FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Interactive Spider Maps for Public Transportation

Francisco Miguel Amaro Maciel

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Maria Teresa Galvão Dias

July 18, 2016

Interactive Spider Maps for Public Transportation

Francisco Miguel Amaro Maciel

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Carla Alexandra Teixeira Lopes
External Examiner: Alberto Manuel Brandão Simões
Supervisor: Maria Teresa Galvão Dias
July 18, 2016

Abstract

In Public Transportation systems, accurate representation of information has been key for users to take the more advantages of the services and fulfil their travel needs. The Spider Map is a particular schematic representation to illustrate all travel possibilities from a given geographical location, stretching out in different directions without disregarding their geographic constraints, but simplifying the information be more accessible to the user. Recently the Spider Map interaction and generation process automation has been studied, although there are yet different possibilities to explore in the technological approach and interactive solution fields. This research proposes the Cobweb solution which defines the components of an Interactive Spider Map, focusing on the interactive dynamic potential that a custom generation has access to, challenging the existing standard representation. This adaptation also relaxes the constraints of the traditional approach, defining the interaction processes required and information representation alternatives. An implementation of the solution is developed, named CobWeb 1.0 as an interactive web application for mobile touch devices, covering the Public Transportation services of the city of Porto, generating spider maps for any location within its area. The result of this development expands and tests different features of the Cobweb, focusing on transforming user interaction in valuable information. This application is evaluated using carefully designed user tests to validate the design decisions, taking into different interaction alternatives for each phase of the map generation. The results show improvements regarding the traditional alternative, with positive user response, valuing highly context awareness features.

Resumo

Em sistemas de Transportes Públicos, a representação precisa da informação é um ponto chave no auxílio ao cliente na capacidade de decisão para tirar o maior partido do serviço, procurando satisfazer as suas necessidades de deslocação. Os "Spider Maps" são um tipo particular de representação esquemática que ilustra todas a possibilidades de viagem a partir de uma determinada localização geográfica, espalhando-se em diferentes direcções sem perder as restrições do contexto geográfico, mas simplificando a acessibilidade de informação para o utilizador. Recentemente a interacção com os "Spider Maps" e o processo de geração automatizado foram aprofundados, embora existam ainda diversas possibilidades para explorar do ponto de vista tecnológico e do ponto de vista da interacção. Este trabalho de investigação propõe a solução Cobweb, dissecando os componentes de um "Spider Map" interactivo, focando-se no potencial dinâmico que a geração de um mapa em tempo real tem acesso, e as diferentes formas de representação de informação, desafiando as convenções de representação existentes, relaxando as restrições da abordagem tradicional e definindo o processo de interacção necessário e as alternativas para a representação de informação. Foi desenvolvida uma implementação da solução, denominada CobWeb 1.0 como uma aplicação web interactiva para dispositivos móveis que suportam touch, abrangendo a rede de Transportes Públicos da cidade do Porto e permitindo a geração de "Spider Maps" em toda a região. O resultado deste processo de desenvolvimento expande e testa diferentes funcionalidades do Cobweb, focando-se na transformação de interacção em informação de qualidade. A aplicação é avaliada utilizando testes com o utilizado preparados cuidadosamente para validar as decisões de desenho, tendo em consideração diferentes alternativas de interação para cada fase da geração do mapa. Os resultados mostram uma melhoria em comparação com a alternativa tradicional, recebendo feedback positivo dos utilizadores, valorizando funcionalidades que permitam aumentar a interligação com o contexto.

Acknowledgements

To my supervisor Professor Teresa Galvão, for all the guidance which allowed me turn an idea into reality, I am deeply thankful. I want to thank the Department of Informatics Engineering for aid in my daily and technological needs. I would like to thank OPT[1] for collaborating with this project and granting me the tools to do so. Thank you to all who agreed to participate in the testing process, for your time and dedication to the benefit of this project.

To my colleagues and friends, Hugo Sousa, Marie Ros and Daniel Moreira, for the kinship while climbing the same mountain while having the ability to turn it into a straight road. Many thanks to all the friends who have been with me on some way, and a special thank you to all the cheerleaders back in the loft, for tirelessly believing in me. To the ones who cared for my health and kept me sane, thank you.

To my parents, for the unending support and encouragement, and to my brother for the never-ending companionship. To my grandfather, for bestowing me with the way to admire all the wonders in this world.

And most of all to you, without whom neither I nor this words would be here today. Thank you from the bottom of my heart Mariana, for all the love and support throughout the this adventure, on all the highs and lows. Thank you, for flipping the lows into highs, and for celebrating the highs higher.

Francisco Maciel

[1] OPT - Optimização e Planeamento de Transportes is a company that is located in Porto, developing software for transportation purposes. www.opt.pt

"All that is gold does not glitter"

J. R. R. Tolkien

Contents

List of Figures

List of Tables

Abbreviations

GIS	Geographic Information System
API	Application Programming Interface
GPS	Global Positioning System
OPT	Optimizacão e Planeamento de Transportes
HCI	Human-Computer Interaction
HCD	Human-Centered Design
GPS	Global Positioning System
STCP	Sociedade de Transportes Colectivos do Porto
POI	Point of Interest
LBS	Location Based Services
GTFS	General Transit Feed Specification
USA	United States of America
CSV	Comma Separated Values
XML	Extensible Markup Language
PT	Public Transportation
REST	Representational State Transfer
HTML	Hypertext Markup Language
JSON	JavaScript Object Notation
WCF	Windows Communication Foundation
ETG	Empresa de Transportes Gondomarense
CP	Comboios de Portugal
HTTP	Hypertext Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
UI	User Interface

Chapter 1

Introduction

This chapter describes the research's primary topic, presenting its context while identifying the major work goals. It also details the research methodology and specifies the structure of the following chapters.

1.1 Context

Taking into account growing concerns on the chaotic traffic averages in highly populated cities and the pollution impact caused by the exacerbated use of personal vehicles[NGJ+00], Public Transport alternatives ought to be encouraged as much as possible in the future. Striving to provide cities and countries with a more usable and efficient substitute to personal transportation, not only through an increased number of physical infrastructures are necessary, but also a "partnership" with technology to play an import part in ensuring a maximization of the usability and ease of the platforms, as wells as accurate information access to every passenger. Maps have been used to portray the transportation networks, translating into mental projections of available services and navigation possibilities to their reader.

A particular subset of maps, schematic maps, are fairly popular in their attempt to simplify the networks, scaling, removing and adapting geographic elements of the real map, resulting in an information focused map, heavily relaxing the accuracy of reality representation constraints. Spider Maps' goal is to aid the user answering the question "From this area, where can I go?", mixing both elements of geographic maps (current area of location of the user) and schematic maps (possible destinations from this area) in a spider-like shape, as seen in figure 1.1. These maps are usually manually produced and then printed at high resolutions and large sizes for relevant points of interest, presenting a considerable amount of available destinations. Recently, the generation process has been studied in order to automate the design of the schematic, hastening the whole procedure [Mou15].

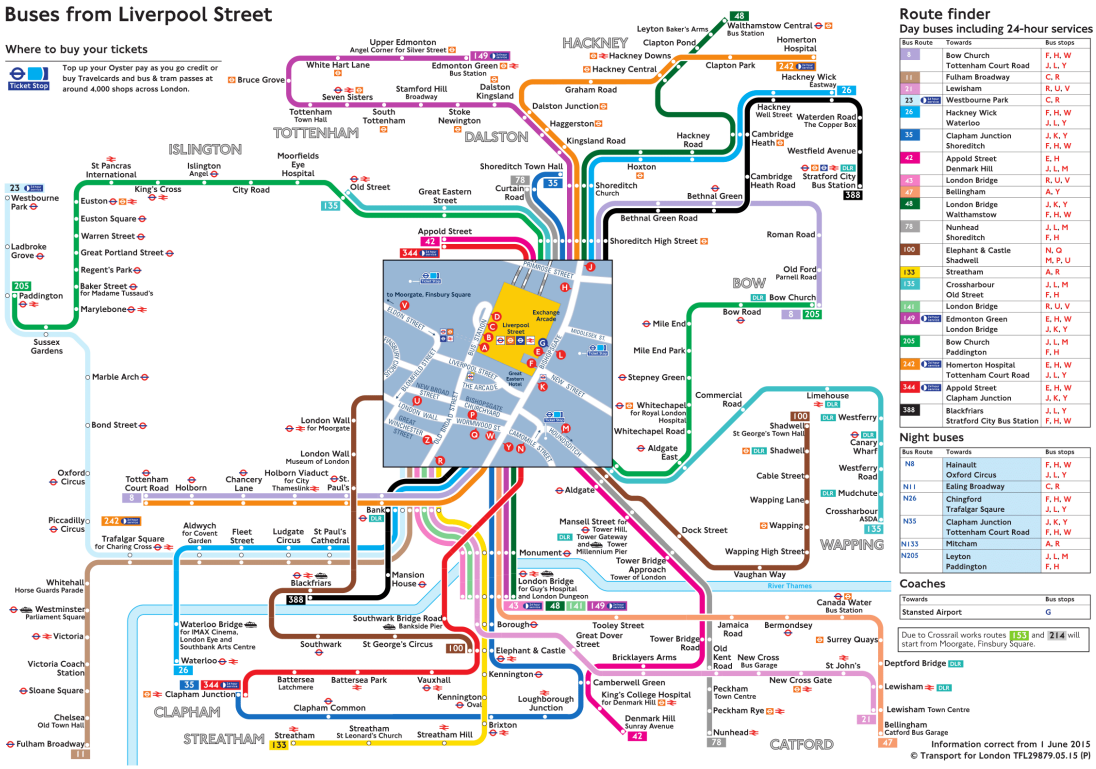


Figure 1.1: Liverpool Street Spider Map [fL15]

1.2 Motivation and Goals

The process of creating a Spider Map is not yet fully automated. Although the process of plotting the schematic lines on the map respecting the physical and representative restrictions has been studied, the result of this approach serves mainly, in most cases, as a basis for a designer's polish work. The resulting map also lacks the geographical section of the map, integrated through manual insertion, often contrasting its classic interface with the nearby schematic.

We aim to design an interactive Spider map solution, to be further implemented into an application to generate a Spider Map for any location in the city of Porto, connected to its public transportation system. This generation challenges the traditional static Spider Map by exploring the possible interaction potential in a dynamic platform, supplying a touch interface. Human-Computer Interaction is crucial in determining the success of this representation [Tor13]. The full automatic integration also opens new possibilities for defining user interface elements, motivating the exploration of the connection that binds the geographic maps and the schematic lines. The ultimate goal of the research is to study how the technological adaptation improves or reduces the informative potential of the spider map, and how can the dynamism and the interaction in a mobile platform improve it.

1.3 Methodology

In this research, the analysis of the problem begins my studying the state of the art regarding the concepts of interaction and maps, further deepened into the specific concepts involved. Furthermore, the results of this study allow the proposal of an interactive Spider Map solution, keeping the default interaction flow as a close approximation possible to the current Spider Maps, and further applying pressure on its boundaries, in an attempt to integrate new improvements.

The designed solution will draw the initial guideline to lead the following interface decisions and stimulate the actual implementation of such an application. The development process should consider the upcoming testing and evaluation, as Human-computer Interaction plays an important part in deciding the relevant alternatives identified. This process culminates in the planning of user tests and their execution, following a user-centered design [PRS02], evaluating success by at least comparing the information of the public transport network in the application matched against the traditional method. The results are then interpreted and conclusions are drawn from the overall research process.

1.4 Document Structure

This document content is structured in 6 sections, outlining the whole research process, starting by introducing it on the current chapter.

In chapter 2 the core concepts linked to "Mobile Interactive Maps for Public Transportation" are analysed exclusively as well intertwined with each other pairs, to widen the approach to the problem.

In chapter 3 the main structure and architecture concepts for the proposed solution are explained, discussing the considered alternatives and decisions.

