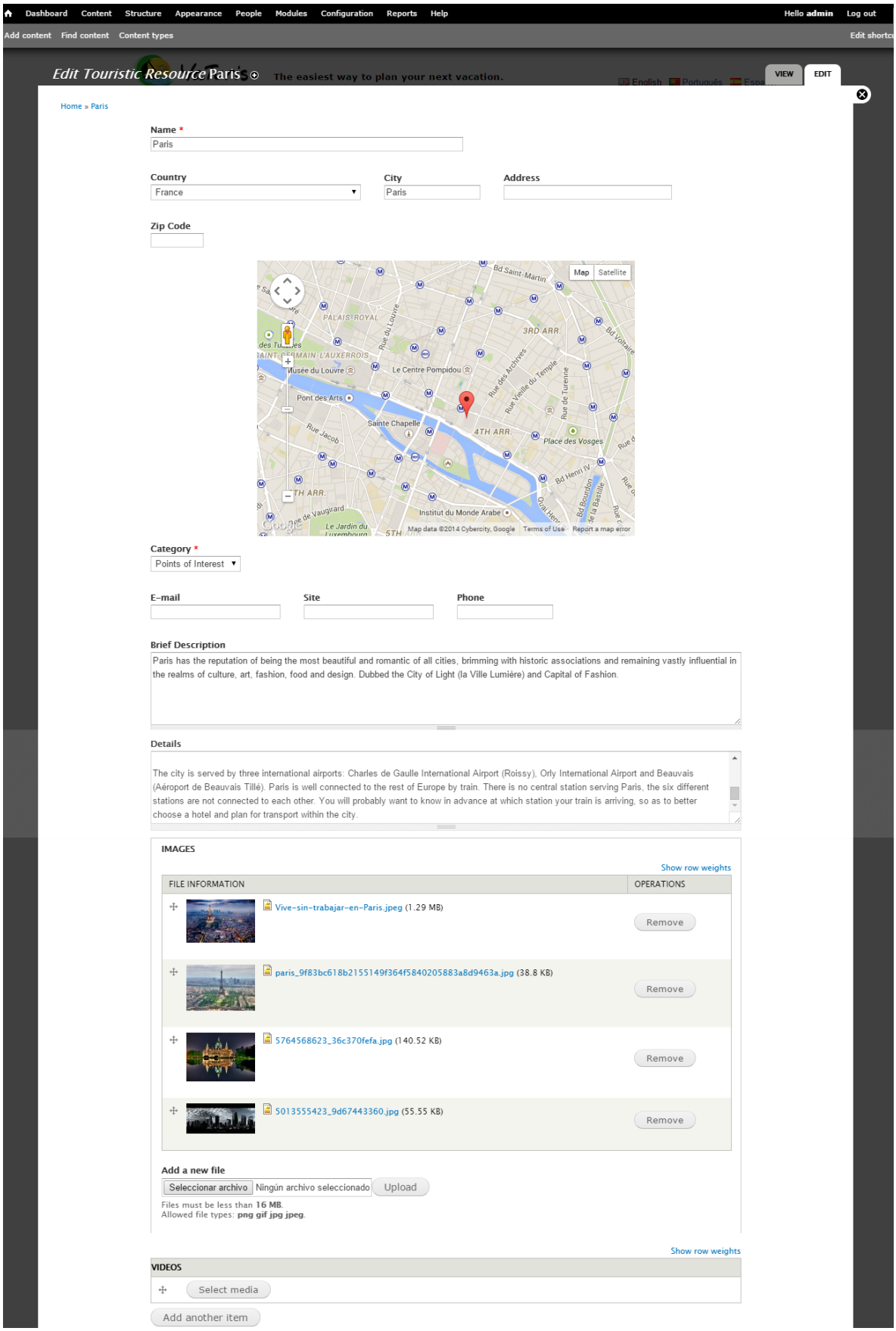


Figure 41.: Form for editing a touristic resource.

5.3 MOBILE APPLICATION

This section describes the implementation of the secure REST Application Programming Interface (API), which is responsible for the interaction between the mobile application and the web portal. It is also presented the user interface of the mobile application, and the advantages and disadvantages of a mobile native application in relation to accessing the web portal via web browser.

5.3.1 *Application Programming Interface*

In order to implement CRUD operations for the mobile native application to interact with the Web Server PostgreSQL database, it was necessary to build a REST API (figure 42). REpresentational State Transfer (REST) is useful to build client/server network applications using the HTTP protocol (Fielding, 2000). The REST API allows us to get the request from the server, interact with the database, and finally to give the response back to the client. The REST API accepts requests in GET, POST, PUT, and DELETE methods and gives a response in JSON format.

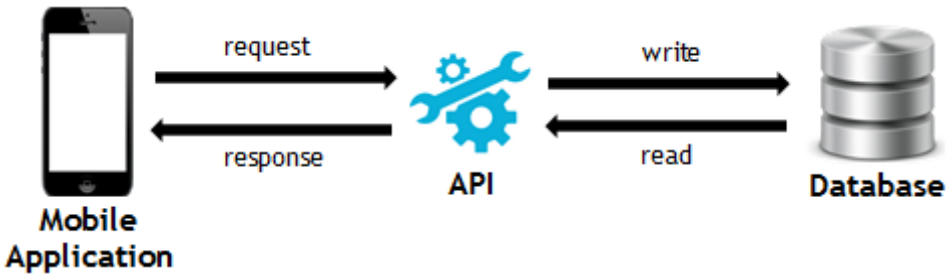


Figure 42.: Application Programming Interface.

HTTP Methods

The designed REST API supports the GET, POST, PUT and DELETE HTTP methods. The HTTP method to be used depends on the type of operation we want to execute, as explained below:

- **GET:** To fetch a resource.
- **POST:** To create a new resource.
- **PUT:** To update an existing resource.
- **DELETE:** To delete a resource.

HTTP Status Code

A REST API request produces a JSON response containing a HTTP status code. On the client side it is verified the response HTTP status code. If the status is 200, the request is processed successfully (W3C, 1999). Its is also possible to receive an `error` code in the response. If the error value is `true`, it means that some error occurred while processing the user data. The following HTTP status codes are supported by the mobile application:

- **200** – OK (Standard response for successful HTTP requests)
- **201** – Created
- **304** – Not Modified
- **400** – Bad Request
- **401** – Unauthorized
- **403** – Forbidden
- **404** – Not Found
- **500** – Internal Server Error

API Key

In order to avoid sniffing vulnerability, the access to the API is restricted. Therefore, it was designed a secure REST (Web) API using a random **API key** for every user, where the user is identified by a unique API key and all the actions can be performed only on the resources belonging to him.

The API key never travels on the HTTP request, only the server and the client know the API key. A client creates a unique **hash** representing its request to the server. The hash is obtained by scrambling the request data with the API key. Then, the client sends that hash to the server, along with all the arguments and values. Next, the server gets the request and re-generates its own unique hash based on the submitted values and using the same methods the client used. Finally, the server compares the two hashes. If they are equal, the server trusts the client and executes the request.

API Calls

Table 6 contains a list of API calls. An API call redirects from a URL to a function. For example, if the API call is `http://api.yaturis.com/trip/12` with the HTTP method `GET`, the trip with the `id` equal to 12 will be read by running the function `read_trip()`.

The base URL of any API call is `http://api.yaturis.com`. On the other side, it is possible to notice that the same URL endpoint is used for multiple API calls, but the difference is in the type of HTTP method. For example, if the API call includes the URL `/trip/:id` and the GET method,

the trip identified by `id` will be read. Moreover, if the API call includes the URL `/trip/:id` and the PUT method, the trip identified by `id` will be updated.

URL	Method	Parameters	Description
/register	POST	name, email, password	User registration
/login	POST	email, password	User login
/trip	POST	name, places, start date, end date, privacy	To create new trip
/trips	GET	user	Fetching all trips
/trip/:id	GET		Fetching single trip
/trip/:id	PUT		Updating single trip
/trip/:id	DELETE		Deleting single trip
/place/:id	GET		Fetching single touristic resource
/place/:id	DELETE	trip	Removing a single touristic resource from the trip

Table 6.: API calls list.

5.3.2 *Mobile Application vs. Access the Web Portal via Browser*

The advantages of a mobile native application in relation to accessing the web portal via web browser are:

- A mobile application can run with higher performance because it uses native code. Therefore, it is not necessary to render all the content of its view.
- A mobile application can access the hardware features of the device, such as GPS, camera, accelerometers, and touchscreens, in order to deliver a better presentation and rich user experience.
- A mobile applications can run in an off-line mode, where the internet connection is not required. It is also possible to use off-line data storage.
- A mobile application can be launched in a single step.
- Mobile applications get full support from the concerned app stores and marketplaces. Users can easily find and download mobile applications from these stores.
- The user can be assured of safety and security of the mobile applications because they have to get the approval of the app store they are intended for.

The disadvantages of a mobile native application in relation to accessing the web portal via web browser are (Lionbridge, 2012):

- A mobile application requires a local installation and it does not run through a web browser.

- A mobile application has significant development cost and time because each platform has its own framework and programming language.
- The process of getting a mobile application approved in the app store can be long and tedious for the developer and not always results in success. There is also no guarantee that the mobile application becomes instantly popular with users.
- Users of different mobile devices may be using different versions of the mobile application, which makes it difficult for the developer to maintain and offer support.
- A mobile application has frequent upgrades in the platforms where it runs. This results in significant overheads for the application developers to test and address compatibility issues.

5.3.3 *Front-End*

This section presents the user interface of the developed mobile application. Figure 43 shows the login screen where the user introduces his account information: login/e-mail and password.

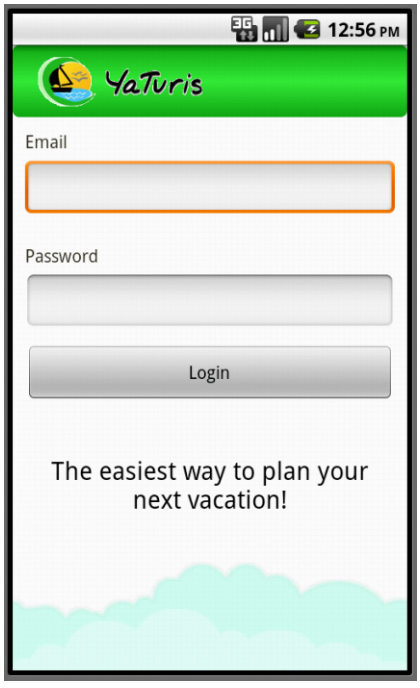


Figure 43.: Login screen.

In figure 44 it is displayed the user's list of planned trips. For each trip, it is presented a short description, the dates, and the places to visit. It is also possible to select a trip. The Java source code used to add a trip to a list is included in appendix C.5

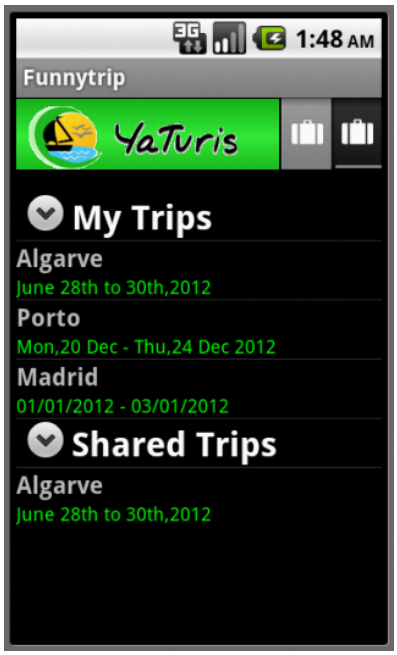


Figure 44.: List of trips.

Figure 45 gives the details of a trip. In the screen, the user can view the list of places of interest grouped by day. For each place, it is displayed the name and type. The XML layout used to list the touristic resources is included in appendix C.6

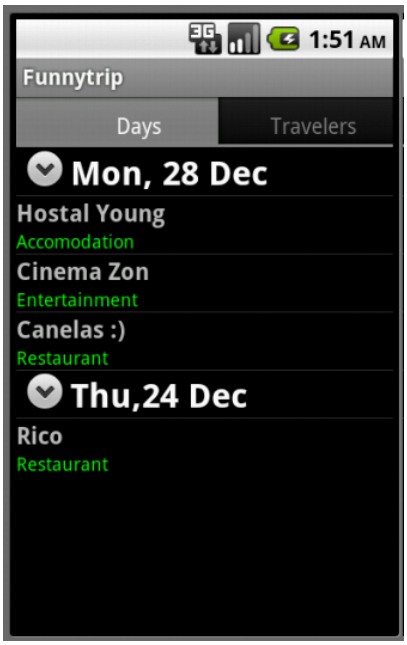


Figure 45.: Details of a trip.

Figure 46 shows a specific touristic resource, the famous city of Porto. It is presented basic information, such as the name of the city, the country, a small description, and a picture. It is also displayed a link to the map of the city. Finally, the interface shows six icons that represent restaurants, hotels, attractions, health, shopping, and other resources.

Figure 46.: Touristic place dashboard.

In figure 47 the touristic resources of the trip are displayed on a map, where is possible to see the location of the user and the closer points of interest.

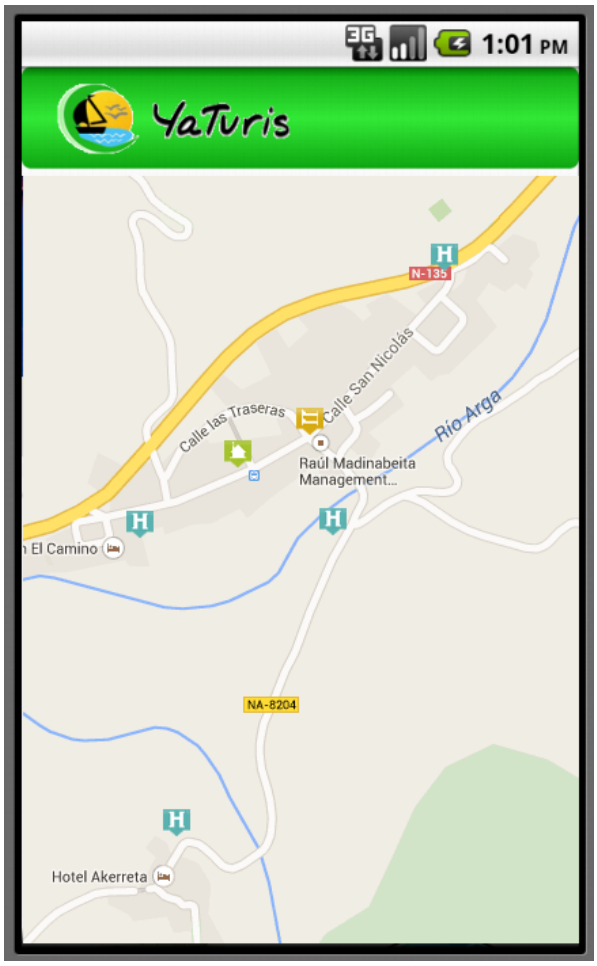


Figure 47.: A trip shown on a map.

6

CONCLUSIONS AND FUTURE WORK

Summary

This chapter summarizes the main conclusions and contributions of the dissertation. To conclude, some notes are given on how this work can be further developed.