In chapter 4 the implementation of the Interactive Spider Map is suggested, describing architecture and technology approach explaining the features developed and considerations taken into account.

Chapter 5 describes the preparation of the User Tests, further execution and analyses the follow-up results.

Chapter 6 sums up the research project, explaining the main conclusions from all the previous chapters and suggesting the future work.

Chapter 2

Interactive Spider Maps in Public Transportation

2.1 Introduction

For analysing the State of the Art, the main research topics are Interaction, Public Transportation, Maps and Mobile Devices. In this chapter we review how they connect with each other and how this bond can be strengthened, aiming to clarify the basis for this work.

2.2 Map Representations

2.2.1 Evolution of Map Portrayal through Technology

Maps have been used as a mean of storing and communicating spatial information throughout the past. Maps fall into different classifications and categories, being able to apply to geographic representations in different manners, classified as topographic maps, differing from thematic maps, which focus on symbolic contexts [Mou15].

Information systems have allowed for more precise mapping alternatives through the more advanced systems such as satellite images, as well as defining Geographic Information Systems relying in geospatial information and GPS positioning. This services have been made available to public and are highly explored in both commercial and personal interests.

2.2.2 Geographic and Schematic Maps

Maps can be classified differently according to the way they represent information, and *"combine pictorial representation with symbolic representation"*[BRB+97], a concept which is exemplified in figure 2.1. In the light of this definition, schematic maps attempt to transform the geographical spatial information used for other maps, such as topographic, into conceptual aspects which can be defined as a spatial relationship.

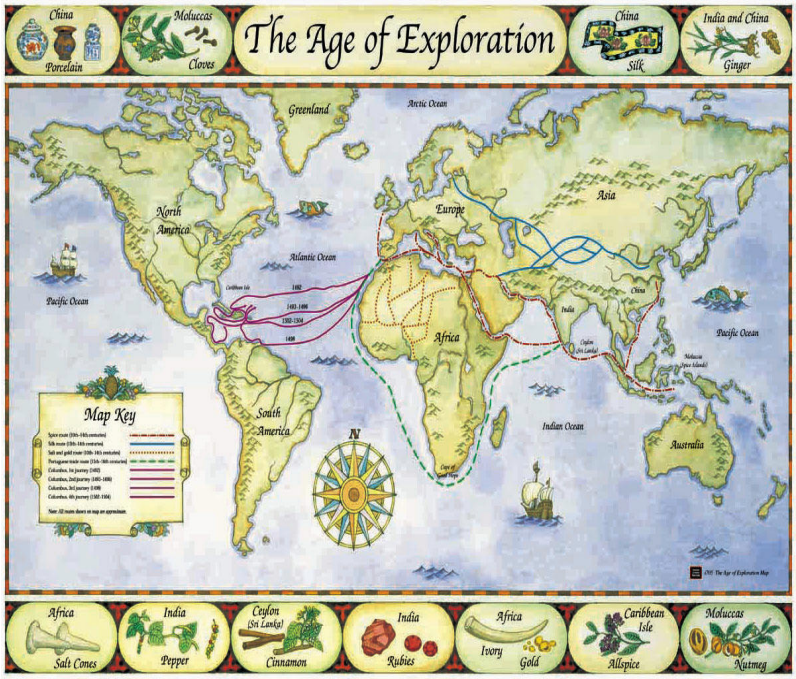


Figure 2.1: The Age of Exploration Map [map05]

In Public Transportation, schematic map relationships typically represent stops and routes, having the spatial relationship between stops a distorted reality of the geographic context, usually to represent information in a more readable way. Mixed approaches between readability and spatial context are often attempted, causing superfluous geographic constraints to relaxed. The resulting grid often differs highly from the original[Smi15b], as the physical map is stretched, as seen in figure 2.2

2.2.3 Time Representation in Geographic Information Systems

Besides adding spatial information, dynamic maps make possible different representations concerning displaying space-time variance, using diversified design strategies[Mon90]. The solution for this problem can be approached in different manners, such as additional interface diagrams or video transformations. The additional dimension representation challenge can also be statically introduced, to be transform a map differently, as seen in figure 2.3 where Sugiura designs Japan as the time it takes to travel using train from Tokyo, illustrating the country's transportation services' deficiencies.

2.3 Public Transportation

Public Transportation services are generally described by services that provide means for a passenger to travel using a shared way of transportation, available to the public. Examples of Public Transportation types such as urban buses, rapid buses, light rails, ferries and trains are widely

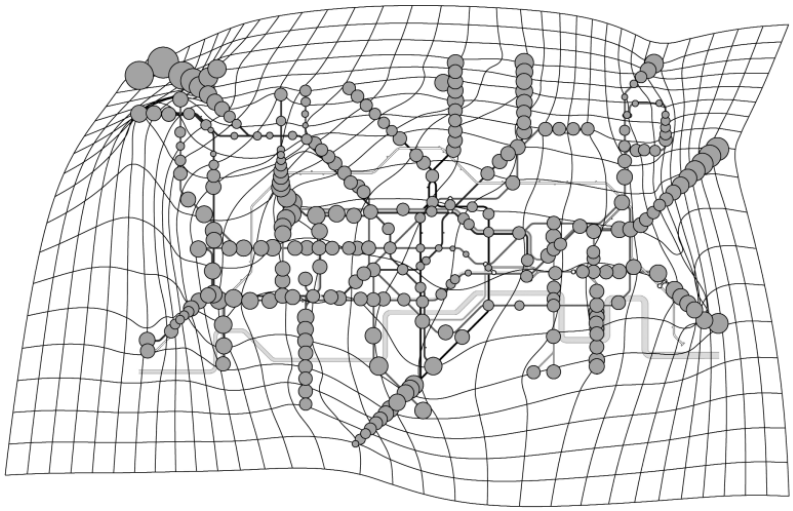


Figure 2.2: The current London Underground diagram with an overlaid distortion grid and displacement circles. The circles' areas are proportional to the distances to the correct locations. [Jen06]

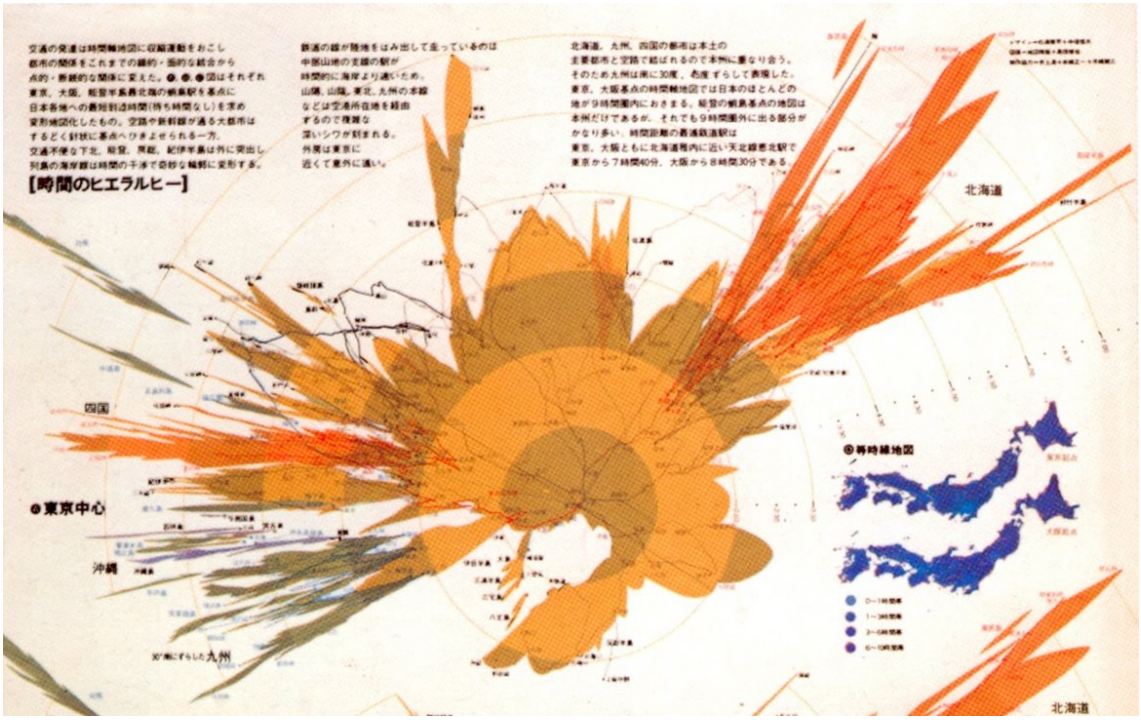


Figure 2.3: Travel-Time map representing travel time distance from Tokyo [Sug69]

spread and adopted around the world. Technology has changed both the ways these vehicles work and operate, as well as how passenger interact with the provided service.

2.3.1 Network Representation

Throughout different information systems, several similar concepts are used to represent information in Public Transportation networks. The standard approach reflects the similarity on the organization of most transport providers, following themselves similar needs. In an effort to make Public Transportation information available and shareable with other applications, common representation standards have been proposed.

The TRANSMODEL [tra97] data model has once been widely used, mostly in Europe. It is generally considered outdated, but might be in used by older system. It mainly concerns the following domains:

- Scheduling
- Passenger Information
- Automatic Vehicle Monitoring
- Personnel Disposition
- Management Information & Statistics

Currently most system adopt *Google*'s General Transit Feed Specification [Goo15] as a way to publish data in a machine readable and understandable manner. This data is supplied as CSV, in text files, including both Public Transportation and geographic information. The table files representing different facets of the service standardized are the following:

- Agency
- Routes
- Trips
- Stop_Times
- Stops
- Calendar

Although a GTFS adaptation also supports real time information, SIRI (Service Interface for Real Time Information[SIR15]) is also a XML protocol mostly used in Europe to exchange real time information. Although it is based on TRANSMODEL, this protocol takes into account not only information about the predicted data information but also the current and up to date status on the transportation network. SIRI comprises the following services:

- Production Timetable Service
- Estimated Timetable Service
- Stop Services (Stop Timetable and Stop Monitoring)
- Vehicle Monitoring Service
- Connection Protection Services (Connection Timetable and Connection Monitoring)
- General Messaging Service

2.3.2 Passenger Information

In a public transportation journey, there are different stages where information access can be evaluated according to how it reaches the user and how it is clear and understandable. This information can be measured in an End to End route provider service, according to the gap existing between what exists at the moment and what is the ideal stage, for each of the steps described in 2.4.

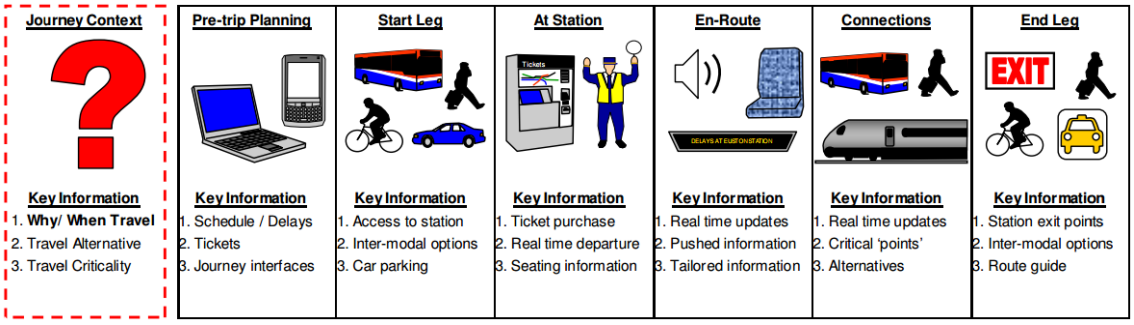


Figure 2.4: End to End Journey Stages [Joh10]

A typical information system architecture should take into account the importance of information to the passenger, namely in the stages that involve direct contact with the traveller supported by the transportation service. However, with the increasing technology advances it is important to reach out to the user even before the trip, allowing for knowledge concerning the service. As such, information access in the initial and final stages of a journey is undoubtedly needed in the physical platform service, but can be enhanced to the desirable integration level by allowing remote passenger access through personal devices.

2.3.3 Interactive Browsing in Passenger Information Applications

In 2005, Washington State Department of Transportation predicted what in 2015 only 20% of travellers would be using devices for travelling needs [TK05]. By then it was already clear the benefits of using of technological traveller information services as means to improve user information, even though the percentage of users was small it was increasing fast enough to be noticeable and require research. Although the predictions were right about the growth direction, the amount was underestimated, according to recent publishers that claim that 60% of adults own a smartphone in USA [Smi15a]. A higher distribution is found amongst young adults (85%), being relevant for this context that 52% of them claim to have used it to *"Find a good way to get somewhere"*.

2.4 Spider Maps

Spider Maps, in the context of Public Transportation, are maps used to represent the transportation support of a given area, mixing elements from both a schematic map and a schematic context.

2.4.1 Definition and Constraints

A Spider Map is defined by a central area, the "hub" and the schematic set of lines that originate from it. The central area map is typically a geographic map with context, including transportation stops, while the schematic represents the route lines, as seen on figure 1.1. These maps are also accompanied by a route table which allows mapping between a chosen destination and the corresponding stop. The schematic component used does not correspond to a real representation of the geographic map lying underneath, but respects some physical constrains related to the positioning of the stations and topographic elements. The simplification is also made by allowing lines to follow only in vertical, horizontal and 45 degrees orientations, maximizing readability.

The main goal of a Spider Map is not to help planning a journey where the destination is known beforehand, but instead one where the starting location is defined and the destination might be unclear. This positions the Spider Map's information utility in the "Journey Context" and "Pre-trip-planning" according to the stages defined in section 2.3.2.

2.4.2 Automated Generation

Spider Maps are typically designed by humans, tailoring the schematic according to restrictions and optimizing user readability. This method is costly, and is might only be done only for certain locations of transportation concentration. However, recent developments have allowed for the creation of algorithms that automate the schematic generation process[Mou15], with a reasonable level of success that suggests an opportunity to improve this concept in the light of Location Based Services.

2.5 Human-Map Interaction

The spatial context of an interactive map derives from an attempt to represent a accurately a geographic context, working as a bridge to help understanding a location's bigger-picture.It is now clear the impact of technology on maps, as "modern technology affords new ways of visualizing maps and supporting wayfinding" [Por07].

2.5.1 Maps in Digital Platforms

When designing a map in a digital platform, the elements of the interface should help the user to seek the information desired. As such, technology allows has for a considerable improvement in helping the user browsing, as a mean to facilitate such a quest, given how "perception is active, not passive." [Joh13].

User involvement in digital map development is crucial, as user's positive usability evaluation is the ultimate success indicator of an interactive design. In reality, User-centered Design is of utter importance when the digital platform aims to support users' behaviour [PRS02]. Maps are a clear example of an adaptation of the human orientation behaviour to technology, and should have human psychology as orientation for its design.

2.5.2 Static Maps *vs* Interactive Maps

Maps have evolved from print support to digital and interactive maps. Verdi suggests that although what is learnt from print maps can also be applied to digital maps, the opposite does not apply, as digital maps have ”learning advantages beyond what is possible with traditional maps” [VCW02]. The interaction potential might only be diminished by the size constraints that apply when browsing in a small digital device, contrary to map browsing in otherwise bigger devices.

2.5.3 Maps in Mobile Devices

Maps in mobile devices have evolved side by side with the usage of LBS. With GPS access and services using user location as key to activities such as guidance and recommendation, maps have been made available in smartphones. It is still important to consider how map applications can be designed so to increase user information. Maps also have been designed to incorporate symbolic information besides traffic, adding other components such as points of interest [Rei04].

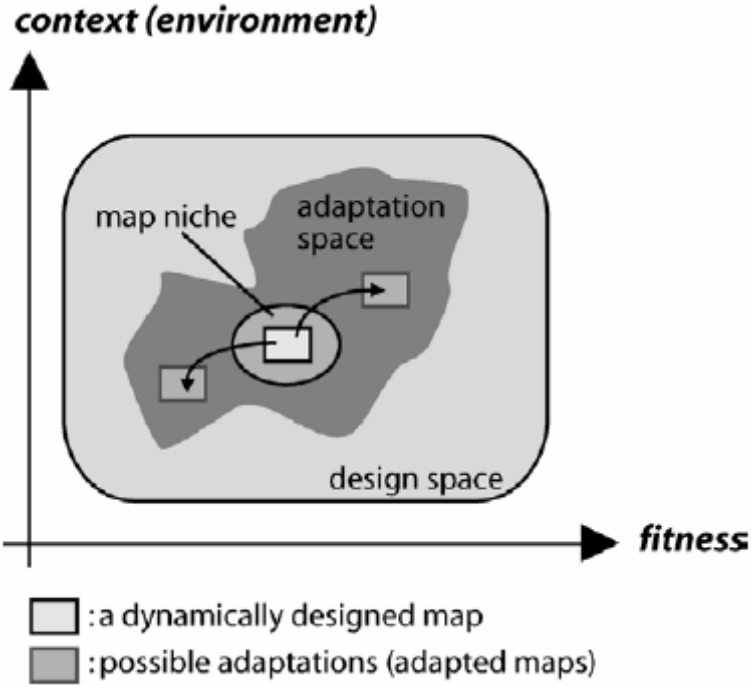


Figure 2.5: Map adaptation to user [Rei04]

There are both challenges and opportunities in map personalization and adaptation to user’s needs, as seen in figure 2.5. Map personalization can optimize information directed to a user, taking into account previous preferences, but should be approached with care, as it might restrict otherwise valid alternatives to a user, being filtered and discarded under biased learning techniques [AB15].

2.5.4 Interaction Evaluation Methodologies in Maps

User Centered Design methodologies and evaluation are core concepts in maps, as the main criteria of success of a map representation is the information retrieved by the user. As such, questioning the user on parameters concerning learning and information gain, as well as measuring time to find information sought takes part in the evaluation methodology of a given map interface.

The evaluation of map interfaces can deviate from the traditional evaluation methods using more advanced techniques such as eye-tracking [ÇHGF09]. This helps keep better track of what our brain acknowledges first in an interface, as our eyes are the primary mean of information acquisition.

It is important to consider when testing different interface alternatives, the equivalence of both representations. It is necessary to assess if the interfaces tested are both informationally and computationally equivalent, as defined by Larkin and Simon's words:

> ”Two representations are informationally equivalent if all the information in one is also inferable from the other, and vice versa. Each could be constructed from the information in the other. Two representations are computationally equivalent if they are informationally equivalent and, in addition, any inference that can be drawn easily and quickly from the information given explicitly in one can also be drawn easily and quickly from the information given explicitly in the other, and vice versa.” [LS87]

2.6 Mobile Interaction

Mobile devices have evolved oddly, in terms of interaction, booming since the appearance of touch technology. While concerns such as portability drove forward the industry before touch technology, mobile phones were getting smaller and smaller, but increased back in size again as soon as display and touch interaction became focus.

2.6.1 Common Touch Gestures

Touch interaction with maps has evolved in complexity since the appearance of touch multi-touch, which highly increased the complexity of possible gestures [KRR12].

Touch support has also increased in web technologies, with the rising in access from mobiles devices and the appearance of larger touch screens for laptop and desktop devices with touch support. The interaction between pointer devices and touch as been commonly adapted to fit the needs of the new technology.

2.6.2 Touch Support in Interactive Maps

Touch input has been used to browse maps according to typically a set of standard gestures [SDKR09]. Inputs such as *"pinch"* are typically used for zoom, as well as *"double tap"*. The *"drag"* gestures

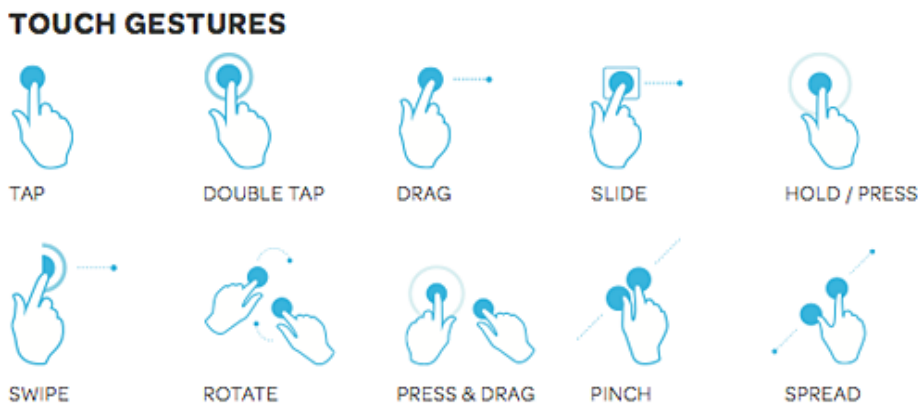


Figure 2.6: Common touch gestures in Touch-enabled devices [ges15]

allow typically for a sliding pane experience, similar to the scrolling animation in digital devices expanded from horizontal and vertical abilities to all different directions.

2.7 Conclusions

In conclusion, conditions are met for both Interaction, Maps, and Public Transportation to intertwine in Interactive Spider Maps, as all concepts focus highly on user design. The technological standards currently available enable previously non-digital services to take advantage of LBS and GIS to better increase value, adapting continuously to the growing information.

Chapter 3

Cobweb - Interactive Spider Map Solution

3.1 Introduction

An Interactive Spider Map implementation challenges the traditional Spider Map displaying method boundaries, presenting an opportunity to explore and define new interactive processes and map exploration possibilities. Such an approach ought to consider decisions ranging from the map and public transport data requirements, to visualization and application flow considerations. For this problem, the solution can be defined as successful if it is able to consistently generate Spider Maps in any location, possibly bound to certain range, covered by one or more Public Transport Networks, allowing user interaction throughout the process.

It is important to consider, for Spider Maps, what this technological approach allows concerning the dynamic map generation potential, for instance through the inclusion of date and time PT information. Extracting advantages of these data challenges the traditional Spider Map, optimizing information display according to the Public Transport lines in operation in a certain date/time range. Another possibility of dynamic data concerns real-time bus information which bestows the ability to predict the destinations' estimated time of arrival.

We shall refer to the proposed solution as *Cobweb*, referencing the interactive spider map solution described in this chapter. The word *Cobweb* was chosen by the relationship it possess with the common arachnid silk device, connecting the *Spider* Map with the complex shape it produces in a spiral pattern, resembling the map representation.

3.2 Hypothesis

The main goal of this solution is to provide a conceptual and structural basis for Interactive Spider Map applications. As such, it is key to identify the baseline goal that this solution aims to reach. Taking into consideration what the solution proposed by this research intends to solve, and the proofing that the results of the evaluation phase will present, we phrase the following hypothesis:

The adaptation of a Spider Map into an interactive application, as is defined by the Cobweb solution, improves tasks such as acquiring, understanding and exploring public transportation network information by users whose goal is finding what locations they can access from a given starting location.

3.3 General Architecture

To create an interactive Spider Map Application, we propose a twofold application linking the data control and information module (the server) and the interaction and visualization engine, as shown in figure 3.1. In this solution, we also consider that the geographic data for the tiles and map browsing section is supplied by an external service, while the Public Transport data are made available by the server, whatever the source (either homogeneous or heterogeneous) may be. This data is further adapted according to a standardized output definition, as proposed in section 3.4.1, to be supplied by the server interface.