6.1 CONCLUSIONS

This work presents the development process of a web portal for touristic social communities, synchronized with a mobile native application, where the users can organize their trips and to define what touristic resources they would like to visit and to keep track of the performed activities. It was developed a website combining the best functionalities from the four sites analysed. We could develop a web portal as TripIt where it is possible to register details for a touristic resource, such as the number of a flight ticket or the number of a table in a restaurant. We could also categorize the users' evaluation of touristic resources. The author believes that we must do a trade-off between usability and complexity. For example, when a site shows a lot of touristic information, as in Yahoo Trip Planner, this possibly causes fatigue or disinterest to the user. Therefore, our goal was to develop a web portal with the essential features present in all analysed web portals, but having strengths as simplicity and navigability of TouristEye, and the detailed classification of touristic resources from TripAdvisor.

In the process of requirements elicitation, it were identified the essential features of the web portal for touristic social communities using techniques such as personas, conversations, and analysis of similar web portals. The design phase only included the enough modeling, namely use cases, domain, classes, and data models. After modeling and reviewing the features offered by Drupal CMS, it was found that half of the system requirements are implemented by Drupal itself. Among the requirements that Drupal implements are the management of users, content, comments, and languages. The modules for managing trips and touristic resources were developed in this thesis but using the Drupal

framework too. In the implementation phase of the web portal, it was noticed that the development with Drupal uses the PAC design pattern, a hierarchical pattern that permits a faster user experience.

It was used the Google Maps mapping service application, since it provides the largest number of touristic places located on its maps and the Google Places API. This API is used for searching and retrieving information about local businesses or points of interest. We also emphasized the importance of the main social networks, such as Facebook or Twitter, for the diffusion of the web portal.

In order to interact and synchronize the mobile native application with the web portal, it was developed a REST API that allows the trips and the touristic resources to be available on both platforms. The API REST was developed using hash functions to restrict the access to the trips and/or the user account.

The mobile native application was developed for the Android operating system due to its popularity and its open source framework. The mobile application was focused on making available hotels, restaurants, health, attractions, and purchases to its user. The mobile native application displays the touristic resources that are closer to the user. When there are problems with internet connection, the mobile application relies on GPS and runs in off-line mode.

The web portal and the mobile native application are fully functional and available online at url `http://www.yaturis.com`, and were designed in agreement with software engineering best practices.

6.2 FUTURE WORK

The portal documented in this thesis runs on a single server. So, it will be very significant to reproduce the same web portal on a large scale with hundreds or thousands of instances. Therefore, it will be possible to check whether it maintains the same behaviour. On the other side, it would be interesting to improve the web portal and the developed mobile application in order to build a large virtual touristic community by developing promotion modules.

It will be also interesting to release an official version for Google Market and to develop mobile applications for the most popular mobile system, besides Android, such as iOS, Blackberry, and Windows Phone. Therefore, it will be interesting to analyse the development costs for every kind of mobile system. Furthermore, in order to reduce the development costs it is possible to analyse the deployment of web applications as native applications by wrapping the web application code using a native framework such as PhoneGap.

Additionally, it will be helpful to send to the users relevant messages about the weather or important events at the touristic resources using SMSs or e-mails.

Another possibility of evolution, is to re-structure the web portal using the responsive web design in order to adapt it to tablets or any device of any screen size through fluid grids, flexible images, and media queries. Another possibility is to log in, both in the web portal and the mobile application, through the Google+ social network.

Another interesting idea for future work, will be to associate our portal to hotels web portals and to use web scrapping for collecting open touristic resources from other touristic web portals.

BIBLIOGRAPHY

Josef Altmann and Nitin Arora. *Web Engineering*. John Wiley and Sons, 111 River Street, Hoboken, NJ 07030, USA, first edition, 2006. ISBN 0-470-01554-3.

E. Bakshy, I. Rosenn, C. Marlow, , and L. Adamic. The role of social networks in information diffusion. *ACM*, page 519–528, 2012.

Cruz Zapata Belén and Hernández Ni nirola. *Testing and Securing Android Studio Applications*. Packt Publishing, 2014. ISBN 978-1-78398-880-8.

Frank Buschmann. *Pattern-Oriented Software Architecture*. Wiley, first edition, 1996.

Matt Butcher, Greg Dunlap, and Matt Farina. *Drupal 7 Module Development*, page 15. PACKT, Birmingham, 2010.

Sven Casteleyn, Florian Daniel, Peter Dolog, and Maristella Matera. *Engineering Web Applications*. Springer, 2009. ISBN 978-3-540-92200-1.

M. De Choudhury, Y.R. Lin, H. Sundaram, K.S. Candan, L. Xie, and A. Kelliher. How does the data sampling strategy impact the discovery of information diffusion in social media. *In Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*, page 34–41, 2010.

Alexander Christopher. *A Pattern Language: Towns, Buildings, Construction*. Oxford, Oxford, UK, first edition, 1977. ISBN 0195019199.

Alan Cooper and Robert Reimann. *About Face 2.0: The Essentials of Interaction Design*, pages 56–62. Wiley, Indianapolis, 2003.

Rui A. Costa. *Manual do SGI(Sistema de Gestão Integrado)*. Ubiwhere Company, Rua Cristovão Pinho Queimado, 47, 3800-012, Aveiro, 1.5 edition, 2011.

Riccardo Consoli Daniela Baglieri. *Learning from Customers: Running Virtual Communities in Tourism*, pages 430–433. Proceedings of the 11th Toulon-Verona International Conference on Quality in Services, Toulon-Verona, 2008.

Laloumis Dimitris and Katsoni Vasiliki. The effect of virtual communities and social network sites on tourism destination choice. *Journal of Tourism Research*, page 7, 2013. URL http://www.jotr.eu/index.php/tourism-management/66-effect.

André Luís dos Santos Domingues, Sandro Lopes Bianchini, Reginaldo Ré, and Fabiano Cutigi Ferrari. A comparison study of web development methods. *Universidade de São Paulo*, page 17, 2008. URL http://www2.dc.ufscar.br/~fabiano/pub_arquivos/Domingues_CLEI2008_CameraReady.pdf.

Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.

Martin Fowler. The new methodology. *Fowler*, 2001. URL http://www.martinfowler.com/articles/newMethodology.html.

PriceDekho The Indian Express Group. Google androidx lords over 85 pct of smartphone os market share, appleś ios distant second: Idc. *Online*, 2014. URL http://www.financialexpress.com/news/google-android-lords-over-85-pct-of-smartphone-os-market/1284435.

Rheingold H. *The Virtual Community: Homesteading on the Electronic Frontier*, pages 3–20. Addison-Wesley Publishing Company, New York, 1993.

Mike Holcombe. *Running an Agile Software Development Project*, pages 19–36. Wiley, New Jersey, 2008.

James and Suzanne Robertson. Volere requirements specification template. *Online*, 2006. URL http://www.volere.co.uk/template.htm.

Cory Janssen. Mobile application (mobile app). *Online*, 2014. URL http://www.techopedia.com/definition/2953/mobile-application-mobile-app.

Lionbridge. Mobile web apps vs. mobile native apps: How to make the right choice. *white papers*, page 13, 2012. URL http://www.lionbridge.com/files/2012/11/Lionbridge-WP_MobileApps2.pdf.

Casaléa L.V., Flavián b C., and Guinalíu M. Relationship quality, community promotion and brand loyalty in virtual communities: Evidence from free software communities. *International Journal of Information Management*, pages 357–367, 2010.

Reto Meier. *Professional Android 2 Application Development*, pages 2–7. Wiley, Indianapolis, 2010.

Michael P. Peterson. *Mapping in the Cloud*, page 261. Guilford, New York, 2014.

Antonio Guevara Plaza. Web 2.0 and its application to the tourism sector. travel 2.0. *bit*, AGO.-SEP. (170):38, 2008.

Paul Ramsey. *PostGIS Manual*, 1.3.2 edition, 2005.

Doug Rosenberg and Matt Stephens. *Use Case Driven Object Modeling with UML*. Apress, New York, 2007. ISBN 978-1-59059-774-3.

Kenneth S. Rubin. *Essential Scrum: A Practical Guide to the Most Popular Agile Process*, pages 13–25. Addison-Wesley, New Jersey, 2013.

Chris Ryan. *Recreational tourism: a social science perspective*. Routledge, New York, 1991. ISBN 0415054249, 9780415054249.

Ric Shreves and Brice Dunwoodie. *Drupal 7 Bible*, page 547. Wiley, Indianapolis, 2011.

Internet World Stats. Internet world stats. *Online*, 2014. URL http://www.internetworldstats.com/stats.htm.

S. Sumathi and S. Esakkirajan. *Fundamentals of Relational Database Management Systems*, page 3. Springer, Warsaw, 2007.

Todd Tomlinson. *Beginning Drupal 7*. Apress, 2010. ISBN 978-1-4302-2859-2.

W3C. Http status code definitions. *Online*, 1999. URL http://www.w3.org/Protocols/HTTP/HTRESP.html.

Karzan Wakil and Dayang N.A. Jawawi. Metamodels evaluation of web engineering methodologies to develop web applications. *International Journal of Software Engineering and Applications (IJSEA)*, page 17, 2014. URL http://airccse.org/journal/ijsea/papers/5514ijsea04.pdf.

Karzan Wakil, Dayang N.A. Jawawi, and Amirhossein Safi. A comparison of navigation model between uwe and webml: Homepage development case study. *International Journal of Information and Education Technology*, pages 5–6, 2015. URL http://www.ijiet.org/papers/585-D102.pdf.

Wikipedia. Mobile operating system. *Online*, 2014a. URL http://en.wikipedia.org/wiki/Mobile_operating_system.

Wikipedia. Turismo em portugal. *Online*, 2014b. URL https://pt.wikipedia.org/wiki/Turismo_em_Portugal.

Ralph R. Young. *The Requirements Engineering Handbook*, pages 51,52. Artech House, Norwood, 2004.

A

REQUIREMENTS SPECIFICATION WITH VOLERE TEMPLATE

A.1 FUNCTIONAL REQUIREMENTS - WEB PORTAL

Requirement #: 1	**Requirement Type:** 1	**UC:** CU02_01, CU02_02, CU02_03, CU02_04, CU02_07, CU02_08, CU02_09, CU02_10, CU02_11, CU02_12, CU02_13, CU02_14, CU02_15, CU02_16, CU02_17, CU02_18, CU02_20, CU02_21, CU02_22, CU02_23
Description: Manage trips, travels, tours, places, and point of interest.		
Rationale: Give administrators the ability to add, modify, view, and delete trips, routes, landmarks, events, travels, tours, places, and point of interest.		
Originator: Statement of work		
Fit Criterion: The Back Office shall be able to manage trips, travels, tours, places and point of interest. The management includes the visualization, creation, modification, and elimination of information. The Front Office shall allow for registered users the management of their own trips.		
Customer Satisfaction: 5	**Customer Dissatisfaction:** 5	
Priority: High		

Requirement #: 2	**Requirement Type:** 1	**UC:** CU02_01, CU02_02, CU02_20, CU02_21, CU02_22
Description: Display the details of the trip and navigate on the trip.		
Rationale: Give users the detailed information about the trip (points of interest, services, places, names, description, distances, etc.) to ponder their choices.		
Originator: Flyers tourist information and statement of work		
Fit Criterion: The user shall be able to display details of the trip.		
Customer Satisfaction: 5	**Customer Dissatisfaction:** 4	
Priority: High		

Requirement #: 3	**Requirement Type:** 1	**UC:** CU02_02, CU02_10
Description: Display on the map the trip, point of interest, attractions, tours or events.		
Rationale: It is important for the tourist, who doesn't know the itinerary, to know the location of their points of interest, for which is possible to give the tourist the itinerary of his trip, points of reference, events, or services geo-located on the map (example: Google Maps).		
Originator: Similar web portals		
Fit Criterion: Users shall be able to display details of their trips.		
Customer Satisfaction: 5	**Customer Dissatisfaction:** 4	
Priority: High		

Requirement #: 4	**Requirement Type:** 1	**UC:** CU04_01, CU04_02, CU04_03, CU04_04, CU04_05, CU04_06, CU04_07
Description: Search trips and places of interest.		
Rationale: Provides users the possibility to search information easily and intuitively about trips, itineraries, point of reference, visits of tourist interest, events of interest and schedules. Possibility to search using filters, for example, to filter trips by countries, etc.		
Originator: Similar web portals		
Fit Criterion: The web portal will have an area destined to the searches, that it allows to the user to search for word-key or to filter information for type, area, or time.		
Customer Satisfaction: 4	**Customer Dissatisfaction:** 3	
Priority: High		

Requirement #: 5	**Requirement Type:** 1	**UC:** CU02_19
Description: Export the trips to PDF format.		
Rationale: Give users the possibility to access to the information without computer or internet connection, in other words, to allow to export an trip with their details to PDF format, being possible to print it.		
Originator: Similar web portals		
Fit Criterion: The user shall be able to export every of theirs trips to the format PDF.		
Customer Satisfaction: 3	**Customer Dissatisfaction:** 3	
Priority: Medium		

Requirement #: 6	**Requirement Type:** 1	**UC:** CU01_01, CU01_04, CU03_04, CU01_11
Description: Allow the user registration.		
Rationale: It will be necessary for some visitors to have access to extra functionalities such as: to organize their own trips. Therefore, it will exist a registration form.		
Originator: Similar web portals		
Fit Criterion: Provides users the possibility to sign in with all their personal data and their access data to the web portal.		
Customer Satisfaction: 4	**Customer Dissatisfaction:** 4	
Priority: High		

Requirement #: 7	**Requirement Type:** 1	**UC:** CU01_01, CU01_03, CU01_07, CU01_08, CU01_09, CU01_10, CU03_05
Description: Allow the user management.		
Rationale: Allows administrators to manage registered users, for example, enable and disable access accounts.		
Originator: Similar web portals		
Fit Criterion: The Back Office shall be able to manage users.		
Customer Satisfaction: 3	**Customer Dissatisfaction:** 4	
Priority: High		

Requirement #: 8	**Requirement Type:** 1	**UC:** CU01_02, CU01_05, CU03_01, CU03_04, CU03_05, CU01_11
Description: Allow users to log in and log out.		
Rationale: It is important for users to validate their registered accounts and ensure their accesses to the extra functionalities. The authentication is made via the username and password.		
Originator: Similar web portals		
Fit Criterion: Section where users fills in their authentication credentials to log in the web portal. Users shall be able to close his session.		
Customer Satisfaction: 4	**Customer Dissatisfaction:** 5	
Priority: High		

Requirement #: 9	**Requirement Type:** 1	**UC:** CU01_03, CU01_08, CU01_09
Description: Modify the profile data of the registered user.		
Rationale: It is important to update the user's personal data. The user will be able to modify all his personal data such as: name, email, username, and password.		
Originator: Similar web portals		
Fit Criterion: Profile area is where users can visualize and modify their personal data.		
Customer Satisfaction: 4	**Customer Dissatisfaction:** 3	
Priority: High		

Requirement #: 10	**Requirement Type:** 1	**UC:** CU02_03, CU02_05, CU02_06, CU01_12
Description: Rate or comment trip, places, attractions, events, or tours.		
Rationale: Give users the possibility to share their opinions with another users about places or trips.		
Originator: Similar web portals		
Fit Criterion: Users shall be able to register comments and ratings in their trips, points of reference, events, or places of interest. The user can classify them between 0 and 5.		
Customer Satisfaction: 4	**Customer Dissatisfaction:** 4	
Priority: High		

Requirement #: 11	**Requirement Type:** 1	**UC:** CU02_03, CU02_15, CU02_15, CU02_24
Description: Display photos or videos of trips, places, events, and tours.		
Rationale: It is important arouse the user interest via inclusion of appellative content in the details of the trip such as photos or videos.		
Originator: Similar web portals		
Fit Criterion: Users shall be able to view photo galleries or videos about trips, places, points of interest, or events.		
Customer Satisfaction: 5	**Customer Dissatisfaction:** 5	
Priority: High		

Requirement #: 12	**Requirement Type:** 1	**UC:** CU01_12
Description: Provide statistics of evaluations made by users.		
Rationale: Allows administrators to have a quantitative idea about the quality of the information of the web portal. It is also important to provide statistics about the number of registered users and the number of comments for each user.		
Originator: The tutor of the thesis		
Fit Criterion: The Back Office shall be able to display statistics via graphs.		
Customer Satisfaction: 3	**Customer Dissatisfaction:** 2	
Priority: Medium		

Requirement #: 13	**Requirement Type:** 1	**UC:** CU02_03, CU04_06
Description: Display the most rated trips by users.		
Rationale: It is important for users to view the highest rated trips because these trips have popular places.		
Originator: Similar web portals		
Fit Criterion: The Back Office shall be able to display the highest rated trips via an ordered list where the first trip will be the highest rated trip by users.		
Customer Satisfaction: 5	**Customer Dissatisfaction:** 4	
Priority: High		

Requirement #: 14	**Requirement Type:** 1	**UC:** CU02_20
Description: Consult and manage useful schedules.		
Rationale: It is important for users to know the schedules of public transport, health services, events and visits of interest for any situation that might affect the trip.		
Originator: The tutor of the thesis		
Fit Criterion: Provides users the possibility to consult schedules of public transport such us metro or bus. Also health services such us pharmacies, etc.		
Customer Satisfaction: 4	**Customer Dissatisfaction:** 3	
Priority: High		

Requirement #: 15	**Requirement Type:** 1	**UC:** CU02_21
Description: Consult the weather.		
Rationale: Many of the trips have outdoor activities, so it is important that users can see the weather in a given region, to choose the most favorable day.		
Originator: Similar web portals		
Fit Criterion: Section where users can consult the weather in several regions by days.		
Customer Satisfaction: 2	**Customer Dissatisfaction:** 2	
Priority: Low		

Requirement #: 16	**Requirement Type:** 1	**UC:** CU02_22
Description: Consult traffic bulletin.		
Rationale: The traffic of the main roads may influence the trip. Therefore, is important to show the user the level of traffic from the main roads.		
Originator: Similar web portals		
Fit Criterion: Section where users can consult, by region, the traffic of the main roads.		
Customer Satisfaction: 1	**Customer Dissatisfaction:** 1	
Priority: Low		

Requirement #: 17	**Requirement Type:** 1	**UC:** None
Description: Allow download the mobile application.		
Rationale: Give users the possibility to download the mobile application to install on their devices.		
Originator: The tutor of the thesis		
Fit Criterion: Mobile application available for download.		
Customer Satisfaction: 4	**Customer Dissatisfaction:** 3	
Priority: Medium		

Requirement #: 18	**Requirement Type:** 1	**UC:** CU02_03, CU03_02, CU03_03, CU03_06
Description: Share trips in social networks.		
Rationale: Give users the possibility to exhibit the portal web to the netizen community. Possibility to share the trips in Facebook and to make "like" to them.		
Originator: The author of the thesis		
Fit Criterion: The Front Office shall be able to integrate with Facebook account. Therefore, users shall be able to publicate and like their trips in their wall.		
Customer Satisfaction: 3	**Customer Dissatisfaction:** 2	
Priority: Medium		

Requirement #: 19	**Requirement Type:** 1	**UC:** CU02_23
Description: Send a trip by email.		
Rationale: Provides users the possibility to send their trips via email in PDF format.		
Originator: The author of the thesis		
Fit Criterion: Users shall be able to send one or more e-mails to several people.		
Customer Satisfaction: 3	**Customer Dissatisfaction:** 2	
Priority: Medium		

Requirement #: 20	**Requirement Type:** 1	**UC:** None
Description: Export the trip for GPS.		
Rationale: Give users the possibility to navigate a route using GPS by downloading a file for GPS that draw the route.		
Originator: Similar web portals		
Fit Criterion: Users shall be able to export the trip to GPS format for their computer.		
Customer Satisfaction: 3	**Customer Dissatisfaction:** 2	
Priority: Medium		

Requirement #: 21	**Requirement Type:** 1	**UC:** CU02_03, CU04_07
Description: Add trip to favourites.		
Rationale: Provides users the possibility to mark their trips favorite. Therefore, users could consult quickly their trips favorite in the future.		
Originator: Similar web portals		
Fit Criterion: Section where users lists all their trips favorite.		
Customer Satisfaction: 3	**Customer Dissatisfaction:** 2	
Priority: Medium		

A.2 NON-FUNCTIONAL REQUIREMENTS - WEB PORTAL

Requirement #: 22	**Requirement Type:** 2	**UC:** CU01_06
Description: Support for multiple languages.		
Rationale: Portugal is a country that receives plenty of tourists from other European countries. Therefore, it is important to provide information to the web portal in several languages.		
Originator: Similar web portals		
Fit Criterion: The Front Office will be in several languages with the possibility of change in every page.		
Customer Satisfaction: 4	**Customer Dissatisfaction:** 3	
Priority: Medium		

Requirement #: 23	**Requirement Type:** 2	**UC:** None
Description: Provide compatibility of the portal to several web browsers and mobile devices.		
Rationale: Allow portability to the site, ie, allow users to view the site regardless of device or web browser.		
Originator: Similar web portals		
Fit Criterion: Provides users the friendly access to information, depending on the browser and the device.		
Customer Satisfaction: 5	**Customer Dissatisfaction:** 4	
Priority: High		

Requirement #: 24	**Requirement Type:** 2	**UC:** None
Description: Look and Feel and Usability.		
Rationale: Intends a system with an appealing and easy to use interface.		
Originator: Similar web portals		
Fit Criterion: Release of prototypes and get feedback from users.		
Customer Satisfaction: 3	**Customer Dissatisfaction:** 3	
Priority: Medium		

A.3 FUNCTIONAL REQUIREMENTS - MOBILE APPLICATION

Requirement #: 25	**Requirement Type:** 1	**UC:** CU02_03, CU02_11, CU02_15, CU02_16, CU02_24, CU02_25
Description: Download trips.		
Rationale: Give users the possibility to download their trips via internet connection.		
Originator: Similar mobile applications		
Fit Criterion: Section where users shall be able to download their available trips.		
Customer Satisfaction: 4	**Customer Dissatisfaction:** 4	
Priority: High		

Requirement #: 26	**Requirement Type:** 1	**UC:** CU02_03, CU02_11, CU02_15
Description: List attractions, events, tours, and useful places (pharmacies, hospitals, etc.)		
Rationale: Provides users the possibility to list useful places around of their location. Give users possibility to navigate to them and access their information.		
Originator: Similar mobile applications		
Fit Criterion: Provides users the possibility to ask to the application, the location of points of reference, events, visits and useful places.		
Customer Satisfaction: 4	**Customer Dissatisfaction:** 3	
Priority: Medium		

Requirement #: 27	**Requirement Type:** 1	**UC:** CU02_03
Description: Show the distance and duration of a trip.		
Rationale: Maintains users informed about their trips, displaying the distance and the total duration, as well as the remaining distance and duration.		
Originator: The tutor of the thesis		
Fit Criterion: Provides users the possibility to visualize the distance and duration of the trip.		
Customer Satisfaction: 3	**Customer Dissatisfaction:** 3	
Priority: Medium		

Requirement #: 28	**Requirement Type:** 1	**UC:** CU03_02, CU03_03
Description: Post a trip or point of interest on Social Networks.		
Rationale: Give users the possibility to share their trips or points of interest on social networks such as Facebook o Twitter.		
Originator: Similar mobile applications		
Fit Criterion: Users shall be able to share their trips o points of interest on walls of social networks.		
Customer Satisfaction: 2	**Customer Dissatisfaction:** 1	
Priority: Low		

Requirement #: 29	**Requirement Type:** 1	**UC:** CU02_04, CU02_08, CU02_25
Description: Delete the downloaded information.		
Rationale: Since this is a mobile application and the trips are saved in the device memory, it is important to give users the possibility to delete a particular trip.		
Originator: The author of the thesis		
Fit Criterion: Users shall be able to delete their trips already downloaded.		
Customer Satisfaction: 3	**Customer Dissatisfaction:** 3	
Priority: Medium		

Requirement #: 30	**Requirement Type:** 1	**UC:** CU01_05, CU01_11, CU03_01
Description: Allow Login/Logout users.		
Rationale: It is important for users to validate their credentials and ensure their accesses to the extra functionalities. The authentication is made via the username and password.		
Originator: Similar mobile applications		
Fit Criterion: Section where users fills in their authentication credentials to log in the mobile application. Users shall be able to close his session.		
Customer Satisfaction: 4	**Customer Dissatisfaction:** 4	
Priority: High		

A.4 NON-FUNCTIONAL REQUIREMENTS - MOBILE APPLICATION

Requirement #: 31	**Requirement Type:** 2	**UC:** None
Description: Performance - Request has response times of less than 2 seconds.		
Rationale: Given the number of accesses expected in an application is necessary that pages are fast loading, using information exchange in light and reliable formats (eg XML and JSON).		
Originator: The tutor of the thesis		
Fit Criterion: Navigation and requests made to the application, having time less than 2 seconds.		
Customer Satisfaction: 3	**Customer Dissatisfaction:** 3	
Priority: Medium		

Requirement #: 32	**Requirement Type:** 2	**UC:** None
Description: Look and Feel and Usability.		
Rationale: Intends a system with an appealing and easy to use interface.		
Originator: Similar web portals		
Fit Criterion: Release of prototypes and get feedback from users.		
Customer Satisfaction: 3	**Customer Dissatisfaction:** 3	
Priority: High		

B

SYSTEM REQUIREMENTS SPECIFICATION (SRS)

B.1 INTRODUCTION

The section introduces the system requirements specification (SRS) for the Web Portal for Touristic Social Communities (WPTSC) to its readers.

B.1.1 *Objectives*

The purpose of the user requirement specifications is to detail the stakeholder needs, expectations, definitions, objectives, context, constraints, and interfaces. The stakeholders are: Minho University advisor and a Masters student in computer science. All models constructed in this document follow the standard UML notations.

B.1.2 *Reference*

The following references are used as sources of requirement analysis:

- Project proposal.
- Formal meeting and informal discussion.
- TouristEye Website.
- Yahoo Trip Planner Website.
- TripIt Website.

B.1.3 *Scope*

The product is basically a web portal where the users can plan, organize their trips and/or define what touristic resources the users would like to visit and keep track of activities performed. The WPTSC will be into an existent platform dedicated to tourism at Ubiwhere Company.

B.1.4 *Overview*

This specification is organized into the following sections:

- Introduction, which introduces the specification for the WPTSC to its readers.
- Overall Description, which provides a brief, high level description of the WPTSC including its definition, context, main functions, features of the users, constraints, and assumptions.
- Specific Requirements, which specifies the functional system requirements in terms of a use case model consisting of each external's use cases and use case paths. Similarly are specified external interface requirements.

B.2 OVERALL DESCRIPTION

This section documents the context of the WPTSC in terms of the significant externals with which it interacts. Tourism Websites give support to 3 different phases of a trip: before, during and after the trip.

Before the trip	The tourist use to plan the trip and can choose: • Places. • Time interval (hours, days or months). • Events (concerts, tours, etc.). • Accommodation (hostel, hotel, etc.). • Food (restaurants, pizzerias, etc.). • Others. Usually, the tourist consults tourism webs, maps, photos, videos, textual description or any information in social networks where he can see the experiences of other tourists.
During the trip	At this point the tourists are on a trip and can publish in social networks, where he is, who is he with, and also if he is participating in an event. It can be observed that, the tourists usually during his/her trip use mobile devices as smartphones or small laptops.
After the trip	At this point the tourist shares experiences, suggestions of places, hotels, restaurants, and other touristic resources. A report of his trip is created with their experiences.

Table 39.: Time phases of tourism in the Web

B.2.1 *Context*

This dissertation will be developed partially at the Minho University, with the objective of integrating the WPTSC to mobile platform dedicated to tourism. Figure below shows the platform dedicated to tourism.

B.2.2 *Product Features*

The data system consists two main parts a central database management system for the uploading, storage, and management of data, and a client application to allow users access and interact with the data. Diverse end-users can access, search, analyze, visualize, edit, and create trips or touristic resources. Types of data will include text, photos, geo-location information, maps, and videos.

B.2.3 *Stakeholders*

Ubiwhere Staff:

- Managers.
- Advisor.
- Customer Representatives, who must approve it.
- Testers, who must ensure that the requirements are valid and whose tests must validate the requirements.

Minho University Staff:

- Professor advisor.
- Master student in informatics.

Users, who are any individual or group:

- Potential Tourist.
- Tourism Agency.
- Hotel Personnel.

B.2.4 *Constrains*

The followings are the constraints identified:

- The client module should be able to run on a majority of personal computers with the browser Firefox o Google Chrome.
- The system should use PosgresSQL as database.
- The default language will be English.
- The system should be finish until end days of October 2014.

B.2.5 *Business Goal*

The business goal of the WPTSC system is to take advantage of the Internet and World Wide Web to radically improve the way users plan trips. It is important to plan a trip, because when we planned a trip plan will be money spent.

B.3 SPECIFIC REQUIREMENTS

B.3.1 *External Requirements*

User Interfaces

The user interface will be simple and user-friendly, using terminology commonly understood by common users. The system will have a simple interface, consistent with industry standard interfaces, to eliminate the need for user training of infrequent users. User testing will be used to ensure the user interface is clear (simple, commonly understood vocabulary, intuitive to use without training), complete (users can perform all functions from the interface), and consistent (buttons and wording are the same throughout the system).

Hardware Interfaces

No extra hardware interfaces are needed. The system will use the standard hardware and data communications resources provided by the Ubiwhere Company. This includes, but is not limited to, the general Ethernet network/T1 connection at the server/hosting site, network servers, and network management tools. The hardware of user client can be personal computers, personal digital assistants (PDAs), and smart phones used by users to communicate with each other and with the WPTSC over the Internet.

Software Interfaces

The system will use the standard software resources preferably open source. This includes, but is not limited to, PHP/Java Scripts, PosgresSQL server, and Apache server. The WPTSC interacts, directly with a Browser like Firefox or Google Chrome, which is the software tool that runs on employee and user personal computers.

Communications Interfaces

The system will use the communications resources provided by the Ubiwhere Company. This includes, but is not limited to, HTTP protocol for communication with the web browser and the web server and TCP/IP network protocol with HTTP protocol.

B.3.2 *Functional Requirements*

Use cases

ACTORS The actors of the system can be divided in:

- Anonymous user: This user is an individual who visits the portal web and does not log in using a user ID and password; therefore he only has permissions to query the contents of the Web.