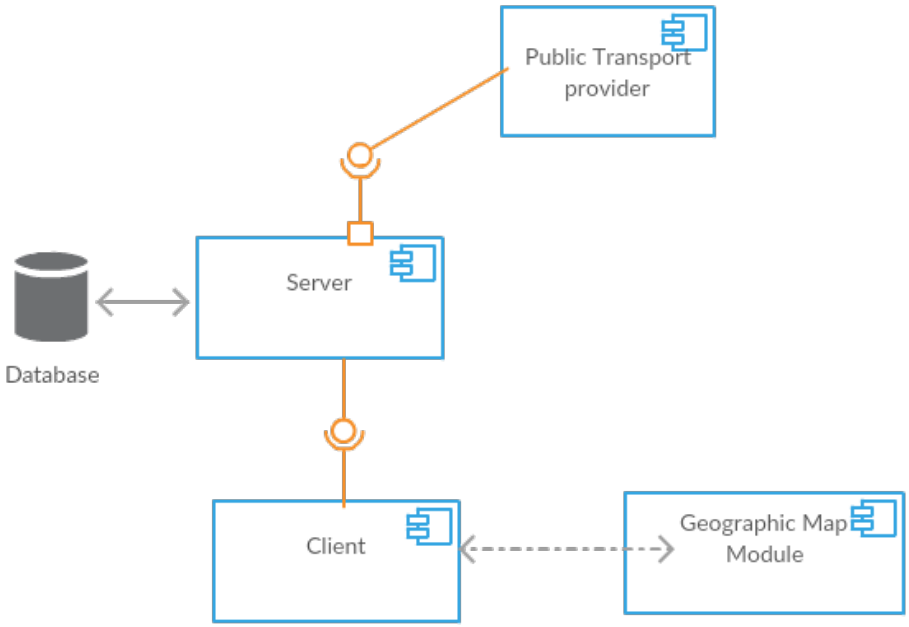


Figure 3.1: Architecture Diagram

The client application is responsible for handling most of the user interaction as well as displaying the processed schematic map and its interaction. This way, the server provides an API for the interactive application, followed by client side processing the gathered information. This ensures secure handling of the Public Transportation Data by the server supplying only the required information, while shifting the responsibility to calculate the smoother, most interactive and accurate display to the user interface.

3.4 Data sources and processing

The server can either store locally all the data used by the solution, or connect external APIs when required to access it. The main data handled by the backend[1] concerns Public Transportation, although metadata concerning user usage can also be kept or processed. Data caching can optimize the server requests by allowing process heavy tasks such as the final map generation output to be re-utilized given the same origin input. The client side should handle the geographic data and the post-processing of the Public Transportation Data.

3.4.1 Public Transportation Data

The Public Transportation Data for the application must be compatible with GTFS [Goo15] standard. This data usually includes stops, routes, trips, stop times and fares, information sufficient to comprise the Cobweb requirements in section 3.7.

To fulfill the base requirements of the solution, the essential data concerns the connections between the routes and stops, requiring an interchangeable link between both data feeds. This link must allow for stops to identify the routes they belong to, as well as access all stops of a route. Additionally, individual browsing of stops is necessary to determine the stops in the Cobweb Hub.

The additional requirements involving customized trip information, times and fares are also processed in the server side of the solution. For achieving this goal, the main PT data required involves trips, stop times and fares, which are required by the adopted standard (excluding the fees). This information allows either a pre-processing calculation of the times on each stop from a starting point in the map, or a post-processing request to ease the performance, supplying the time and fare info on demand for a given route, e.g. on a user interaction.

The specific Public Transport Data from a single source is enough for a proof of concept application, but the Cobweb Solution should take into account future PT services integration. This interoperability can be achieved using an adapter pattern to modularize the data importing process, in an attempt to simplify the process. As such, a black box adapter should be developed for every necessary origin of public transport network data [KM99].

3.4.2 Geographic Support Map

The geographic Support Map plays a larger role in a Cobweb application in comparison to a traditional Spider Map. While a normal Spider Map contains only a small bit of geographic information in the Hub, a Cobweb must consider an automated method for acquiring geographic imaging for usage through the whole process.

In the Selection process (chapter 3.5.1) the user must browse the a map to pick an area for search, which requires a map tiling display method [WJH+02], and thus a graphic map tiles provider. Considering a general source for the tiles, a simple way to define an API for retrieving the imaging is defined by the expression

[1] In an application, *backend* refers to a layer, typically consisting of one or more servers, which handle the data access and process the information

$$\{Source\}/\{Z\}/\{X\}/\{Y\}.png$$

where *Source* could be local source for performance reasons, such as a file system directory or a web source e.g, *http://.../* ; where Z indicates the zoom value and X and Y identify the tile position.

In a standard map, these variables are set to static values, and require no further graphic tiles other than the one which contains the centre Hub. However a Cobweb solution requires constant access to images as long as the user browses the map in the selection process. Moreover, in the Review process (chapter 3.5.3) the centre hub piece can use either a static image, clipped from the chosen hub bounds, or it can maintain a dynamic hub according to the zoom level (Z variable), which improves map detail according to the visualization perspective.

3.4.3 Other Data

An application such as this one might require additional data concerning user preferences. Additionally, the application can hold data concerning other details surrounding the transportation network, which can be of use to help the comprehension of the user. Data containing Points of Interest identifying specific location ranges in maps can aid simplifying the schematic map key stop locations and simplify the routes. To achieve this goal, a list of important locations data with the coordinates and range (in radius) could filter out the stops that fall within its area.

In order to increase the user context awareness, physical surroundings are important to be depicted in Spider Maps, such as rivers, mountains and seashores, etc. [Mou15]. In a Cobweb, the representation of this elements has the additional challenge of enabling consistency according to different zoom levels and beyond the borders of the visible viewport to match user navigation. This data could be depicted as polygons with fixed coordinate vertexes, however this approach assumes a consistent schematic-to-geographic scale. Large areas such as travel zones can also be defined in the same way, particularly useful as additional geographic map information (e.g. for different fares) although they present a similar challenge as physical surroundings in the Review process(chapter 3.5.3).

3.5 Interaction Process Flow

The application usage is divided in three main user interaction phases, to be considered separately:

- **Selection Process** - The user defines the area from which he wants to generate the map.
- **Generation Process** - The user locks in the selected area and interprets the map transformation into a schematic map.
- **Review Process** - The user explores the Cobweb, reviewing the available routes and stops, understanding the different possibilities and the alternative routes to travel to the available destinations.

While a standard Spider Map reading comprises the review process solely, in a Cobweb solution the generation process must be included, thus requiring the input variables that define the map to be chosen by the end user in the selection process, adding additional complexity to the Interactive Spider Map concept.

3.5.1 Selection Process

One of the main differences of an interactive Spider Map compared to a static nature of a traditional one is the ability to customize the area where one wishes to draw the map, allowing for any given input to generate a map, in contrast with a single static selection. Taking into account how a static Spider Map is made to fit an important location, the process requires editing the schematic map to improve overall quality and readability of the resulting image. However, an automated process such as this one values customization above quality, meaning that a custom selection of an area may result in a poor map or an over-fitted one, according to the coverage of the network. However, the concerns for the improvement of the schematic quality are a responsibility of the forthcoming interaction processes, needing the selection process only to ensure that any area can be selected and turned into a map, requiring the of at least one stop to lie within the Hub bounds. Consequently, this analysis must be made before generating the rest of the map, to validate that this condition is always met.

In the overall selection process, the user interaction follows closely the usual guidelines for a normal geographic map exploration [HS05]. The key operations in navigation are the abilities to zoom, pan and search. The process adds the additional ability to select an area, or to identify the current area around the user location(through GPS or other Location Based Services), ultimately resulting in a custom hub definition.

While the traditional Spider Map concerns square Hubs, it is open to the implementation of the solution to decide the shape of the area bounds. A rectangular selection has the benefit of simplifying calculating of which stops fall inside the bounds, needing only two vertexes and up to 4 coordinate comparison operations to verify a matching stop, while ruling out a stop in 3 or less comparisons. Similarly, a circular bound translates into simpler calculations of the relevant Stops, enabling the usage of the haversine formula [SS84] to calculate the distance in meters to the centre point and comparing the result the radius. In addition, it is possible for a Solution to allow for a fully customized hub, which is defined by a (preferably convex, non self-intersecting) polygon, with a number of vertexes larger than 2. This selection may be accomplished with a set of points that form the desired polygons. In this approach, an algorithm such as the point-in-polygon test [Shi62] must be applied to all the stops to calculate if its coordinates lie within the bounds. In this process there can also be constraints to the minimum and maximum size of the area, so as not to over-fit the Hub to the network, select a stopping station void location, or to match performance/readability concerns.

The technique to select the Hub location can be varied. The term "input" will be used in the selection action table referring to both click and touch actions in a given point of a viewport, while "touch" only refers to actions that are not compatible with pointer devices. The alternatives for the

selection task are described in table 3.1, in response to the indication of a selection intention using an interface element i.e. a button by the user.

Type	Action	Description
Single Input	Hold to fill	Hold down the input on a point of the map, for a shape (circle or rectangle) to generate the shape around the point, growing as time passes while holding. Input up completes the shape fixing it on the map.
	Centre and Drag	Choose a point on the map, and drag the input from that point. The finger will match the edge of the Hub, and resize while moving the input around. Input up completes the shape fixing it on the map.
	Edge and Drag	Same as "Centre and drag" but the point selected is on the edge of the shape, e.g. top left, and keeps the aspect ratio while using the drag to resize based on an invisible scale on the diagonal axis.
	Single hit	Select a point and add a shape of a fixed size to hit, for further editing if needed (or Double hit in case of touch devices).
	Fixed overlay	Fixed overlay between the user and the map, showing the area to be selected. User can drag and zoom the map at will, as the overlay stays in the centre of the viewport. Hit interface element to confirm selection, and lock the fixed shape to the map.
	Smart Draw	Each input adds a vertex to the map, completing the shape by hitting the starting point. Works with custom polygons and rectangles (if the point selected is adapted according to the position of the remaining vertexes and 90 degrees radius constraints).
Multi Touch	Pinch	Override the pinch gesture, usually used for zoom, by adding a shape in the centre of the two fingers, with a size according to the distance between them, following a linear or non-linear scale. Pinch gesture edits the size, locked and fixed on the map by the touch up action.
	Multi Hit	Use a point on each of the user fingers, adapt on movement and lock by touch up on all fingers. Requires at least three simultaneous touchdown events.

Table 3.1: Hub Selection process interaction alternatives

Editing the resulting Hub is important for either selection interaction alternatives, and essential to some. As such, after an area is defined there are two necessary operations that the solution must provide: resizing (while keeping aspect ratio) and moving the positional anchor. Additional UI elements might also be useful in this process, to show public transport elements such as stops or even routes.

If real time spider map generation is possible, this process could move on to the generation

process as soon as the area is initially selected. Alternatively, the user input could be used on the hub or the side interface to determine that he finished the selection. If implemented, pre-compiled map tiling is also relevant, as the area selected could suggest snapping to an existing spider map, as suggested in 3.7.

3.5.2 Generation Process

While the generation process might be the one in the whole application flow that requires less user interaction, it is nonetheless a critical component. Whereas the other processes require user input, the generation process mainly displays the switch between the geographic map and spider map. The main concern of this change is reducing the degree of disorientation potentially felt by the user, thus being crucial for every graphic elements to serve its purpose on providing a smooth intuitive transition.

Some alternatives are possible to better represent this transition. There are two elements whose display will change: the selected area (hub) and the remaining map. As for the hub, a loading animation could prove to be useful, if the generation is not instantaneous. For both elements, the opacity could change to indicate the change, either suddenly or gradually. The area outside of the hub could disappear entirely, fading out so as not to lose context, or fade only to certain extent, remaining transparent. In a mixed approach the hub limiting edges could be straight solid lines, or smooth edges with increasing opacity from inside to outside, portraying the notion that they belong to a bigger geographic map.

The map zoom level can also change in two ways:

- The application viewport can zoom in to fit the hub area, showing the new stations and keeping details about the schematic routes on the edges of the screen, for the moment.
- The application viewport can zoom out to show the whole generated spider map, in an attempt to help the user to understand the possible paths now created by contemplating the full extend of the result.