- Authenticated user: This is a visitor to the web portal who logs in using a unique user ID and password. This type of visitor has certain permissions, such as organizing a trip or changing his profile. The authenticated user has all the permissions of anonymous users.

- Administrator: This actor has access to all the system's DB. So, he can modify, add or remove any content of the system. He owns all the permissions of the authenticated user.

PACKAGE `user administration`

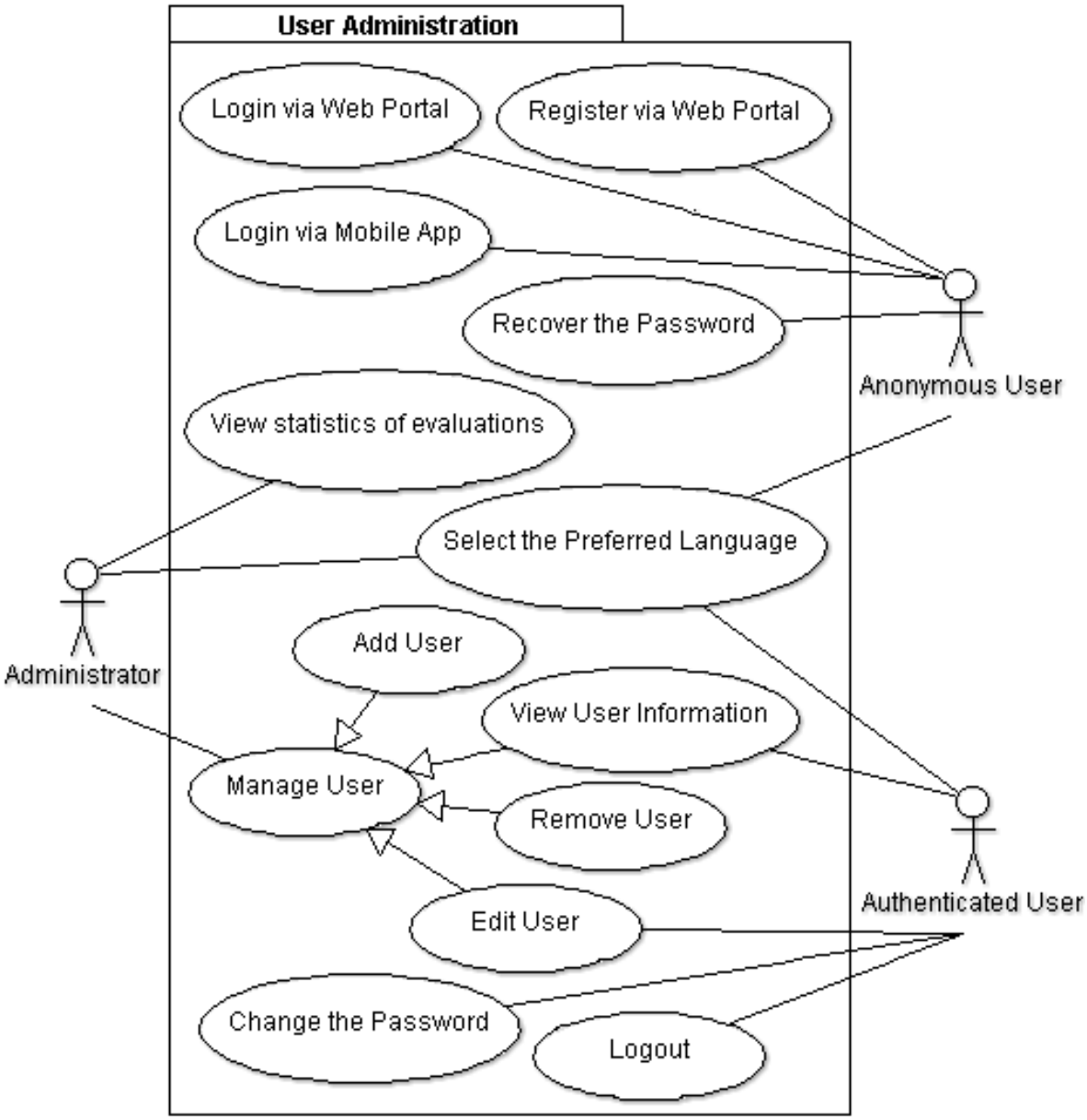


Figure 48.: Use case diagram for package `User Administration`

Use Case `Register via Web Portal`

Use Case ID:		CU01_01	
Use Case Name:		Register via Web Portal	
Author:		Miguel Angel Lopez Mamani	
Date:		March 1, 2012	
Actors:		Anonymous User	
Brief Description:		This is a use case of user registration into the web portal through User ID and password.	
Preconditions:		The web portal must be online and available for use and the actor is not already logged.	
		Actor Input	**System Response**
Flow of Events	1	Presses the link `Sign Up`.	
	2		Displays the registration form to the user.
	3	Enters registration information (name, email, and password) and accept the terms of use and the privacy policy.	
	4		Verifies the email, name, and password.
	5		Saves the data.
	6		Sends a confirmation email to the user.
Alternative Flow: 4) Invalid information		**Actor Input**	**System Response**
	1		Identifies which part is erroneous.
	2		Shows the error to the actor.
	3		Go to step 3 of main flow.
Post-conditions:		The actor is registered and logged into the system.	
Priority:		High	
Involved Class:		User, Role	

Table 40.: Textual description of use case `Register via Web Portal`

Use Case `Login via Web Portal`

Use Case ID:		CU01_02	
Use Case Name:		Login via Web Portal	
Author:		Miguel Angel Lopez Mamani	
Date:		March 1, 2012	
Actors:		Anonymous User	
Brief Description:		This is a use case for logging in a user.	
Preconditions:		The web portal must be online and available for use	
		Actor Input	**System Response**
Flow of Events	1	Enters his user ID and password.	
	2	Presses the link `Login`.	
	3		Checks the validity of the credentials.
	4		Displays all features available for the role of the user.
Alternative Flow: 3) Invalid User ID or Password		**Actor Input**	**System Response**
	1		Displays an error message.
	2		Go to step 1 of main flow.
Post-conditions:		The actor is authenticated and the system displays all features available for the role of the user.	
Priority:		High	
Involved Class:		User, Role, Session	

Table 41.: Textual description of use case `Login via Web Portal`

Use Case `Change the Password`

Use Case ID:		CU01_03	
Use Case Name:		Change the Password	
Author:		Miguel Angel Lopez Mamani	
Date:		March 1, 2012	
Actors:		Authenticated User	
Brief Description:		The actor modifies his password	
Preconditions:		The actor is logged.	
		Actor Input	**System Response**
Flow of Events	1	Presses the link `Change Password`	
	2		Displays a form to change the password
	3	Enters current password and the new password twice.	
	4		Checks the validity of credentials.
	5		Saves the new password.
	6		Shows a success message.
Alternative Flow: 4) Invalid or weak password		**Actor Input**	**System Response**
	1		Displays an error message.
	2		Go to step 3 of main flow.
Post-conditions:		The password is saved into the database.	
Priority:		Medium	
Involved Class:		User, Role	

Table 42.: Textual description of use case `Change the Password`

Use Case `Recover the Password`

Use Case ID:		CU01_04	
Use Case Name:		Recover the Password	
Author:		Miguel Angel Lopez Mamani	
Date:		March 1, 2012	
Actors:		Anonymous User	
Brief Description:		If the user forgets his user name or password, the system will generate a new password and will send it to his email address.	
Preconditions:		The web portal must be online and available for use and the actor is not logged.	
		Actor Input	**System Response**
Flow of Events	1	Presses the link `Request new password`.	
	2	Provides his email address or user name.	
	3		Checks if there is an account with the provided email address or user name.
	4		Sets the password of this user to a randomly generated new password.
	5		Sends the password to the user's email.
	6	Receives the new password.	
Alternative Flow: 3)Invalid email or user name		**Actor Input**	**System Response**
	1		Displays a message saying that a new password has been sent to the provided email address.
Post-conditions:		The generated password is saved into the database.	
Priority:		High	
Involved Class:		User, Role	

Table 43.: Textual description of use case `Recover the Password`

Use Case `Logout`

Use Case ID:		CU01_05	
Use Case Name:		Logout	
Author:		Miguel Angel Lopez Mamani	
Date:		March 1, 2012	
Actors:		Authenticated User	
Brief Description:		This use case describes the logout process.	
Preconditions:		The actor is logged.	
		Actor Input	**System Response**
Flow of Events	1	Presses the link `Logout`	
	2		Finishes the session.
	3		Displays a message `Successful Signed Off`.
	4		Re-direct the user to the Main Window.
Alternative Flow: 1) Closes the window		**Actor Input**	**System Response**
	1		Finishes the session.
Post-conditions:		Displays the Main Window.	
Priority:		High	
Involved Class:		User, Role, Session	

Table 44.: Textual description of use case `Logout`

Use Case `Select the Preferred Language`

Use Case ID:		CU01_06	
Use Case Name:		Select the Preferred Language	
Author:		Miguel Angel Lopez Mamani	
Date:		March 1, 2012	
Actors:		Anonymous User, Authenticated User, Administrator	
Brief Description:		This use case allows to the actor to select a different language for the web portal, where the default language was configured by the Administrator.	
		Actor Input	**System Response**
Flow of Events	1	Presses the link `Choose Language`.	
	2		Displays the available languages: English, Portuguese, and Spanish.
	3	Choose the language.	
	4		Changes the language of the web portal.
Preconditions:		The web portal must be online and available for use.	
Post-conditions:		The web portal is translated to the selected language.	
Priority:		Low	
Involved Class:		Language	

Table 45.: Textual description of use case `Select the Preferred Language`

Use Case `Add User`

Use Case ID:		CU01_07	
Use Case Name:		Add User	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2012	
Actors:		Administrator	
Brief Description:		This use case describes the process used by the Administrator to add a new user to the web portal.	
Preconditions:		The actor is logged.	
		Actor Input	**System Response**
Flow of Events	1	Presses the link `new user`.	
	2		Displays a form with the following fields: username and name.
	3	Enters the user information.	
	4	Chooses the role.	
	5	Presses the link “Create”.	
	6		Validates the information entered by the user.
	7		Creates a new user.
Alternative Flow: 6) Invalid information		**Actor Input**	**System Response**
	1		Shows an error message.
	2		Go to step 3 of main flow.
Post-conditions:		A new user account was created.	
Priority:		Medium	
Involved Class:		User, Role	

Table 46.: Textual description of use case `Add User`

Use Case `Edit User`

Use Case ID:		CU01_08	
Use Case Name:		Edit User	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2012	
Actors:		Authenticated User, Administrator	
Brief Description:		This use case describes the process of editing a user account.	
Preconditions:		The actor is logged.	
		Actor Input	**System Response**
Flow of Events	1	Presses the link `edit user`.	
	2		Displays a form filled with the following fields: username and name.
	3	Changes the desired fields.	
	4	Presses the link `Save`.	
	5		Validates the information.
	6		Saves the new user data.
Alternative Flow: 5) Invalidate information		**Actor Input**	**System Response**
	1		Shows an error message.
	2		Go to step 2 of main flow.
Post-conditions:		The new user data is saved.	
Priority:		Medium	
Involved Class:		User, Role	

Table 47.: Textual description of use case `Edit User`

Use Case `View User Information`

Use Case ID:		CU01_09	
Use Case Name:		View User Information	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2012	
Actors:		Authenticated User, Administrator	
Brief Description:		This use case describes the process of visualizing an user information.	
Preconditions:		The actor is logged.	
		Actor Input	**System Response**
Flow of Events	1	Presses the link to view user information.	
	2		Displays the user information.
Post-conditions:		The user information was displayed.	
Priority:		Medium	
Involved Class:		User, Role	

Table 48.: Textual description of use case `View User Information`

Use Case `Remove User`

Use Case ID:		CU01_10	
Use Case Name:		Remove User	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2012	
Actors:		Administrator	
Brief Description:		This use case describes the process of deleting a user account from the web portal.	
Preconditions:		The actor is logged.	
		Actor Input	**System Response**
Flow of Events	1	Select the user account to be deleted.	
	2	Presses the link `Remove user`.	
	3		Displays a message of confirmation like `Are you sure?`
	4	Replies with an affirmative answer.	
	5		Deletes the user account.
Alternative Flow: 4) Negative answer		**Actor Input**	**System Response**
	1		Cancel the operation.
	2		Go to step 1 of main flow.
Post-conditions:		The user account was deleted.	
Priority:		Medium	
Involved Class:		User, Role	

Table 49.: Textual description of use case `Remove User`

Use Case `Login via Mobile Application`

Use Case ID:		CU01_11	
Use Case Name:		Login via Mobile Application	
Author:		Miguel Angel Lopez Mamani	
Date:		March 1, 2014	
Actors:		Anonymous User	
Brief Description:		This is a use case for logging in a user.	
		Actor Input	**System Response**
Flow of Events	1	Enters his user ID and password.	
	2	Presses the link `Login`.	
	3		Checks the validity of the credentials.
	4		Displays all features available for the role of the user.
Alternative Flow: 3) Invalid User ID or Password		**Actor Input**	**System Response**
	1		Displays an error message.
	2		Go to step 1 of main flow.
Preconditions:		The mobile application must be available for use.	
Post-conditions:		The actor is authenticated and the system displays all features available for the role of the user.	
Priority:		High	
Involved Class:		User, Role, Session	

Table 50.: Textual description of use case `Login via Mobile application`

Use Case `View statistics of evaluations`

Use Case ID:		CU01_12	
Use Case Name:		View statistics of evaluations	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2014	
Actors:		Administrator	
Brief Description:		This use case describes the process of visualizing the statistics of evaluations made by users.	
Preconditions:		The actor is logged.	
		Actor Input	**System Response**
Flow of Events	1	Presses the link `View statics`.	
	2		Calculates the statics.
	3		Displays the statics.
Post-conditions:		The statics of evaluations made by user was displayed.	
Priority:		Low	
Involved Class:		User, Trips	

Table 51.: Textual description of use case `View statistics of evaluations`

PACKAGE trip management

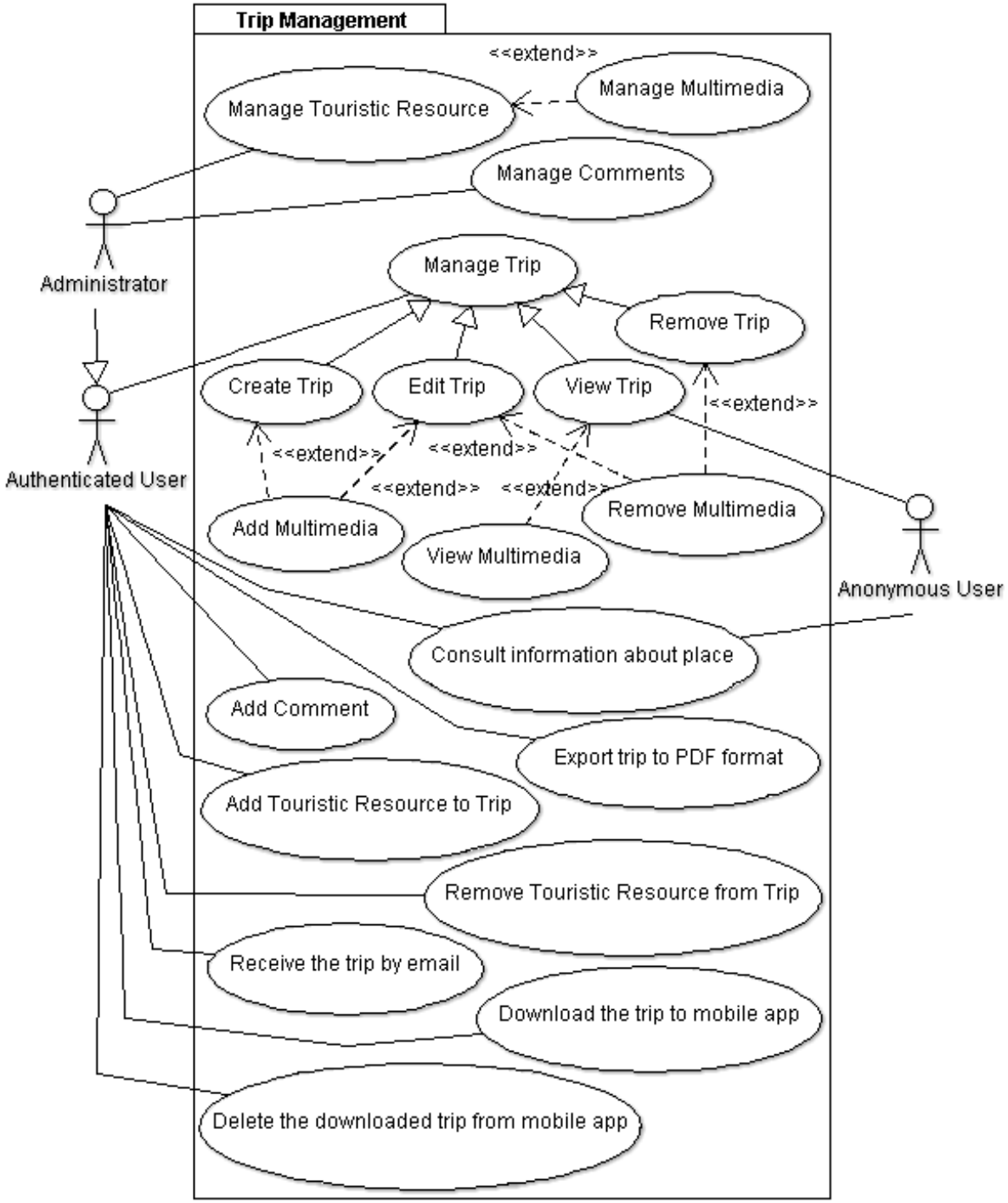


Figure 49.: Use case diagram for package Trip Management

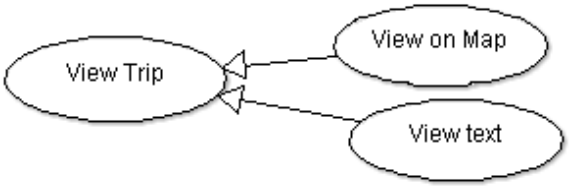


Figure 50.: Disaggregation of use case `View Trip`

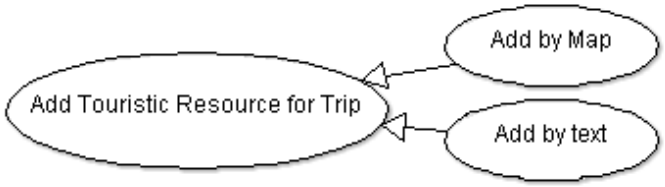


Figure 51.: Disaggregation of use case `Manage Multimedia`

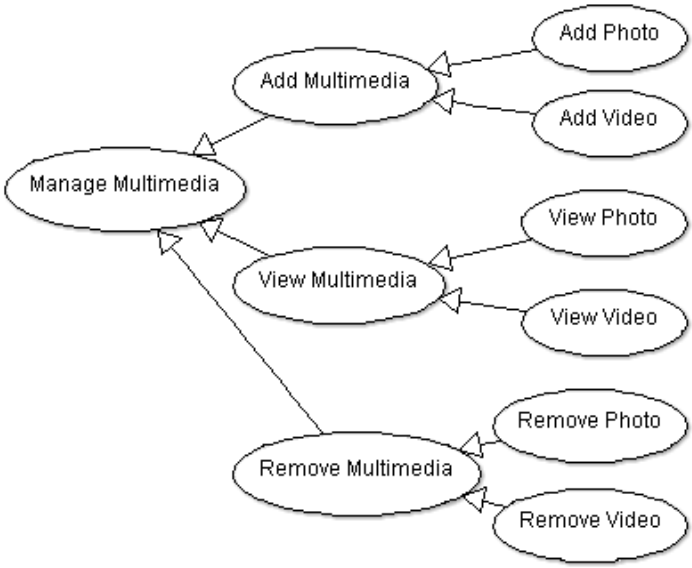


Figure 52.: Trip Management

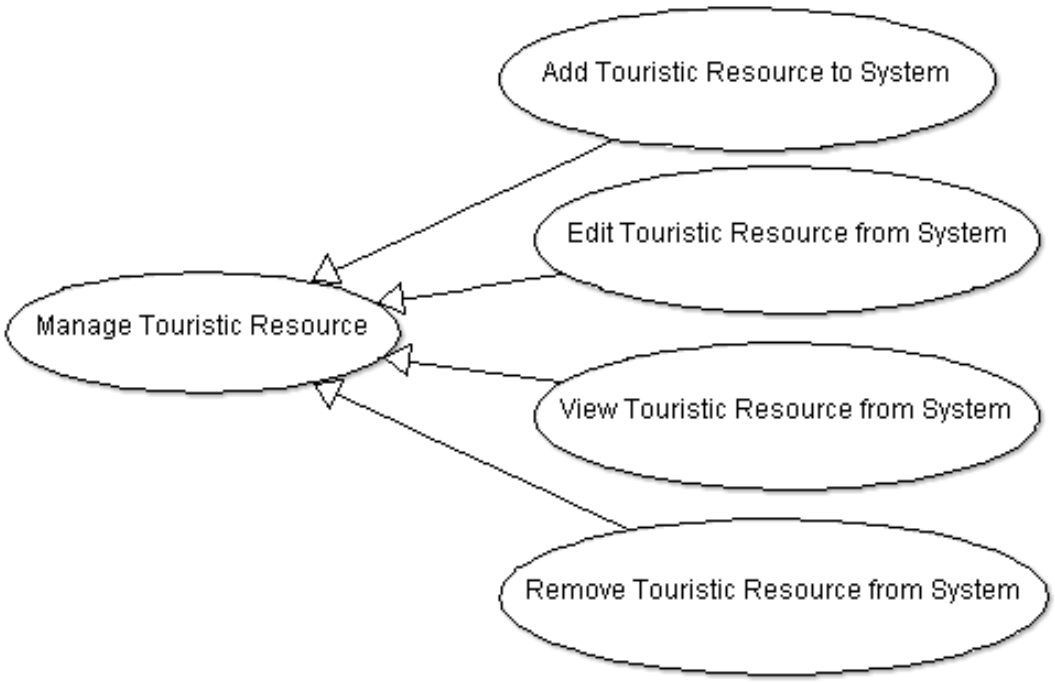


Figure 53.: Disaggregation of use case `Manage Touristic Resource`

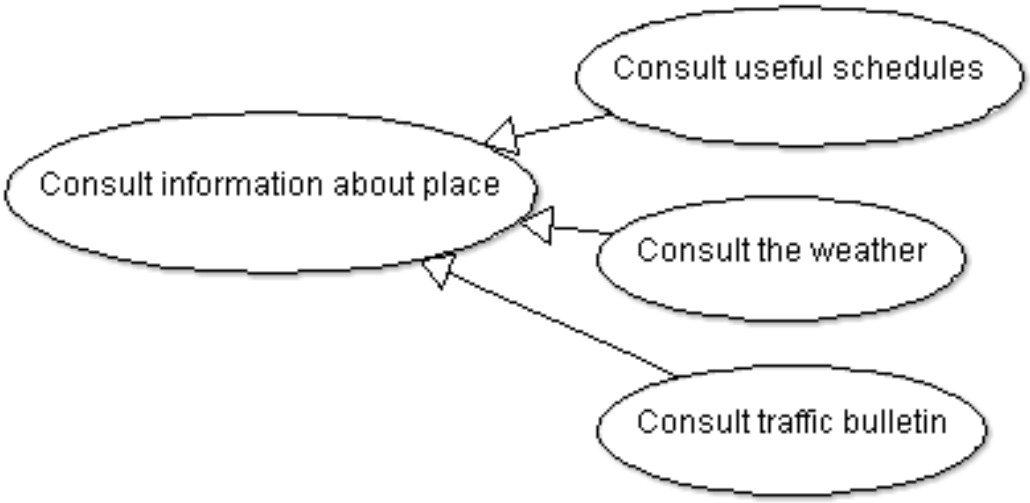


Figure 54.: Disaggregation of use case `Consult information about place`

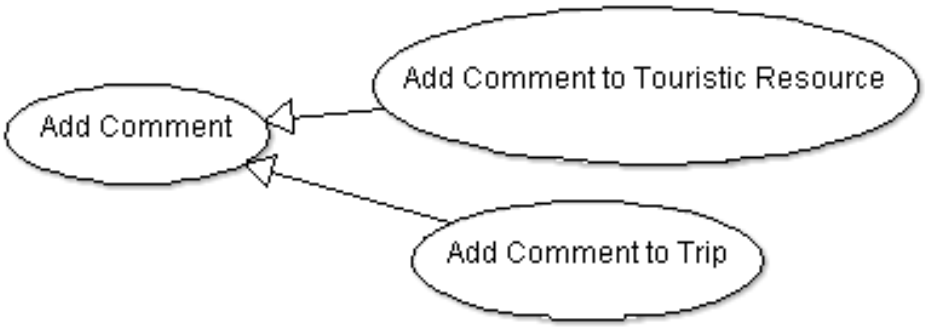


Figure 55.: Disaggregation of use case `Add Comment`

Use Case `Create Trip`

Use Case ID:		CU02_01	
Use Case Name:		Create Trip	
Author:		Miguel Angel Lopez Mamani	
Date:		March 1, 2012	
Brief Description:		Process of creating a new trip in the web portal.	
		Actor Input	**System Response**
Flow of Events	1	Presses the link `Start Trip`	
	2		Displays a form with the followings fields: title, dates and destinations.
	3	Enters the value for each field.	
	4	Chooses the privacy of the trip (public or private).	
	5	Presses the link `Create Trip`.	
	6		Validates the information.
	7		<<Extended by>> Add Photo
	8		<<Extended by>> Add Video
	9		Saves the data.
Alternative Flow: 6) Invalid Information		**Actor Input**	**System Response**
	1		Displays an error message.
	2		Go to step 3 of main flow.
Actors:		Authenticated User, Administrator	
Preconditions:		The actor is logged.	
Post-conditions:		The trip was created.	
Priority:		High	
Involved Class:		Trip	

Table 52.: Textual description of use case `Create Trip`

Use Case `Edit Trip`

Use Case ID:		CU02_02	
Use Case Name:		Edit Trip	
Author:		Miguel Angel Lopez Mamani	
Date:		March 1, 2012	
Actors:		Authenticated User, Administrator	
Brief Description:		This use case describes the process of editing a trip.	
Preconditions:		The actor is logged.	
		Actor Input	**System Response**
Flow of Events	1	Presses the link `Edit Trip`.	
	2		Displays the trip to be edited.
	3	Modifies the fields of the trip.	
	4	Presses the link `Save`.	
	5		Validates the information.
	6		<<Extended by>> Add Photo
	7		<<Extended by>> Add Video
	8		Saves the data.
Alternative Flow: 5) Invalid Information		**Actor Input**	**System Response**
	1		Displays an error message.
	2		Go to step 3 of main flow.
Post-conditions:		The new data for the trip was saved into the database.	
Priority:		High	
Involved Class:		Trip	

Table 53.: Textual description of use case `Edit Trip`

Use Case `View Trip`

Use Case ID:		CU02_03	
Use Case Name:		View Trip	
Author:		Miguel Angel Lopez Mamani	
Date:		March 1, 2012	
Actors:		Anonymous User, Authenticated User, Administrator	
Brief Description:		This use case describes the process of visualizing a trip.	
Preconditions:		The web portal must be online and available for the use. In case, of the mobile application, this must available for the use.	
		Actor Input	**System Response**
Flow of Events	1	Presses the link `Trip`.	
	2		Displays a list of trips associated to the actor.
	3	Selects a trip.	
	4		Display the trip.
Post-conditions:		The trip is visualized.	
Priority:		High	
Involved Class:		Trip	

Table 54.: Textual description of use case `View Trip`

Use Case `Remove Trip`

Use Case ID:		CU02_04	
Use Case Name:		Remove Trip	
Author:		Miguel Angel Lopez Mamani	
Date:		March 1, 2012	
Actors:		Authenticated User, Administrator	
Brief Description:		This use case describes the process of deleting a trip.	
Preconditions:		The actor is logged on his trip.	
		Actor Input	**System Response**
Flow of Events	1	Presses the link `Trip`	
	2		Displays a list of trips associated to the actor.
	3	Selects a trip.	
	4	Presses the link `Delete`.	
	5		Displays a confirmation message.
	6	Confirm the deletion.	
	7		Deletes the trip.
Alternative Flow: 6) Trip confirmation is canceled		**Actor Input**	**System Response**
	1		Go to step 2 of main flow.
Post-conditions:		The deleted trip was removed from the database.	
Priority:		Medium	
Involved Class:		Trip	

Table 55.: Textual description of use case `Remove Trip`

Use Case `Add Comment to Trip`

Use Case ID:		CU02_05	
Use Case Name:		Add Comment to Trip	
Author:		Miguel Angel Lopez Mamani	
Date:		March 1, 2012	
Actors:		Authenticated User	
Brief Description:		This Use Case describes how to add a comment to a Trip (on success). Do nothing (on failure).	
Preconditions:		The actor has selected a Trip.	
		Actor Input	**System Response**
Flow of Events	1	Presses the text box `add comment`.	
	2		Enables text box for a comment.
	3	Enters a comment.	
	4	Presses the link `Comment`.	
	5		Checks the validity of the comment.
	6		Saves the comment.
Alternative Flow: 5) Invalid Comment		**Actor Input**	**System Response**
	1		Shows a warning message.
	2		Do not publish the comment.
Post-conditions:		The comment is saved in the database.	
Priority:		High	
Involved Class:		Comment, Trip	

Table 56.: Textual description of use case `Add Comment to Trip`

Use Case `Add Comment to Touristic Resource`

Use Case ID:		CU02_06	
Use Case Name:		Add Comment to Touristic Resource	
Author:		Miguel Angel Lopez Mamani	
Date:		March 1, 2012	
Actors:		Authenticated User	
Brief Description:		This Use Case describes how to add a comment to a touristic resource.	
Preconditions:		The actor has selected a touristic resource.	
Post-conditions:		The comment is saved in the database (on success).	
		Actor Input	**System Response**
Flow of Events	1	Presses the text box `add comment`.	
	2		Enables the text box for comments
	3	Enters a comment.	
	4	Presses the link `Comment`.	
	5		Checks the validity of the comment.
	6		Saves the comment.
Alternative Flow: 5) Invalid Comment		**Actor Input**	**System Response**
	1		Shows a warning message.
	2		Do not publish the comment.
Priority:		High	
Involved Class:		Comment, Touristic Resource	