3.5.3 Review Process

The review process is where most of the changes to the end result of the traditional Spider Map are displayed, and allow for the innovative exploration possibilities. The main activity of the review is to browse the interactive map, using user input and to explore the available interface, in a manner similar to a geographic map. Core goals such as clearly understanding the routes and all other available information are key to match, and ideally improve the user experience when comparing with the physical traditional alternative. This Cobweb solution can be as simple as a draggable/zoomable image of a spider map to a fully dynamic multi layer set of elements portraying the map and the Public Transportation available alternatives as the user focus shifts. A dynamic approach can also take into account elements such as the current time and date to improve the quality of the supplied information, in a way which was not possible before.

One problem caused by the ability to zoom in the schematic map is that information outside the viewport will be out of sight, in contrast with the traditional map, which may cause poor user awareness of the destinations not shown on screen. To emphasize the idea that beyond the borders of the viewport there other destination alternatives, the line snipping point could add labels that inform the user of the relevant points of interest or destinations that lie beyond such a route, according to the line point of interception with the viewport. This alternative could help collapsed schematic lines to keep relevance, but it could also cause disturbance in reading, as well as overlap with other lines' labels.

The main feature that stands out in an interactive spider map is related to the ability to explore specific lines or stops according to user desire, while having the ability to alter the focus of information according to the user preference. In the traditional spider map, information display is reduced in order to include the most important stops and routes in detriment of full detail. In an interactive map it is harder to automatically decide which are the important sections to display, taking into account the large amount of possibilities. On the other hand the high burst of information can be all displayed at small bursts, when a user wants to know more about a particular stop or line. The term *highlight* is therefore used to indicate the prevalence of information regarding a particular section in detriment of the remaining background noise. The main focus of review process in this solution is therefore to explore how can highlight be used and applied intuitively, while achieving the goal of switching the informational focus display. For this goal, the options considered are in table 3.2.

While traditional Spider Maps have only some relevant stops through the line, an interactive map can have all of them according to the user desires. However, in an effort to avoid information crumbling and stop name overlapping, clustering of the stops can be made by reducing the number of stops displayed according to the zoom. Ideally, an importance scale rating could be attached to the stops to change the level of the detail desired on the display. Moreover, stop display names from nearby stops of different lines can be interpolated to result in single stops in the centroid position between them, allowing for less obsolete information display.

Traditional Spider Maps connect the hub geographic map elements and the schematic map routes through a table on the side connecting stop identifiers inside the hub with the destinations routes the user can access, emphasizing this bond by placing the start of schematic lines (from the hub) close to the streets/rails where transportation occurs and using colour schemes that match linked elements of the network. Interactive Spider maps challenge this interface by completely removing the side table and keeping a minimalistic on-demand interface which only shows the line and stops identification according to user input.

The physical map can represent road elements or other points of reference which allow a user to identify the surrounding area. In traditional spider maps, this area has been commonly related with labelled public transport stop stations, which are then associated with one or more routes. In interactive Spider Maps, the hub design can change to also include route path, while staying connected with the outside schematic, once the user highlights the route or stop, supplying all the necessary connection information to the user, upon browsing.

Option	Description
Highlight all stops	On line selection, all stops of this line can be highlighted, while removing/fading out the stops of other lines. This process allows for a normal view to have fewer stops involved whereas on highlight all information is displayed
Geographic line Hub	When highlighting a line, the way to determine which stop it corresponds in the Hub can be done by highlighting the stop. However, another way to do so would be to do display the line inside the hub connecting to the stop or stops that correspond to it
Change line opacity	While highlighting a line, the other lines opacity can be changed to values from 0% to 30% in order to remove the noise from the background.
Simple lines to Real lines	If the lines in the schematic were shifted to present a more readable display, highlighting a line could change it back to the actual geographic position (in the map) to better understand the actual functioning of the line
Stop highlight	When selecting a stop, all the lines that go through said stop could be highlighted, as well as additional information on this stop could be displayed
Simple Soft highlight	One possibility of display could be to have all lines with minimum information (destination and line identification only) and on highlight the schematic line stops could be marked, but only display information on user interaction.
Custom highlight	A user could choose to hide or show some stops for better improving line visualization, keeping this change as long as the user wishes to do so.
Highlight Popup	When an element is selected, a Popup could appear with additional line information that does not disturb the rest of the map, for user information

Table 3.2: Review process highlighting option

The updated hub and schematic map should make additional interface elements unnecessary, displaying the needed information through intuitive interaction. However, this change must be handled with care so as not lose context or information in the decision making phase. This could be done by station or stop highlighting once route is selected, or emphasizing all routes derived from a given station when a stop is selected.

Time awareness is an important factor which can change how the Spider Map is displayed, both removing unnecessary information and adding relevant new one. Without requiring user input, the spider map can fade out routes which are not available at the current time, for example, in night and day bus cycle changes. This option could also be controlled by a slider element in the user interface, where one would define the time range where he intends to use the service, to become aware of the specific details. Moreover, time can be used to predict the estimated time of arrival, if given information about the current vehicles in circulation, or by estimating using a

static schedule.

3.6 Schematic Map Generation Algorithm

The schematic map in an Cobweb Solution should look forward to its main purpose, discarding the unnecessary map elements in order to provide a clear vision of the available possibilities. A schematic route may include all the stops it goes through or only the points of interest. Zoom can also affect which elements should appear, allowing for a detailed exploration to be made while zoomed in, the most important elements when zoomed out. Time labels could also be added to the interactive map, mainly in the final destination, in an attempt to represent distance as the time it takes for a user to get there. Fading out routes and transparency are important to hide routes which are unavailable, focusing on available or selected destination.

Although there are algorithms to generate a schematic map concerning imposed restrictions [Mou15], an heuristic approach can be used to accelerate the process. Oversimplifying the schematic component can be done by matching the mapped route on the geographic map, while hiding the background. However, we highlight that for the purpose of interaction, the implications of discarding the attempt to generate the ideal design are minor, in favour of the added information value.

3.7 Geospatial System Scalability

Spider Map generation can be a deterministic process considering that any generated map which has the same input hub, using the same public transport network data, in the same geographic background has the same result. This deterministic approach does not consider dynamic data, assuming that the generation process result in a complete set of information which can then be filtered by time and date constraints on the display process. This allows storage and indexation of Spider Maps, identified by the following key inputs:

- Coordinates Vertex 1, Coordinates Vertex 2
- Centre Coordinates, Horizontal length, Vertical Length
- Centre Coordinates, Length (assuming square hub)
- Centre Coordinates (assuming square hub and default dimensions)

The advantage of defining Spider Map allow the storage of generated maps for future usage, although there are a large number maps when considering the possible hub sizes and positions. However, if a map is tiled as a grid of same-sized shapes, it might possible to obtain a reasonable number of pre-generated Spider Maps. This city-map tiling can be done for different hub sizes and initial spacing, in an effort to cover most of the possible browsable area. An overlapping algorithm can be used for a new hub selected to match with the pre-generated Spider Map hub which intersects along a larger area. This implementation can be done in a similar fashion to the tiling method described in section 3.4.2

User input on the map itself could be used to cache pre-generated maps. Users can supply information on the most relevant stops to display on the Spider Map, allowing further generated

maps to take into account user rating to optimize future user experiences. One step further in user intervention could be point by point reorganization, allowing the user to interact with the schematic map as an editable element, to shift the position of the lines to increase readability.

Storing the maps also has the benefit of users being able to share maps and retrieving maps from previous application usages.

Requirements Proposal

A list of requirements for the application implementation is created, with varying priorities related to the need of a given functionality of the solution.

Priories match with the following meanings, by order of importance:

- **Required -** Essential to the definition as a Cobweb Solution
- **Crucial -** Crucial to improve Cobweb Solution application utility
- **Ideal -** Not required, but a good improvement to the overall experience.
- **Optional -** Optional to implement, but useful nonetheless.

The requirements proposal is listed in table 3.3.

ID	Description	Priority
Browsing	As a user, I would like to browse the geographic map, using common map browsing functionalities such as drag and zoom and search.	Required
Selection	As a user, I would like to be able to select an area of the map, in an intuitive and editable fashion.	Required
Edition	As a user, I would like edit the selected area of the map.	High
Location	As a user, I would view to find my location.	Crucial
Stops Display	As a user, I would view the stops on my selection before proceeding to the map generation.	Optional
Hub Shapes	As a user, I would like to select areas with a shape different to that of a square.	Optional
Generate	As a user, I would like to validate a selected area, pointing out my desire to move forward and generate a spider Map.	Required
Transform	As a user, I would like to be able to visualize the transformation between the geographic map and the spider map, understanding the change.	Required
Zoom	As a user, I would like to view map zoomed in or out after the generation.	Ideal
Gradual	As a user, I would like to view the change to the map in a gradual fashion.	Optional
Browse	As a user, I would like to be able to browse an interactive spider map, exploring the public transport network.	Required
Highlight route	As a user, I would like to select a specific route or stop, and understand the available alternatives.	Crucial
Schedule	As a user, I would like to be able to receive schedule information that ads information to destinations, providing knowledge on how long it takes to get there.	Ideal
Time Range	As a user, I would like to alter the time range or current time of the simulation, in order to understand how the routes change.	Optional
Save & Share	As a user, I would like to get a Spider Map identifier, allowing me to access it later or share it with others.	Optional
Background Map	As a user, I would like to understand how a destination is printed in the original geographic map, to better understand my destination.	Ideal

Table 3.3: Solution Requirements Proposal

Chapter 4

CobWeb 1.0 - Implementation of an Interactive Web Application for Touch devices

4.1 Introduction

This chapter describes the implementation of a Cobweb Solution, as defined in chapter 3. The developed application aims create a fully functional application to be used for testing purposes, to validate the proposed solution, given certain additional constraints. As an interactive Cobweb application, the main target platform is a mobile device. This approach requires full adaptation to touch inputs from an interaction point of view.

For this implementation, the Interactive Spider Map application meets the minimum criteria of success if the implementation enables the generation of Spider Maps in the City of Porto (Portugal), using Porto Transportation providers, presenting a touch-enabled interactive map.

To achieve this goal, the application must use the input of the user selection (area map) and the data concerning Public Transportation, supplied by OPT [OPT16], and generate a moderately visually accurate schematic map. The application should also allow the user to interact with the result, exploring the different navigation alternatives presented, as defined in the Solution.

This application is highly dependent on interaction. As stated in chapter 2.6.1, the browser already allows for a high range of touch inputs, within an acceptable performance range. As such, we have chosen to develop the application for a web platform (such as a browser), allowing for a faster implementation and deployment, requiring no dependencies on the device. This decision is also more generic in the sense that any device can run it, being portable for both different mobile and desktop clients that support touch input. It is also possible to adapt a web application into a native mobile application [CL11], which create an installable application package which utilizes the same code as the web application, using the default browser of the device in a fullscreen mode, and launching it solely for the local hosted website. This means that developing for the web widens the range of usable devices while also being able to deploy it. Moreover, the application is also

supported on a computer device, being the touch inputs compatible with click inputs, allowing for the usage of standard pointer input peripherals.

The application is supplied by a server for handling and controlling the access to data, further described in chapter 4.3. The client browser will use map and touch *JavaScript* libraries and handle most of the interaction process as well as the calculations necessary to create a Cobweb application.

In a wordplay with the Solution name *Cobweb* the implemented application will be called **CobWeb 1.0** (note the capital *W*), using the word *Web* in the application name, identifying that it is a Web application, deployed as a website.

4.2 Deployment and Architecture

The developed system is hosted on a server that allows for a mobile device to use HTTP requests and a browser application to access it remotely. The application is aimed towards an averaged sized mobile device such as a tablet with a medium sized screen of around 10 inches, while still being functional on any browser for smaller devices such as mobile phones. This poses a challenge for the development process, requiring continuous integration to carry out testing on the specific touch inputs. A browser such as Google Chrome [chr16] supports developer tools that emulate touch inputs, but physical device testing is also key to ensure that the support is kept. However, besides the user testing phase, development of the server and non-touch input features require only a local set-up browser. The programming language focus is mostly *JavaScript*, for client, sufficing a text-editor development environment with minimal highlighting abilities. The server programming language is C# [HWG03] in Visual Studio [vis16] environment, deployed as a .Net web service.