Table 57.: Textual description of use case `Add Comment to Touristic Resource`

Use Case `Add Touristic Resource to Trip`

Use Case ID:		CU02_07	
Use Case Name:		Add Touristic Resource to Trip	
Author:		Miguel Angel Lopez Mamani	
Date:		March 1, 2012	
Actors:		Authenticated User	
Brief Description:		This Use Case describes how add a touristic resource to a trip.	
Preconditions:		The actor is on his trip	
Post-conditions:		The association of touristic resource and trip is saved in the database every 5 seconds.	
		Actor Input	**System Response**
Flow of Events	1	Selects a touristic resource from the map.	
	2	Drags and drops the touristic resource into a day of the trip represented by a container.	
	3		Associates the day with the touristic resource.
Alternative Flow: 1) Selects a touristic resource using a list		**Actor Input**	**System Response**
	1	Select a touristic resource from a list.	
	2	Go to step 2 of main flow.	
Priority:		High	
Involved Class:		Trip, Touristic Resource	

Table 58.: Textual description of use case `Add Touristic Resource to Trip`

Use Case `Remove Touristic Resource from Trip`

Use Case ID:		CU02_08	
Use Case Name:		Remove Touristic Resource from Trip	
Author:		Miguel Angel Lopez Mamani	
Date:		March 1, 2012	
Brief Description:		This Use Case describes how to remove a touristic resource from a trip.	
		Actor Input	**System Response**
Flow of Events	1	Selects a day from the list.	
	2	Presses the icon `Remove`.	
	3		Disassociate the day with the touristic resource.
Actors:		Authenticated User	
Preconditions:		The actor has selected a trip.	
Post-conditions:		The association between touristic resource and trip was removed from the database.	
Priority:		Medium	
Involved Class:		Trip, Touristic Resource	

Table 59.: Textual description of use case `Remove Touristic Resource from Trip`

Use Case `Add Touristic Resource to System`

Use Case ID:		CU02_09	
Use Case Name:		Add Touristic Resource to System	
Author:		Miguel Angel Lopez Mamani	
Date:		March 1, 2012	
Actors:		Administrator	
Brief Description:		This Use Case describes the process of adding a touristic resource to the System.	
		Actor Input	**System Response**
Flow of Events	1	Presses the link `Add touristic resource`.	
	2		Displays a form for adding a new touristic resource.
	3	Enters the title and description.	
	4	Selects the localization on the map.	
	5	Presses the link `Create`.	
	6		Validates the title and description.
	7		Saves the data.
Alternative Flow: 6) Invalid Information		**Actor Input**	**System Response**
	1		Displays an error text.
	2		Go to step 3 of main flow.
Preconditions:		The actor is logged.	
Post-conditions:		The touristic resource was saved in the database.	
Priority:		High	
Involved Class:		Touristic Resource	

Table 60.: Textual description of use case `Add Touristic Resource to System`

Use Case `Edit Touristic Resource from System`

Use Case ID:		CU02_10	
Use Case Name:		Edit Touristic Resource from System	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2012	
Actors:		Administrator	
		Actor Input	**System Response**
Flow of Events	1	Presses the link `Touristic Resource`.	
	2		Displays a list of touristic resources.
	3	Selects the touristic resource.	
	4	Presses the icon `Edit`.	
	5		Displays the touristic resource for its editing.
	6	Edits title and description.	
	7	Modifies the localization on the map.	
	8		<<Extended by>> Add Photo
	9		<<Extended by>> Remove Photo
	10		<<Extended by>> Add Video
	11		<<Extended by>> Remove Video
	12		Validates the title and description.
	13		Saves the data.
Alternative Flow: 12) Invalid Information		**Actor Input**	**System Response**
	1		Displays an error message.
	2		Go to step 6 of main flow.

Brief Description:	This use case describes the process of editing a touristic resource from System.
Preconditions:	The actor is logged.
Post-conditions:	The touristic resource was saved in the database.
Priority:	High
Involved Class:	Touristic Resource

Table 61.: Textual description of use case `Edit Touristic Resource from System`

Use Case `View Touristic Resource from System`

Use Case ID:		CU02_11	
Use Case Name:		View Touristic Resource from System	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2012	
Actors:		Anonymous User, Authenticated User, Administrator	
Brief Description:		This use case describes the process of visualizing a touristic resource.	
Preconditions:		The web portal must be online and available for the use.	
		Actor Input	**System Response**
Flow of Events	1	Presses the link of the `Touristic Resource`.	
	2		Displays the touristic resource content.
Post-conditions:		The touristic resource content was visualized to the user.	
Priority:		High	
Involved Class:		Touristic Resource	

Table 62.: Textual description of use case `View Touristic Resource from System`

Use Case `Remove Touristic Resource from System`

Use Case ID:		CU02_12	
Use Case Name:		Remove Touristic Resource from System	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2012	
Actors:		Administrator	
Brief Description:		This use case describes the process of deleting a touristic resource from system.	
Preconditions:		The actor is logged.	
		Actor Input	**System Response**
Flow of Events	1	Presses the link `Touristic Resource`.	
	2		Displays a list of touristic resources associated to the actor.
	3	Selects a touristic resource.	
	4	Activates the icon `Delete`.	
	5		Displays a confirmation message.
	6	Confirm the deletion.	
	7		Delete the touristic resource.
Alternative Flow: 6) Touristic Resource removal is canceled		**Actor Input**	**System Response**
	1	Cancel the touristic resource removal.	
	2		Go to step 2 of main flow.
Post-conditions:		The deleted touristic resource was removed from the database.	
Priority:		High	
Involved Class:		Touristic Resource	

Table 63.: Textual description of use case `Remove Touristic Resource from System`

Use Case `Add Photo`

Use Case ID:		CU02_13	
Use Case Name:		Add Photo	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2012	
Actors:		Authenticated User, Administrator	
Brief Description:		Allows actor to add a photo to the touristic resource or trip	
Preconditions:		The actor has selected trip or touristic resource.	
		Actor Input	**System Response**
Flow of Events	1	Presses the link `Add Photos`.	
	2		Shows a dialog box to select file.
	3	Selects a photo.	
	4		Registers the localization of selected photo.
	5	Presses the link `Upload Photo`.	
	6		Validates the size (max 16Mb).
	7		Validates the extension (png, gif, jpg).
	8		Validates the width and height.
	9		Validates the filename.
	10		Saves the photo.
Alternative Flow: 6) Invalid Size		**Actor Input**	**System Response**
	1		Displays an error message.
	2		Go to step 3 of main flow.
Alternative Flow: 7) Invalid extension		**Actor Input**	**System Response**
	1		Displays an error message.
	2		Go to step 3 of main flow.

Alternative Flow: 8) Invalid width or height		**Actor Input**	**System Response**
	1		Displays a warning message.
	2		Changes the size.
	3		Go to step 9 of main flow.
Alternative Flow: 9) Invalid filename		**Actor Input**	**System Response**
	1		Displays a warning message.
	2		Changes the name of file to a valid name.
	3		Go to step 10 of main flow.
Post-conditions:		The photo is uploaded into the server.	
Priority:		High	
Involved Class:		Trip, Touristic Resource, Photo, Multimedia	

Table 64.: Textual description of use case `Add Photo`

Use Case `View Video`

Use Case ID:		CU02_14	
Use Case Name:		View Video	
Author:		Miguel Angel Lopez Mamani	
Date:		March 20, 2012	
Actors:		Anonymous User, Authenticated User, Administrator	
Brief Description:		This use case describes the process of visualizing a video.	
Preconditions:		The web portal must be online and available for the use.	
		Actor Input	**System Response**
Flow of Events	1	Presses the link `Play`.	
	2		Plays the Video.
Post-conditions:		The Video is visualized.	
Priority:		Low	
Involved Class:		Video, Multimedia	

Table 65.: Textual description of use case `View Video`

Use Case `Add Video`

Use Case ID:		CU02_15	
Use Case Name:		Add Video	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2012	
Actors:		Authenticated User, Administrator	
Brief Description:		Allows actor to add a video to touristic resource or trip.	
		Actor Input	**System Response**
Flow of Events	1	Presses the link `Add Videos`.	
	2		Shows a dialog box to select file.
	3	Selects a video.	
	4		Registers the localization of selected photo.
	5	Presses the link `Upload Video`.	
	6		Validates the max size.
	7		Validates the extension (avi, mpg, mp4).
	8		Validates the filename.
	9		Saves the video.
Alternative Flow: 4) Invalid Size		**Actor Input**	**System Response**
	1		Displays an error message.
	2		Go to step 1 of main flow.
Alternative Flow: 5) Invalid extension		**Actor Input**	**System Response**
	1		Displays an error message.
	2		Go to step 2 of main flow.
Alternative Flow: 6) Invalid filename		**Actor Input**	**System Response**
	1		Displays a warning message.
	2		Changes the name of file.
	3		Go to step 9 of main flow.
Preconditions:		The actor is on his trip or a touristic resource.	

Post-conditions:	The video is uploaded into the server.
Priority:	Low
Involved Class:	Trip, Touristic Resource, Video, Multimedia

Table 66.: Textual description of use case `Add Video`

Use Case `View Photo`

Use Case ID:		CU02_16	
Use Case Name:		View Photo	
Author:		Miguel Angel Lopez Mamani	
Date:		March 20, 2012	
Actors:		Anonymous User, Authenticated User, Administrator	
Brief Description:		This use case describes the process of visualizing a photo.	
Preconditions:		The web portal must be online and available for the use.	
		Actor Input	**System Response**
Flow of Events	1	Presses the link of the Photo from the gallery.	
	2		Displays the Photo.
Post-conditions:		The Photo is visualized.	
Priority:		High	
Involved Class:		Photo, Multimedia	

Table 67.: Textual description of use case `View Photo`

Use Case `Remove Video`

Use Case ID:		CU02_17	
Use Case Name:		Remove Video	
Author:		Miguel Angel Lopez Mamani	
Date:		March 20, 2012	
Actors:		Authenticated User, Administrator	
Brief Description:		This use case describes the process of deleting a video.	
Preconditions:		The actor is logged on his trip or touristic resource.	
		Actor Input	**System Response**
Flow of Events	1	Selects a Video.	
	2	Presses the link `Delete`.	
	3		Displays a confirmation message.
	4	Confirm the deletion.	
	5		Deletes the video.
Alternative Flow: 6) Video confirmation is canceled		**Actor Input**	**System Response**
	1		Go to step 1 of main flow.
Post-conditions:		The deleted video was removed from the database and of the file system.	
Priority:		Low	
Involved Class:		Video, Multimedia	

Table 68.: Textual description of use case `Remove Video`

Use Case `Remove Photo`

Use Case ID:		CU02_18	
Use Case Name:		Remove Photo	
Author:		Miguel Angel Lopez Mamani	
Date:		March 20, 2012	
Actors:		Authenticated User, Administrator	
Brief Description:		This use case describes the process of deleting a photo.	
Preconditions:		The actor is logged on his trip or touristic resource.	
		Actor Input	**System Response**
Flow of Events	1	Selects the photo from the gallery.	
	2	Presses the link `Delete`.	
	3		Displays a confirmation message.
	4	Confirm the deletion.	
	5		Deletes the photo.
Alternative Flow: 6) Photo confirmation is canceled		**Actor Input**	**System Response**
	1		Go to step 1 of main flow.
Post-conditions:		The deleted photo was removed from the database and of the file system.	
Priority:		Low	
Involved Class:		Photo, Multimedia	

Table 69.: Textual description of use case `Remove Photo`

Use Case `Export trip to PDF format`

Use Case ID:		CU02_19	
Use Case Name:		Export trip to PDF format	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2014	
Actors:		Authenticated User, Administrator	
Brief Description:		Allows actor to generate a PDF file from every trip.	
		Actor Input	**System Response**
Flow of Events	1	Click the link `Export to PDF`.	
	2		Generate the PDF file.
	3		Save the PDF file.
Alternative Flow: 2) Actor cancel		**Actor Input**	**System Response**
	2		Go to step 1.
Exception Flow 2) Close the Window		**Actor Input**	**System Response**
	1		Ends the use case unsuccessfully.
Preconditions:		The actor is logged on his trip.	
Post-conditions:		The actor returns to his trip.	
Priority:		Medium	
Involved Class:		Trip, Touristic Resource, User	

Table 70.: Textual description of use case `Export trip to PDF format`

Use Case `Consult useful schedules`

Use Case ID:		CU02_20	
Use Case Name:		Consult useful schedules	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2014	
Actors:		Anonymous User, Authenticated User, Administrator	
Brief Description:		Allows users to consult several schedules about the events or point of interest in order to organize a better trip.	
Preconditions:		The actor has selected a touristic resource.	
		Actor Input	**System Response**
Flow of Events	1	Goes to section `Schedules`.	
	2		Visualize the schedules.
Exception Flow:		**Actor Input**	**System Response**
1) Actor close the window	2		Ends the use case unsuccessfully.
Exception Flow		**Actor Input**	**System Response**
1) Schedules are not available	1		Do not visualize the schedules.
Post-conditions:		The actor returns to the touristic resource.	
Priority:		Medium	
Involved Class:		Trip, Touristic Resource	

Table 71.: Textual description of use case `Consult useful schedules`

Use Case `Consult the weather`

Use Case ID:		CU02_21	
Use Case Name:		Consult the weather	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2014	
Actors:		Anonymous User, Authenticated User, Administrator	
Brief Description:		Allows users to consult the wheater about a place.	
Preconditions:		The actor has selected a touristic resource.	
		Actor Input	**System Response**
Flow of Events	1	Goes to section `The wheater`.	
	2		Visualize the wheather.
Exception Flow:		**Actor Input**	**System Response**
1) Actor close the window	1		Ends the use case unsuccessfully.
Exception Flow		**Actor Input**	**System Response**
1) The wheater are not available	1		Do not visualize the wheater.
Post-conditions:		The actor returns to the touristic resource.	
Priority:		Medium	
Involved Class:		Trip, Touristic Resource	

Table 72.: Textual description of use case `Consult the weather`

Use Case `Consult traffic bulletin`

Use Case ID:		CU02_22	
Use Case Name:		Consult traffic bulletin	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2014	
Actors:		Anonymous User, Authenticated User, Administrator	
Brief Description:		Allows users to consult the traffic bulletin about a place.	
Preconditions:		The actor has selected a touristic resource.	
		Actor Input	**System Response**
Flow of Events	1	Goes to section `Traffic Bulletin`.	
	2		Visualize the traffic bulletin.
Exception Flow: 1) Actor close the window		**Actor Input**	**System Response**
	1		Ends the use case unsuccessfully.
Exception Flow 1) The traffic Bulletin are not available		**Actor Input**	**System Response**
	1		Do not visualize the traffic Bulletin.
Post-conditions:		The actor returns to the touristic resource.	
Priority:		Medium	
Involved Class:		Trip, Touristic Resource	