The architecture of this project is similar to the proposed solution, being the Public Transport Data in a remote API, which requires private access credentials, making server security a moderate non-functional requirement. Database usage is not necessary, unless data are downloaded as a bulk and stored locally, or city-map tiling implemented as described in chapter 3.7, which is not the case.

In figure 4.1 an architecture diagram shows the overall view of the CobWeb 1.0 Application. Starting with the .Net Web service, the server is responsible for the access and supply of Data. To access the external data supplied by the *OPT Web Service*, a *WSDL Specification* allows the creation of an external service reference which connects to the *.Net Service Endpoint* using a service class with no need for further worries about the way the connection is processed. However, this endpoint specification from the *OPT Web Service* does not expose the full detail of the connection, requiring for all methods solely the input of a string named *encryptedJSON*. This requires a module on the *CobWeb 1.0 Web Service* named *Encryption.cs* to encrypt the *json* string with the arguments filling (supplied partly by OPT). In *CobService*, the module responsible for preparing the calls and processing the outputs to OPT is the *OTPAccessPoint.cs*, which poses an adapter for

the provided methods, allowing for the *CobService* to request Data information and provide it as a REST [Mas11] API.

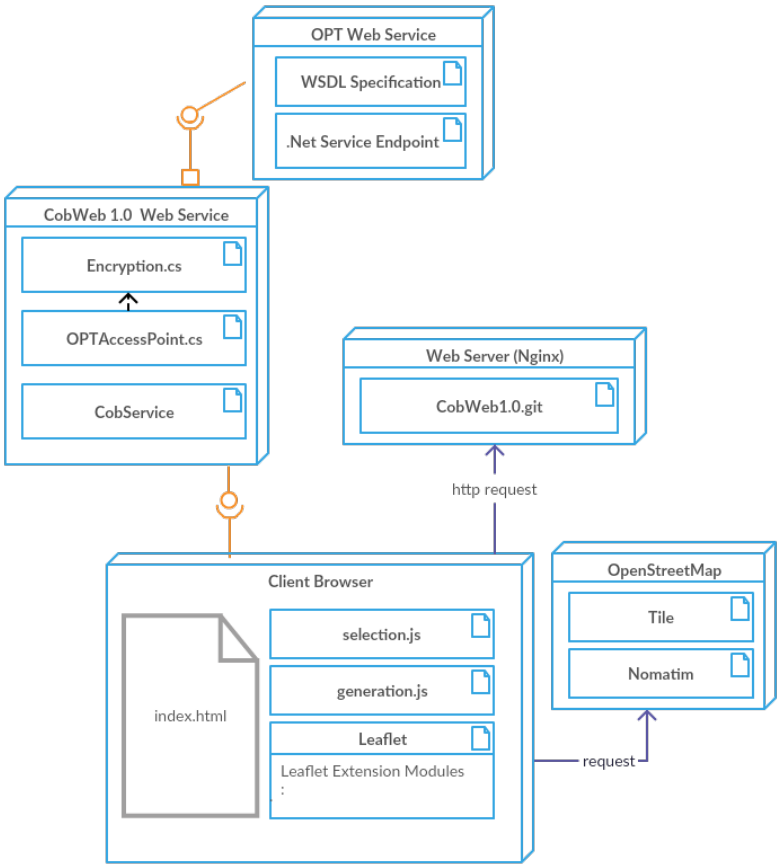


Figure 4.1: CobWeb 1.0 Architecture Diagram

The client side of the CobWeb application is hosted as a git repository on an Nginx [Ree08] Web Server which supplies the *index.html* page to the browser. The CobWeb application runs on a single HTML [RLHJ+99] page, which accesses the *CobWeb 1.0 Web Service* using HTTP requests for the data. This page contains the developed *selection.js* and *generation.js* modules, which are the main cores of the map representation, and uses Leaflet [lea16] and Leaflet modules for the map interaction and representation. Some of this modules require the *OpenStreetMap* [HW08] external server for the *Nomatim* requests (search module) and map tiles.

4.3 Server

As described in the general architecture, the Server was developed using C# programming language as a .NET Web Service. The usage of this technology was chosen to adapt with the *encryption.cs* module, supplied by OPT. The .NET Framework allows for de development of different

service alternatives, with fast interconnection between similar services following a set of standards, enabling the connection to the external OPT service as well as supplying a REST server to the client application. We shall refer to the server as application as CobService, according to the name of the project. This project was developed using Microsoft Visual Studio Community 2015, providing powerful tools for debugging and deploying the project and managing the required modules.

The CobService project was deployed using Microsoft Azure, which was configured to expose a REST API (defined in 4.3.2), accessible by the client application. This deployment was also tested locally using IIS[1], providing the same functionalities as Azure, but only for local access. The main information flow in the server is quite straightforward: The *Service* class supplies the REST endpoint which is open to requests from outside the application. Once a request arrives, depending on the data required, this class calls the *OPTAccessPoint* which uses *encryption.cs* to request data from OPT, and uses the remainder class to decrypt the data in json using *Newtonsoft Json* [new16] classes, transforming it directly into object structures. The general class model of the server is displayed on diagram 4.2.

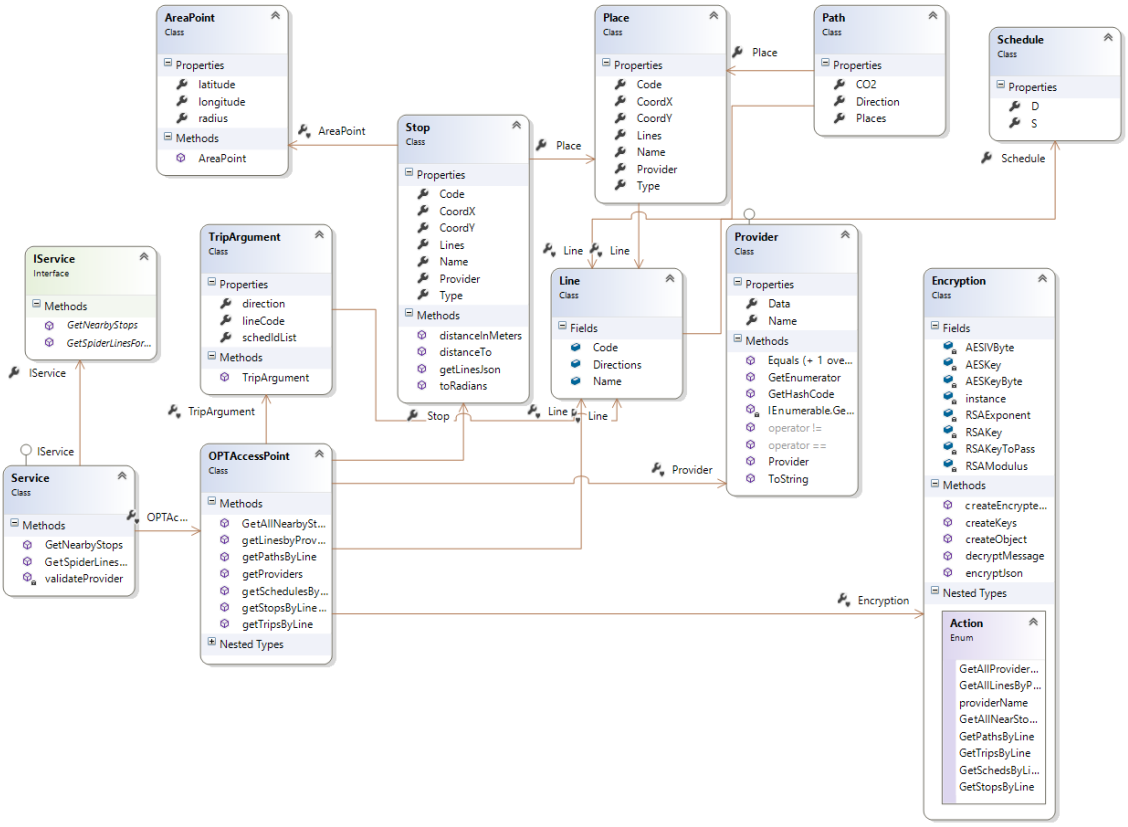


Figure 4.2: CobService Class Model

[1]IIS - Internet Information Services for Windows

4.3.1 Public Transportation Data Model and API

Even if the OPT .NET service does not expose nor define clear argument restrictions in the requested calls, the encrypted json data sent does require some conditions to be met in order to attain a reply different than an empty json object (*{}*). The exposed service allows for the following methods, all requiring an encrypted *encryptedJSON* string:

- GetStartMessages
- GetStopTimes
- GetNearStops
- GetNearPois
- GetStopsByWord
- GetPoisByWord
- GetProviders
- GetLinesByProvider
- GetStopsByLine
- GetPathsByLine
- GetSchedsByLine
- GetTripsByLine
- GetLogos
- CalculateTrip
- ReadMessage

However, not all of these methods were used by the application. The access module which handles the call to OPT service and handles the response implements methods to transform these calls into fully enabled requests. The following table describes the methods created and their respective inputs and outputs:

Method	**Input**	**Output**
GetProviders	-	List<Provider>
GetAllNearbyStops	AreaPoint	List<Stop>
GetLinesbyProvider	Provider	List<Line>
GetSchedulesByLine	Line	Schedule
GetPathsByLine	Line	List<Path>
GetTripsByLine	Line Direction List<Schedule>	List<Trip>
GetStopsByLineCode	Line	List<Stop>

Table 4.1: OPTAccessPoint Method Adaptation

4.3.2 CobService REST API

The main goal of the CobWeb 1.0 application server is to provide methods to the client browser in order to generate the spider map. This is achieved by exposing the service as a WCF [MMWW07] web service using a REST interface. This interface is defined by the methods present in table 4.2.

Type	Name	Input	Output
GET	GetNearbyStops	/nearbyStops/{lat}/{lng}/{radius}/{list<Providers>}	List<Stop>
POST	GetSpiderLinesForStops	/spiderMap/List<stop>	List<Line>

Table 4.2: CobService Interface

These methods are implemented in the *service.cs* module, which handles the requests, connects to the *OPTAccessPoint* and prepares the appropriate reply. These method do the following:

- **GetNearbyStops:** Receives the latitude and longitude of a given point, the radius desired and the providers. The providers list is used to control which providers will be ignored in the end result, as the CobWeb applications works with only with certain providers, as described in *providers.js* module in chapter 4.4.2. This list can be sent as empty for all providers to be considered. This calls the *GetAllNearbyStops* method of the interface with the specified radius. This radius is equivalent to the diagonal of the square selections and all remaining stops outside the square (but belonging to the circle) are cropped out. This method returns a list of stops with the line codes of the routes that are connected to them.

- **GetSpiderLinesForStops:** Returns the information necessary for the creation of the spider map. It receives the list of stops and retrieves the stops of every line. It does not repeat the same line more than once. The return value is a list of line identifiers followed by the stop list.

4.4 Client Web Application

The user interface of the CobWeb 1.0 application is displayed on the client side web application. This application runs on a single *index.html* page hosted on a DigitalOcean [dig16] server running Nginx, synchronized with a local Git [LM12] repository, which is prepared for Continuous Deployment by allowing all pushed changes ("pushing" is the equivalent of uploading the changes to the latest version to the repository) to be applied to the Nginx directory folder where the website is hosted. However, the website was developed on a local machine using Wamp [wam16] to pose as a *localhost*[2] page. This page can be accessed by any browser, but it was developed and optimized for Google Chrome for having the edge features required to a browser and for being available to a mobile operative system such as Android [and16].