Table 73.: Textual description of use case `Consult traffic bulletin`

Use Case `Receive the trip by email`

Use Case ID:		CU02_23	
Use Case Name:		Receive the trip by email	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2014	
Actors:		Authenticated User	
Brief Description:		Allows users to view their trips in their inboxes.	
Preconditions:		The actor is logged on his trip.	
		Actor Input	**System Response**
Flow of Events	1	Goes to section `Send me the trip`.	
	2		Displays a form with a field to introduce the email address.
	3	Introduces the email address	
	4	Click the link `Send me`	
	5		Validate the email address
	6		Send the trip by email to recipients
Alternative Flow: 3) Actor cancel		**Actor Input**	**System Response**
	1		Go to step 1.
Alternative Flow 5) Invalid email		**Actor Input**	**System Response**
	1		Displays an error message.
	2		Go to step 2.
Exception Flow: 1) Actor close the window		**Actor Input**	**System Response**
	1		Ends the use case unsuccessfully.
Post-conditions:		The trip is sent by email.	
Priority:		Medium	
Involved Class:		Trip, Touristic Resource, User	

Table 74.: Textual description of use case `Receive the trip by email`

Use Case `Download the trip to mobile application`

Use Case ID:		CU02_24	
Use Case Name:		Download the trip to mobile application	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2014	
Actors:		Authenticated User	
Brief Description:		In order to give a greater mobility to the system, it is allowed users to download their trips to their mobile phones.	
Preconditions:		The actor is logged inside the mobile application.	
		Actor Input	**System Response**
Flow of Events	1	Goes to section `Download trips`.	
	2		Displays a list of trips
	3	Selects the trips	
	4		Download the selected trips to the mobile application
	5		Displays a message saying that the trips has been downloaded
Alternative Flow: 3) Actor cancel		**Actor Input**	**System Response**
	1		Go to step 1.
Exception Flow: 3) Actor close the window		**Actor Input**	**System Response**
	1		Ends the use case unsuccessfully.
Post-conditions:		The trip is downloaded to the mobile application.	
Priority:		Medium	
Involved Class:		Trip, Touristic Resource, User	

Table 75.: Textual description of use case `Download the trip to mobile application`

Use Case `Delete the downloaded trip from mobile application`

Use Case ID:		CU02_25	
Use Case Name:		Delete the downloaded trip from mobile application	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2014	
Actors:		Authenticated User	
Brief Description:		Allows users to delete their downloaded trips from mobile application.	
Preconditions:		The actor is logged inside the mobile application.	
		Actor Input	**System Response**
Flow of Events	1	Goes to section `Remove trips`.	
	2		Displays a list of trips
	3	Selects the trips	
	4		Displays a confirmation message
	5	Confirm the deletion	
	6		Deletes the trips
	7		Displays a message saying that the trips has been deleted
Alternative Flow: 5) Actor cancel		**Actor Input**	**System Response**
	1		Go to step 1.
Exception Flow: 5) Actor close the window		**Actor Input**	**System Response**
	1		Ends the use case unsuccessfully.
Post-conditions:		The selected trips are deleted from the mobile application.	
Priority:		Medium	
Involved Class:		Trip, Touristic Resource, User	

Table 76.: Textual description of use case `Delete the downloaded trip from mobile application`

PACKAGE `social networks`

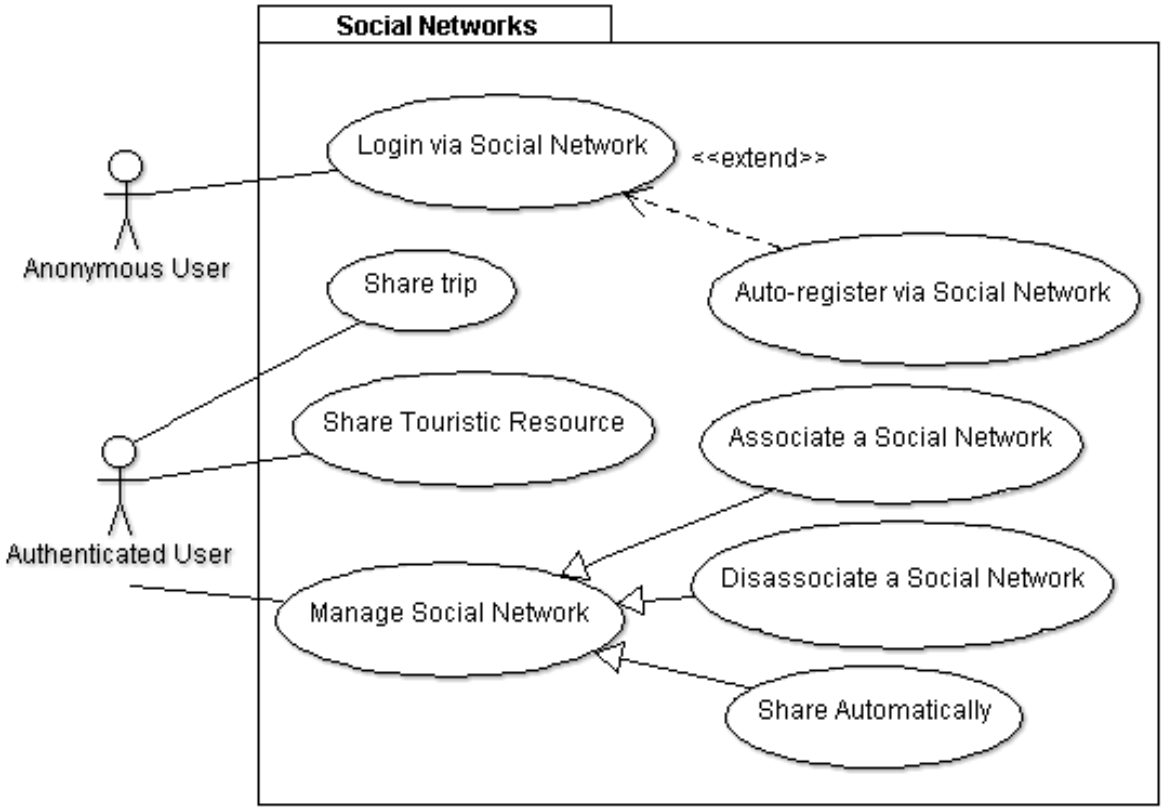


Figure 56.: Use case diagram for package `Social Networks`

Use Case `Login via Social Network`

Use Case ID:		CU03_01	
Use Case Name:		Login via Social Network	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2012	
Actors:		Anonymous User	
Brief Description:		Allows users to auto-register and/or login into the web portal or mobile application via their Social Network account like Facebook or Twitter.	
Preconditions:		Actor has an existing Social Network account.	
		Actor Input	**System Response**
Flow of Events	1	Choose login by Social Network to use the System.	
	2	Actor will be redirected to Social Network application for verification	
	3		Upon successful verification, actor is auto-registered.
	4		Extern Window Social Network re-directs to the home screen.
Alternative Flow: 3) Invalid User or Password		**Actor Input**	**System Response**
	1		Verification fails and actor will be prompted to re-enter Social Network username and password correctly.
Post-conditions:		The actor is authenticated and the system displays all features available for the role of the user.	
Priority:		High	
Involved Class:		User, Role, Session, Social Network	

Table 77.: Textual description of use case `Login via Social Network`

Use Case `Share Trip`

Use Case ID:		CU03_02	
Use Case Name:		Share Trip	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2012	
Actors:		Authenticated User, Administrator	
Brief Description:		Allows actor to share a trip via Social Network.	
Preconditions:		The actor is on a trip.	
		Actor Input	**System Response**
Flow of Events	1	Click the link `Share` from a social network.	
	2		The extern window Social Networks post the trip.
Post-conditions:		The trip was shared via social network.	
Priority:		Medium	
Involved Class:		Trip, Social Network	

Table 78.: Textual description of use case `Share Trip`

Use Case `Share Touristic Resource`

Use Case ID:		CU03_03	
Use Case Name:		Share Touristic Resource	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2012	
Actors:		Authenticated User, Administrator	
Brief Description:		Allows actor to share a touristic resource via Social Network.	
Preconditions:		The actor is on a touristic resource.	
		Actor Input	**System Response**
Flow of Events	1	Click the link `Share` from a social network.	
	2		The extern window Social Networks post the trip.
Post-conditions:		The touristic resource was shared via social network.	
Priority:		Medium	
Involved Class:		Touristic Resource, Social Network	

Table 79.: Textual description of use case `Share Touristic Resource`

Use Case `Associate a Social Network`

Use Case ID:		CU03_04	
Use Case Name:		Associate a Social Network	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2012	
Actors:		Authenticated User	
Brief Description:		Allows user to connect his social network account to web portal account.	
Preconditions:		The actor has web portal and social network accounts	
		Actor Input	**System Response**
Flow of Events	1	Navigate to his account.	
	2	Add a social network account.	
	3		Re-direct the actor for social network verification.
	4		Upon successful verification, page redirects to actor's account.
Alternative Flow: 4) Invalid verification		**Actor Input**	**System Response**
	1		When verification fails, user is prompted to re-enter his social network username and password correctly.
Post-conditions:		The social network account is associated with web portal account	
Priority:		Medium	
Involved Class:		Social Network	

Table 80.: Textual description of use case `Associate a Social Network`

Use Case `Dissociate a Social Network`

Use Case ID:		CU03_05	
Use Case Name:		Dissociate a Social Network	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2012	
Actors:		Authenticated User	
Brief Description:		Allows user to disconnect his social network account to web portal account.	
Preconditions:		User has web portal and social network accounts	
		Actor Input	**System Response**
Flow of Events	1	Navigate to his account.	
	2	Select the social network.	
	3	Presses the link `Quit`.	
	4		Disassociate the social network selected
Post-conditions:		The social network account is disassociated with web portal account.	
Priority:		Medium	
Involved Class:		Social Network	

Table 81.: Textual description of use case `Dissociate a Social Network`

Use Case `Share Automatically`

Use Case ID:		CU03_06	
Use Case Name:		Share Automatically	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2012	
Actors:		Authenticated User, Administrator	
Brief Description:		This use case describes the process of sharing automatically a trip or touristic resource via social network.	
Preconditions:		User has web portal account and at least a social network account.	
		Actor Input	**System Response**
Flow of Events	1	Presses the link `Setting Sharing`.	
	2		Display a list of social networks.
	3	Select a social network.	
	4	Presses the link `Share Automatically`.	
	5		The setting is uploaded into the database.
Post-conditions:		When the authenticated user creates a trip or the administrator creates a touristic resource the post is automatically shared.	
Priority:		High	
Involved Class:		Trip, Touristic Resource	

Table 82.: Textual description of use case `Share Automatically`

PACKAGE searching

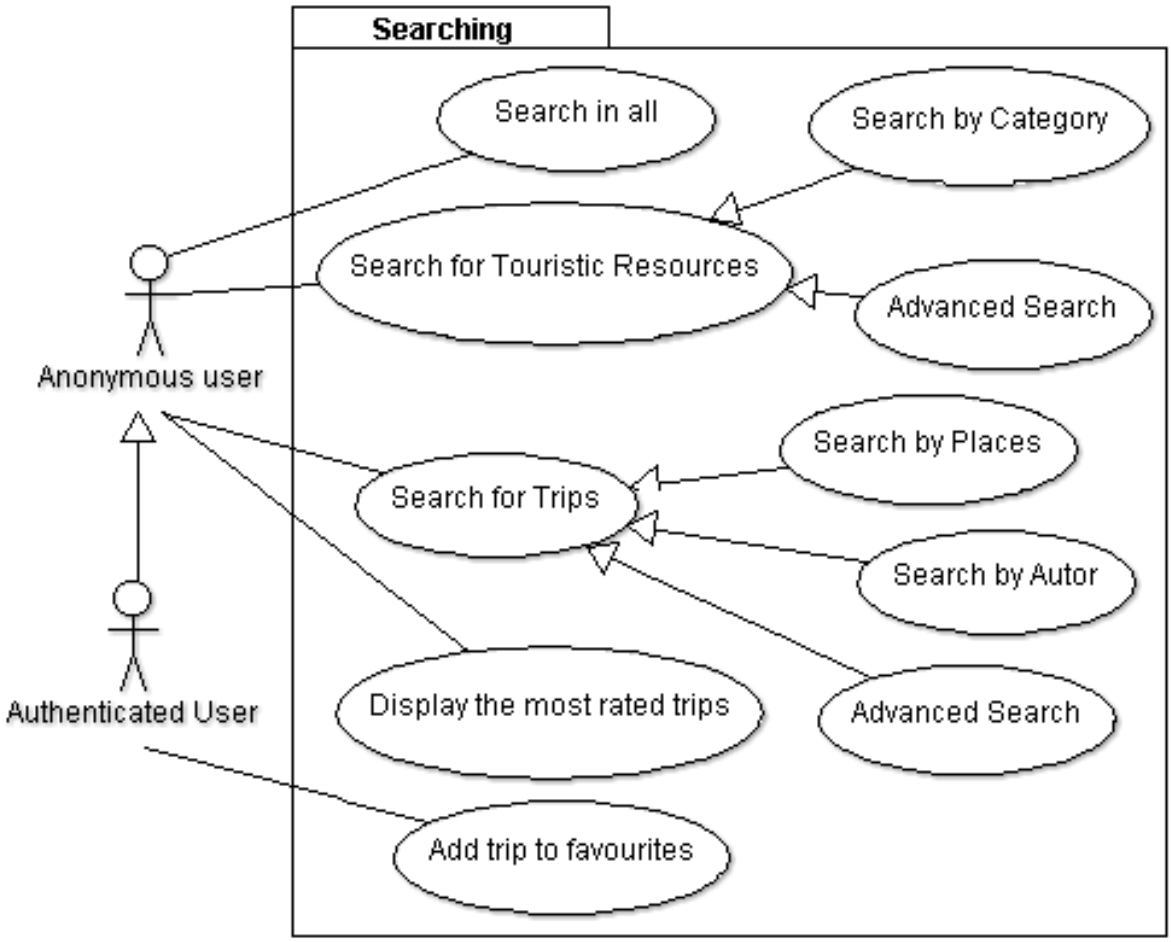


Figure 57.: Use case diagram for package Searching

Use Case `Advanced Search`

Use Case ID:		CU04_01	
Use Case Name:		Advanced Search	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2012	
Actors:		Anonymous User, Authenticated User, Administrator	
Brief Description:		An advanced method to search for items available for purchase. This use case returns a response with links within their own website.	
Preconditions:		The web portal must be online and available.	
		Actor Input	**System Response**
Flow of Events	1	Write keywords or any text into the searchbox.	
	2	Select the features of searching (trip, touristic resources, autors, etc).	
	3		Search for the request text.
	4		Show results.
Alternative Flow: 3) The introduced text is a city		**Actor Input**	**System Response**
	1		Show a link to the city guide at top.
	2		Go to step 4.
Exception Flow 3) Results not found		**Actor Input**	**System Response**
	1		Show the message `Results not found`.
Post-conditions:		The actor resturns to search box.	
Priority:		Medium	
Involved Class:		Trip, Touristic Resource, User	

Table 83.: Textual description of use case `Advanced Search`

Use Case `Search in all`

Use Case ID:		CU04_02	
Use Case Name:		Search in all	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2012	
Actors:		Anonymous User, Authenticated User, Administrator	
Brief Description:		A method to search in all the web portal for items available for purchase. This use case returns a response with links within their own website.	
Preconditions:		The web portal must be online and available	
		Actor Input	**System Response**
Flow of Events	1	Write keywords or any text into the searchbox.	
	2		Search for the request text.
	3		Show results.
Alternative Flow: 2) The introduced text is a city		**Actor Input**	**System Response**
	1		Show a link to the city guide at top.
	2		Go to step 3.
Exception Flow 2) Results not found		**Actor Input**	**System Response**
	1		Show the message `Results not found`.
Post-conditions:		The actor resturns to search box.	
Priority:		Medium	
Involved Class:		Trip, Touristic Resource, User	

Table 84.: Textual description of use case `Search in all`

Use Case `Search by Category`

Use Case ID:		CU04_03	
Use Case Name:		Search by Category	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2012	
Actors:		Anonymous User, Authenticated User, Administrator	
Brief Description:		Search for Touristic Resources by Category. This use case returns a response with links within their own website. The category can be a kind of place like: attraction, entertenamient, ect.	
Preconditions:		The web portal must be online and available.	
Post-conditions:		The actor resturns to search box.	
		Actor Input	**System Response**
Flow of Events	1	Write keywords or any text into the searchbox.	
	2	Select the category.	
	3		Search for the request text.
	4		Show results.
Exception Flow 2) Results not found		**Actor Input**	**System Response**
	1		Show the message `Results not found`.
Priority:		Medium	
Involved Class:		Touristic Resource	

Table 85.: Textual description of use case `Search by Category`

Use Case `Search by Places`

Use Case ID:		CU04_04	
Use Case Name:		Search by Places	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2012	
Actors:		Anonymous User, Authenticated User, Administrator	
Brief Description:		Search for trips by places. This use case returns a response with links within their own website.	
Preconditions:		The web portal must be online and available.	
		Actor Input	**System Response**
Flow of Events	1	Write keywords or any text into the searchbox.	
	2		Search for the request text.
	3		Show results.
Exception Flow 2) Results not found		**Actor Input**	**System Response**
	1		Show the message `Results not found`.
Post-conditions:		The actor resturns to search box.	
Priority:		Medium	
Involved Class:		Trip, Touristic Resource	

Table 86.: Textual description of use case `Search by Places`

Use Case `Search by Autor`

Use Case ID:		CU04_05	
Use Case Name:		Search by Autor	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2012	
Actors:		Anonymous User, Authenticated User, Administrator	
Brief Description:		Search for trips by Autor. This use case returns a response with links within their own website.	
Preconditions:		The web portal must be online and available.	
		Actor Input	**System Response**
Flow of Events	1	Presses the link of the autor of any trip.	
	2		Search for the request text.
	3		Show results.
Exception Flow 2) Results not found		**Actor Input**	**System Response**
	1		Show the message `Results not found`.
Post-conditions:		The actor resturns to search box.	
Priority:		Medium	
Involved Class:		Trip, User	

Table 87.: Textual description of use case `Search by Autor`

Use Case `Display the most rated trips`

Use Case ID:		CU04_06	
Use Case Name:		Display the most rated trips	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2014	
Actors:		Anonymous User, Authenticated User, Administrator	
Brief Description:		Allows users to display the most rated trips.	
Preconditions:		The web portal must be online and available for use.	
		Actor Input	**System Response**
Flow of Events	1	Goes to section `The popular trips`.	
	2		Visualize the most rated trips.
Exception Flow:		**Actor Input**	**System Response**
1) Actor close the window	1		Ends the use case unsuccessfully.
Post-conditions:		The actor returns to the home page.	
Priority:		Medium	
Involved Class:		Trip, Touristic Resource	

Table 88.: Textual description of use case `Display the most rated trips`

Use Case `Add trip to favourites`

Use Case ID:		CU04_07	
Use Case Name:		Add trip to favourites	
Author:		Miguel Angel Lopez Mamani	
Date:		March 8, 2014	
Actors:		Authenticated User	
Brief Description:		Allows users to add their trips to the favourites section.	
Preconditions:		The actor is logged on his trip.	
		Actor Input	**System Response**
Flow of Events	1	Presses the link `Add to my favourites`.	
	2		Add the trip to the favourites section.
Exception Flow: 1) Actor close the window		**Actor Input**	**System Response**
	1		Ends the use case unsuccessfully.
Post-conditions:		The trip was added to the favourites section.	
Priority:		Low	
Involved Class:		Trip, Touristic Resource	

Table 89.: Textual description of use case `Add trip to favourites`

B.3.3 *Information Requirements*

It will use external information about points of interest from a web mapping service application like Google maps, Yahoo maps, or Open Street Maps. It will use external information about the weather of particular points of interest through dates and geographical coordinates.

C

IMPLEMENTATION SOURCE CODE

C.1 PHP FUNCTION FOR CREATING A TRIP

```
function trips_insert($node, $form_state) {
  $listpois = array();
  $listpois = explode(",", $_POST['listpois']);
  $timestampa = date('Y-m-d G:i:s.BO');

  $idtrip = db_insert('wptsc_trips')
    ->fields(array(
        'trip_id' => $node->nid,
        'name' => $node->t_name,
        'inicio' =>  substr($node->field_dates['und'][0]['value'],0,10),
        'fin' =>  substr($node->field_dates['und'][0]['value2'],0,10),
        'public' => $node->t_publico,
        'count_likes' => 0,
        'count_dislikes' => 0,
        'created_on' => $timestampa,
        'updated_on' => $timestampa,
    ))
  ->execute();

  foreach($listpois as $th){
    db_insert('wptsc_trips_resources')
      ->fields(array(
          'trip_id' => $node->nid,
          'resource_id' => $th,
          'daydata' => '-1',
          'category' => 'C',
        ))
      ->execute();
  }
}
```

Source Code C.1: PHP function for creating a trip

C.2 PHP FUNCTION FOR CREATING A TOURISTIC RESOURCE

```
function touristic_resources_insert($node, $form_state) {
  $timestampa = date('Y-m-d G:i:s.BO');
  $id_rt = $node->nid;
  $name_rt = $node->rt_name;
  $category_rt = $node->rt_category;
  $brief_description_rt = $node->rt_brief_description;
  $details_rt = $node->rt_details;
  $email_rt = $node->rt_email;
  $site_rt = $node->rt_site;
  $phone_rt = $node->rt_phone;
  $location_rt = '('.$node->rt_location_x.' '.$node->rt_location_y.')';
  $full_address_rt = $node->rt_full_address;

  db_query('INSERT INTO wptsc_resources (id, count_likes, count_dislikes,
      created_on, updated_on, type) VALUES (:arg1, :arg2, :arg3, :arg4, :
     arg5, :arg6)', array(':arg1' => $node->nid, ':arg2' => 0, ':arg3' =>
      0, ':arg4' => $timestampa, ':arg5' => $timestampa, ':arg6' =>
     $category_rt));

  db_query('INSERT INTO wptsc_resource_infos(resource_id, language_id, "
     name", brief_description, details,created_on, updated_on) VALUES (:
     arg1, :arg2, :arg3, :arg4, :arg5, :arg6, :arg7)', array(':arg1' =>
     $id_rt, ':arg2' => 'en', ':arg3' => $name_rt, ':arg4' =>
     $brief_description_rt, ':arg5' => $details_rt, ':arg6' =>
     $timestampa, ':arg7' => $timestampa));

  if($category_rt==1){
    db_query('INSERT INTO wptsc_pois(resource_ptr_id, address, email,
       site, phone, "location") VALUES ('.$id_rt.', \''.$full_address_rt.
       '\', \''.$email_rt.'\', \''.$site_rt.'\', \''.$phone_rt.'\',
       GeometryFromText(\'POINT'.$location_rt.'\',4326))');
  }elseif($category_rt==2){
    $subcategory_rt = $node->rt_subcategory;
    db_query('INSERT INTO wptsc_services(resource_ptr_id, address, email,
       site, phone, service_type, "location") VALUES ('.$id_rt.', \''.
       $full_address_rt.'\', \''.$email_rt.'\', \''.$site_rt.'\', \''.
```

```
        $phone_rt.'\', '.$subcategory_rt.'  ,GeometryFromText(\'POINT'.
        $location_rt.'\',4326))');
    insert_subcategory_touristic_resources($node, $subcategory_rt);
  }
}
```

Source Code C.2: PHP function for creating a touristic resource

C.3 PHP FUNCTION FOR UPDATING A TOURISTIC RESOURCE

```
function touristic_resources_update($node) {
  $timestampa = date('Y-m-d G:i:s.BO');
  $id_rt = $node->nid;
  $name_rt = $node->rt_name;
  $category_rt = $node->rt_category;
  $brief_description_rt = $node->rt_brief_description;
  $details_rt = $node->rt_details;
  $email_rt = $node->rt_email;
  $site_rt = $node->rt_site;
  $phone_rt = $node->rt_phone;
  $location_rt = '('.$node->rt_location_x.' '.$node->rt_location_y.')';
  $full_address_rt = $node->rt_full_address;

  db_update('wptsc_resources')
    ->fields(array(
        'updated_on' => $timestampa,
        'type' => $category_rt,
      ))
    ->condition('id', $id_rt)
    ->execute();

  db_update('wptsc_resource_infos')
    ->fields(array(
        'name' => $name_rt,
        'brief_description' => $brief_description_rt,
        'details' => $details_rt,
        'updated_on' => $timestampa,
    ))
    ->condition('resource_id', $id_rt)
    ->execute();

  if($category_rt==1){
```

```
    $result = db_query('SELECT * FROM wptsc_pois WHERE resource_ptr_id='.
        $id_rt);
    if($result->rowCount()>0){//the same type
      db_query('UPDATE wptsc_pois SET address=\''.$full_address_rt.'\',
          email=\''.$email_rt.'\', site=\''.$site_rt.'\', phone=\''.
          $phone_rt.'\', "location"=GeometryFromText(\'POINT'.$location_rt
          .'\',4326) WHERE resource_ptr_id='.$id_rt);
    }else{
      db_query('INSERT INTO wptsc_pois(resource_ptr_id, address, email,
          site, phone, "location") VALUES ('.$id_rt.', \''.
          $full_address_rt.'\', \''.$email_rt.'\', \''.$site_rt.'\', \''.
          $phone_rt.'\', GeometryFromText(\'POINT'.$location_rt.'\',4326))
          ');
    }
  }elseif($category_rt==2){
    $subcategory_rt = $node->rt_subcategory;
    $result = db_query('SELECT * FROM wptsc_services WHERE
        resource_ptr_id='.$id_rt);
    if($result->rowCount()>0){//the same type
      db_query('UPDATE wptsc_services SET address=\''.$full_address_rt.'
          \', email=\''.$email_rt.'\', site=\''.$site_rt.'\', phone=\''.
          $phone_rt.'\', service_type='.$subcategory_rt.', "location"=
          GeometryFromText(\'POINT'.$location_rt.'\',4326) WHERE
          resource_ptr_id='.$id_rt);
    }else{
      db_query('INSERT INTO wptsc_services(resource_ptr_id, address,
          email, site, phone, service_type, "location") VALUES ('.$id_rt.'
          , \''.$full_address_rt.'\', \''.$email_rt.'\', \''.$site_rt.'\',
           \''.$phone_rt.'\', '.$subcategory_rt.'  ,GeometryFromText(\'
          POINT'.$location_rt.'\',4326))');
      db_delete('wptsc_pois')->condition('resource_ptr_id', $id_rt)->
          execute();
    }
    update_subcategory_touristic_resources($node, $subcategory_rt)
  }
}
```

Source Code C.3: PHP function for updating a touristic resource

C.4 PHP FUNCTION TO PRINT THE LIST OF TRIPS

```
function trip_list_page($form, &$form_state) {
```

```
  global $user;
  $query = db_select('wptsc_trips', 't')->extend('PagerDefault');
  $query->join('node', 'n', 'n.nid = t.trip_id');
  $query->fields('t', array('trip_id','name', 'inicio', 'fin'));
  $query->condition('n.uid', $user->uid);

  $result = $query
    ->limit(6)
    ->orderBy('t.inicio','DESC')
    ->execute();

  $output='<h1 class="title" id="page-title">Trips</h1><br />';
  if($result->rowCount()<1){
  $output.='You do not have trips';
  }

  foreach ($result as $row) {
  $coordinates='';
  $stacke = array();
  $places = db_query('SELECT ri."name", x(p."location"), y(p."location")
     FROM wptsc_pois AS p, wptsc_trips_resources AS tr,
     wptsc_resource_infos AS ri WHERE p.resource_ptr_id = tr.resource_id
     AND ri.resource_id=p.resource_ptr_id AND tr.trip_id='.$row->trip_id)
     ;
  foreach ($places as $place) {
    $coordinates.=('|'.$place->x.','.$place->y);
    array_push($stacke, $place->name);
  }
  $output.='<img width="145" height="97" alt="'.$row->name.'" src="http
     ://maps.google.com/maps/api/staticmap?size=300x200&maptype=terrain&
     sensor=false&markers=size:small';
  $output.=$coordinates.'&path=color:0x0000ff|weight:5'.$coordinates.'"
     style="float:left">';
  $output.='<div style="float:left"><a href="'.url("node/$row->trip_id").
     '"><h3> '.$row->name.'</h3></a>';
  $output.='</div><div style="clear:both"></div><br />';
  }

  $form['contenido'] = array('#markup' => $output);
  $form['pager'] = array('#theme' => 'pager');
  return $form;
}
```

Source Code C.4: PHP function for listing a trip

C.5 JAVA METHOD FOR ADDING A TRIP

```
private int addTrip(int trip_id, String title_group, String title_item,
    String dateTrip){
  int groupPosition = 0;
  TripInfoHeader tripInfoHeader = myGroups.get(title_group);
  if(tripInfoHeader == null){
    tripInfoHeader = new TripInfoHeader();
    tripInfoHeader.setName(title_group);
    myGroups.put(title_group, tripInfoHeader);
    deptList.add(tripInfoHeader);
  }

  ArrayList<TripInfo> productList = tripInfoHeader.getProductList();
  int listSize = productList.size();
  listSize++;

  TripInfo tripInfo = new TripInfo();
  tripInfo.setId(trip_id);
  tripInfo.setTitle_group(title_group);
  tripInfo.setTitle_item(title_item);
  tripInfo.setFecha(dateTrip);
  productList.add(tripInfo);
  tripInfoHeader.setProductList(productList);

  groupPosition = deptList.indexOf(tripInfoHeader);
  return groupPosition;
}
```

Source Code C.5: Java method for adding a trip

C.6 XML LAYOUT TO PRINT THE LIST OF TOURISTIC RESOURCES PER DAY

```
<?xml version="1.0" encoding="utf-8"?>
<TabHost xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/tabhost"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <LinearLayout
        android:id="@+id/linearLayout1"
        android:layout_width="fill_parent"
```

```
        android:layout_height="fill_parent"
        android:orientation="vertical" >
        <TabWidget
            android:id="@android:id/tabs"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" >
        </TabWidget>
        <FrameLayout
            android:id="@android:id/tabcontent"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent" >
            <LinearLayout
                android:id="@+id/tab1"
                android:layout_width="match_parent"
                android:layout_height="match_parent" >
                <ExpandableListView
                    android:id="@+id/myListdays"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:layout_weight="1" >
                </ExpandableListView>
            </LinearLayout>
            <LinearLayout
                android:id="@+id/tab2"
                android:layout_width="match_parent"
                android:layout_height="match_parent" >
            </LinearLayout>
        </FrameLayout>
    </LinearLayout>
</TabHost>
```

Source Code C.6: XML layout to print the list of touristic resources per day