[2]localhost - hostname that means this computer

4.4.1 Geographic Map

The core of the client application is the geographic map interaction. As such, it is important to consider which javascript library to handle the basic map interaction and information request, and then further add to this basic interface all the CobWeb 1.0 functionalities. There are several available libraries to achieve this function, so those considered for this application were the following:

- OpenLayers[3]
- Leaflet
- MapBox[4]
- Google Maps[5]

From these options the one chosen was Leaflet. There are hundreds of plugins and libraries that customize the library, being open source and prepared to extend and add more features. OpenLayers, while still being open source, does not have so many libraries and less support (Leaflet has around nine thousand posts in StackOverflow while OpenLayer has around seven thousand) while both having good documentation. MapBox is built on top of Leaflet, and besides adding additional functionalities, it also hinders the ability to customize its usage. Google Maps is a popular library, ideal for simple map integrations in websites, but however it is not open source, although it has a Javascript API using Google's code for the core display functionalities.

In Leaflet, it is possible to define the map tiling source to utilize. OpenStreetMaps was chosen for its popularity and constantly updated tiles with information. OpenStreetMap provides tiles for the whole globe, with great support in the area of Porto, being the Porto centre area filled with information not only regarding the roads and buildings but also establishments and points of interest. OpenStreetMaps provides 19 zoom levels, to be used according to the specification described in 3.4.2.

The geographic map in this application was made to occupy 100% of the page width and height, being the only element present on a page, disallowing scrolling and keeping the user locked in the map as the focus of the application from the very beginning. This map only contains the basic interface controls of Zoom in and Zoom out.

4.4.2 Modules

There are three main javascript modules developed for the cobWeb 1.0 application. The main module starts by initializing all the variables and the leaflet map with its restrictions. The selection module handles the Selection process and its necessary interaction, calculations and requests to the server. The generation module is responsible to generate the spider map and to handle the Review process. These modules are described in detail bellow:

[3]OpenLayers - Open Source mapping library. http://openlayers.org
[4]Mapbox - Mapbox is a mapping platform built on Leaflet. https://www.mapbox.com
[5]Google Maps - Google Mapping API .https://developers.google.com/maps

- **main**: The main module purpose is to initialize the Leaflet map with all its restrictions. The initial definition includes the OpenStreetMaps tiling source and the center point. It adds only one of the basic controls (the scale) for the zooming in and out functionalities. In this module the map restrictions are also set to the area surrounding Porto so that a user cannot drag beyond this point without the view snapping back to the map bounds. The external control of the Search functionality is also added using the *leaflet-search* library and using the OpenStreetMaps *nomatim* API.

- **selection**: The selection module (starting screen shown in figure 4.3) handles the Selection Process (chapter 3.5.1) adaptation of the CobWeb 1.0 API. It is responsible for implementing a side interface with all the functionalities available for this phase. It is responsible mostly for handling the selection and editing of hub. The main result of the interaction with this module is to generate a dark coloured hub in the map, marking the area where a Spider Map is to be generated, as shown in figure 4.4. It also allows for a user to find its location, focusing on the area and enabling the draw of a Hub around it. It is also possible to preview the stops in the Hub and proceed to the map generation.

- **generation**: The generation module (starting screen shown in figure 4.5) handles the creation and interaction with the CobWeb 1.0 spider map. This module allows for highlight features and overall analysis of the end result of the application. It implements different interactive perspectives. There were some factors to be considered in the way to implement this new interface with the Hub in the centre. One option was to cut an image of the selected hub in the previous phase and implement the whole interface on a new canvas using the image as a base for the layer. However, it was decided to keep the Leaflet interface to maintain context with the map bellow the hiding layer. This approach had the advantage of keeping the Hub tiles adapted to to the zoom level while the lines drawing could easily be mapped to standard positions, appearing to be on a plain surface. However, in the hiding process it was non-trivial to ensure that the layers bellow were not only hidden by a 100% opacity white layer, nor overwhelming the performance with unnecessary tile fetching operations. Another simplified option considered hiding, being faster to implement with four white rectangles infinitely sized on each side of the Hub, having the downside of being subpar in hiding the background map in the zoom operations, while also hindering performance.

  This module also is responsible for complex operations in drawing the lines. The lines points of interception with the Hub are calculated and the lines are cut into segments so that the segments falling inside the Hub are not draw, respecting the standard approach of the Spider Map. There are also other operations such as handling zoom adaptation of the new map elements (lines, stops, markers, labels) so that the user experience is improved by reducing the crumbling of the objects on the view. This map also adds a new side interface with new operations, allowing for the return to the previous screen. One core feature implement was

the ability to revert back to the origin map layer while keeping the Spider Map drawn, to give the idea of context to the user.

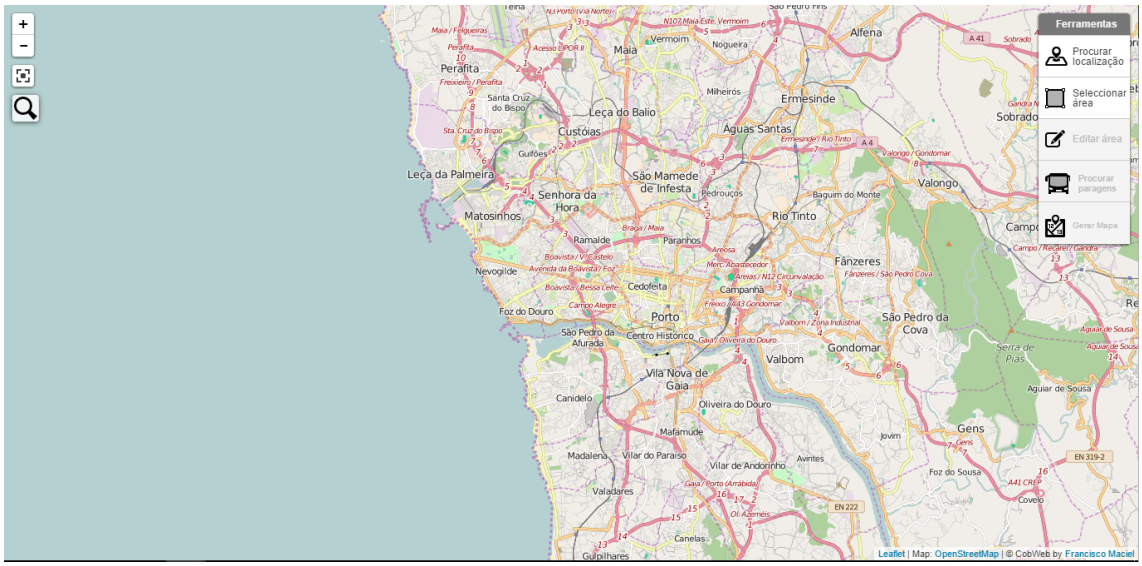


Figure 4.3: Selection process example

There are two other modules in the application which provide support for some secondary tasks:

- **providers**: The providers module is responsible for holding information on the approved providers for the application. This application uses the providers most present in central Porto area: STCP[6], Metro do Porto[7], ETG[8], Resende[9] and CP[10]. This module is also required to define each providers stop colours, stop icons and the Metro do Porto lines' colours.

- **river**: The river module defines a polygon with 305 points which depict the Douro River and the Atlantic Ocean in the city of Porto. This helps provide context in the review process and is also common in standard Spider Maps [Mou15]. An image of the river specification can be seen of figure 4.6.

4.5 Features

This section describes the final functionalities of the CobWeb 1.0 client application.

[6]SCTP - Porto Bus' public transport provider. http://www.stcp.pt
[7]Metro do Porto - Porto's Metro provider http://www.metrodoporto.pt
[8]ETG - Empresa de Transportes Gondomarense (buses). http://www.gondomarense.pt
[9]Resende - Bus Service provider. http://transportesresende.com
[10]CP - Comboios de Portugal (train service). http://www.cp.pt

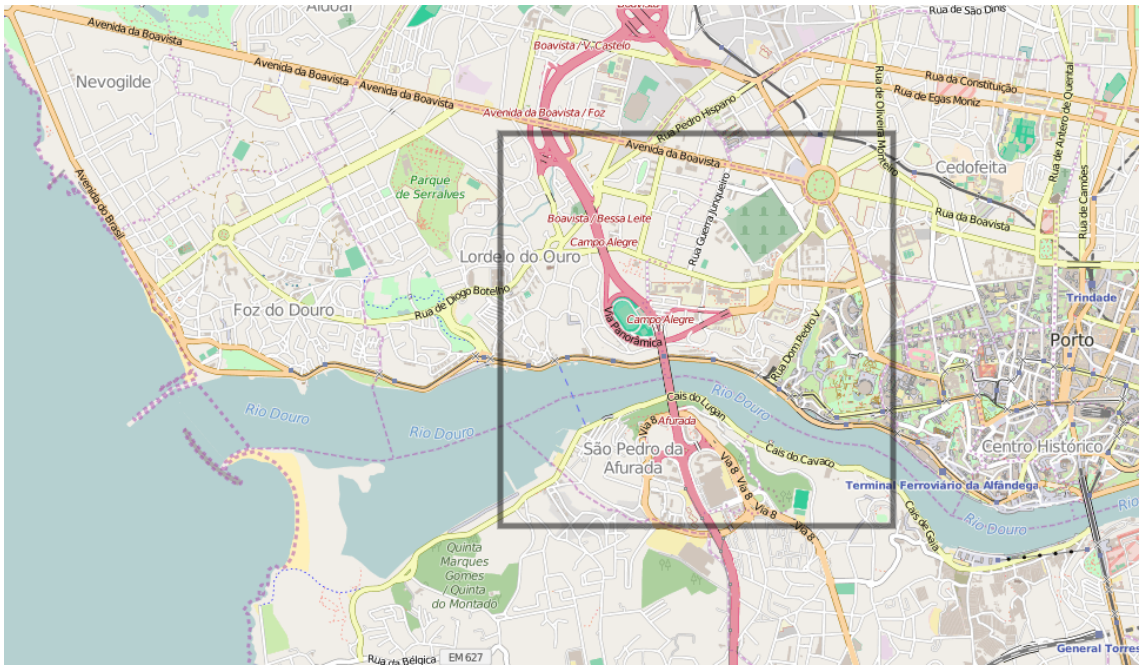


Figure 4.4: Drawn Hub example

4.5.1 Selection Process Features

These features are made available by the side interface in figure 4.7a.

- **Search:** Clicking the magnifying glass on the top left of the screen brings up a menu which can be used to search the map, using OpenStreetMaps nomatim calls, directing the user to a location after it is chosen from the list of alternatives that are presented immediately after the user starts writing.

- **Find Location:** Clicking this button will allow a user to find his own location on a map. This feature utilizes the browser location services, using HTML5 geolocation and GPS when available on a mobile device. In recent versions of Google Chrome, these types of functionalities were disabled for non-secure websites (HTTP), which means that a certificate had to be acquired and added to the CobWeb 1.0 repository server (HTTPS). However, the functionality works on most of other browsers, which poses as warning for the need to switch to a different browser when testing the application. The result of this feature can be viewed on figure 4.8.

- **Select Area:** The usage of this feature is required to access further functionalities of the selection process. It displays an interface directing the user on the way to select an area, and results on a Hub on the map as seen on figure 4.4. The different interaction alternatives for the Selection process are further described in section 4.8.1.

Figure 4.5: Generation/Review Process Example

- **Edit Selection:** This feature allows the editing of an already selected Hub, enabling the resizing (with a fixed square aspect ratio) and movement of the Hub, around the map to a new desired location, as suggested on figure 4.9.

- **Search for stops:** Searching for stops has a short delay while the application connects with server to get the nearby stops in the Hub, and displays the stops inside the Hub while still in the selection process, allowing the user to better decide on which area to generate the map, as shown on figure 4.10. These process is optimized by sending the request to the server as soon as a user locks down an area, while showing the result immediately as soon as the server has returns the answer.

- **Generate Map** This feature will move on to the generation and review process of the application, showing an animation as seen in picture 4.11. This functionality is only allowed when the user has successfully selected an area containing at least one Public Transportation stop.

4.5.2 Review Process Features

These features are made available by the side interface in figure 4.7b and through highlight operations when the user selects the map elements.

- **Focus on centre map:** Focus on the Hub, with 100px minimum padding on width and height, enabling the user to see the beginning of the lines (figure 4.12).

- **Focus on whole map:** Pressing this button focus the map on a view which contains all the Lines and Stops, as seen on figure 4.13.

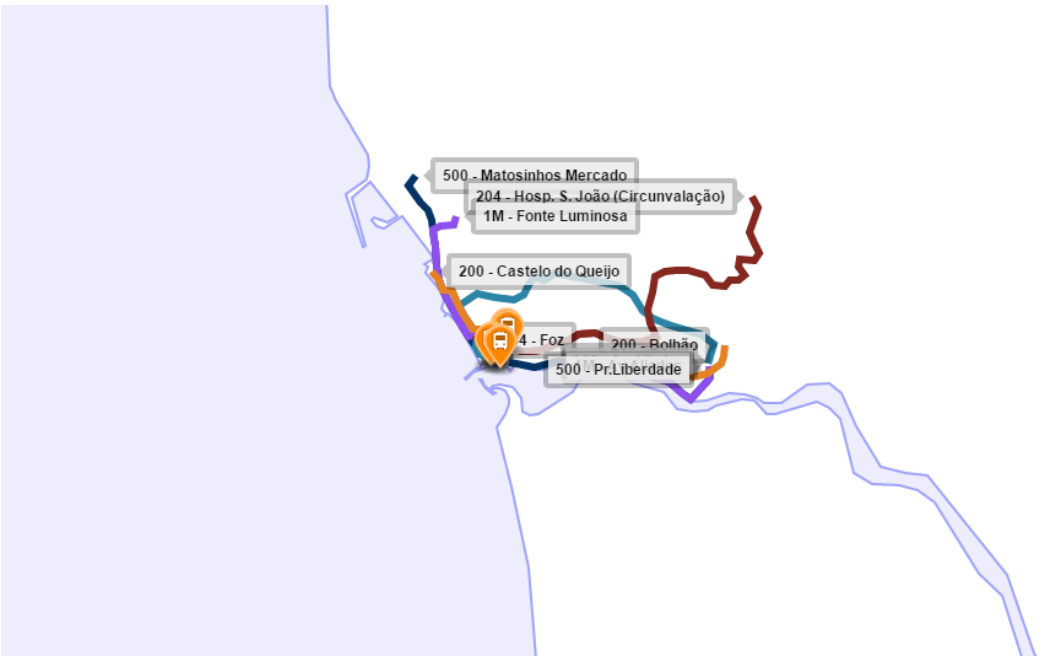


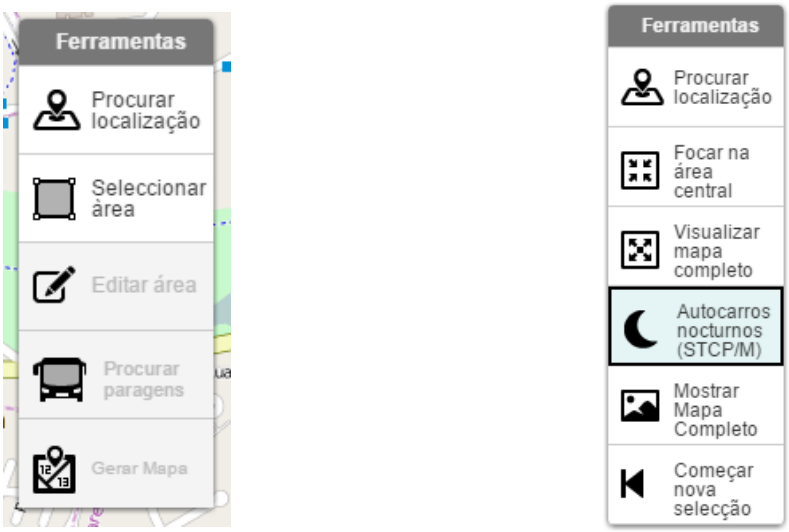
Figure 4.6: River display example (darkened colour for visibility)

- **Toggle Night Buses:** This feature considers time information regarding PT services (to some extent), as it allows the toggle of the bus lines that only work at night.
- **Show Background Map:** This functionality is a core feature of the review process, allowing for the spider map to be placed on the origin map enabling the user to compare and gain context of the location of the features, as seen on figure 4.14.
- **Start New Selection:** Goes back to the selection process, after a confirmation warning.
- **Highlighting:** This feature is not activated by an interface element but rather by clicking stops and lines, changing the focus of the map. It is possible to select stops to highlight all lines that go through it, and also to select a line to highlight it solely. The interaction alternatives in this process are further described on section 4.8.3, and displayed in figures 4.19 and 4.20.

4.6 Use cases

Being an interactive application, the main focus of CobWeb 1.0 is user centered. Diagram 4.15 shows all the possible interactions the user can make with the application, being the normal usage flow according the following pattern:

1. The user accesses the website using a browser.
2. It is possible though not required that the user finds his location.
3. The user selects a square area corresponding to the map hub.



(a) Selection Process Interface

(b) Review Process Interface

Figure 4.7: Side Interface

4. The user can edit the area selected.
5. The user can view the stops on the map.
6. The user can select a new area going back to step 3.
7. The user can generate a map. If the area selected has no stops the user goes back to step 3.
8. The user waits while the map animation is shown and the map is being generated.
9. With the generated map, the user can focus on the hub or on the whole map.
10. The user can hide or show night-time bus lines.
11. The user can hide or show the map behind the schematic map.
12. The user can click a line to highlight it.
13. The user can click a stop to highlight all lines of that stop.
14. The user can start a new selection, going back to step 3.

4.7 Plexus Testing Module

One of the goals of this research is to test the implementation of the Cobweb solution with users. A testing module named **Plexus** was developed to allow to test different interaction alternatives in the CobWeb 1.0 application. This module features the ability to create *Groups* and *Paths*. It works by displaying an interface (which can be hidden) on the lower left corner of the browser, allowing the user to customize which interaction alternative to use. The application calls the Plexus module by creating *Groups*, which correspond to sets of alternatives, populating them with *Paths*, which are the alternatives. Each path defines enable and disable actions which trigger (respectively) when it is selected or deselected respectively. This allows for on fly adaptation of the variables, without requiring refreshing the browser. This module will also keep the data regarding the options in the

Figure 4.8: Find My Location Feature

HTML5 local storage[11], allowing further access to the same page and keep the same option. The Plexus module, as it was customized for this application, can be seen on figure 4.16.

4.8 Optional Alternatives

With the development of the Plexus module, the three critical points of the application (selection, generation and review processes) were implemented with two different interaction alternatives each.

4.8.1 Square Selection

For the selection process, the two implemented are:

- **Square Fixed:** In this selection a square overlay is added to centre of the screen, covering the map. The user can drag the map bellow it, and resize the area with the help of the handles on the corners. This will change the size of the selected area, which can also be done by zooming. The selection button will remain toggled until the user confirms the selection, locking the hub onto the map. This interaction can be view on figure 4.17

- **Hold to Fill:** This selection is achieved by holding down the finger in the screen, while the hub grows at a constant speed around it. The activation of the selection button will add an overlay to the map with guidance to the user. This interaction can be view on figure 4.18

[11] HTML5 Local Storage - http://www.w3schools.com/html/html5_webstorage.asp

Figure 4.9: Editing Hub Feature

4.8.2 Generation type

In the generation process, the alternatives are related to what happens to the map as soon as it is generated:

- **Converge** The map will focus on the hub after generation, as shown in figure 4.12.
- **Diverge** The map focus on the whole map, showing all lines as stop as seen on figure 4.13.

4.8.3 Review type

In the review process, two types of interface are displayed: one showing all the information, and minimalistic on the information displayed:

- **Complete:** In the complete review, the names of all stops are shown all the time. On highlight, the line number and provider is displayed on a label, as seen on figure 4.19.
- **Minified:** In the minified review only the last stop names (final destinations) are visible, as well as the line providers and numbers at the start. After highlight, the stops of the line gain markers that a user can explore to learn more about the stops (figure 4.20.

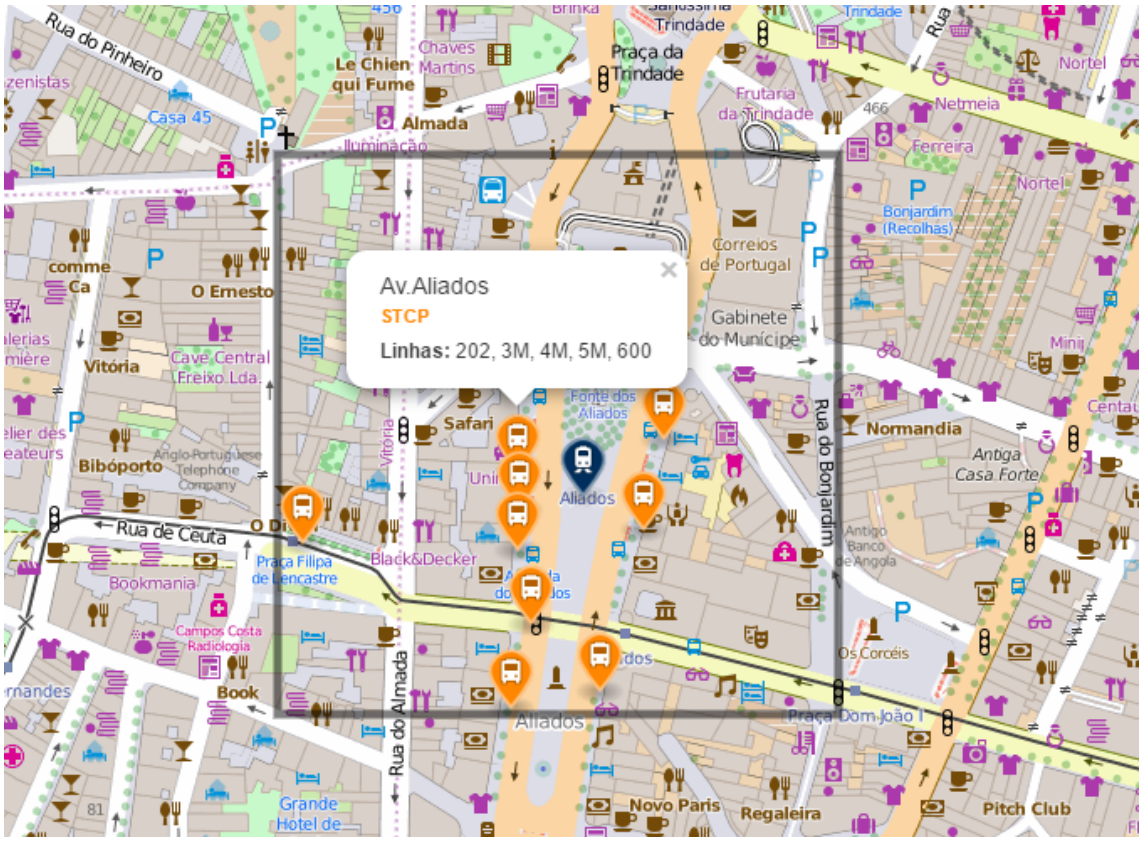


Figure 4.10: Search for Stops Feature

4.8.4 Libraries

To implement the features in the CobWeb 1.0 Web application several open source plugins were used, mostly concerning extensions to the Leaflet library. Some of this plugins source code was also customized to better fit the needs of the application. A list of the used libraries is described bellow:

- **Leaflet:** Leaflet is an "open-source JavaScript library for mobile-friendly interactive maps". This library handles most of the map features and some of the interaction.

- **Leaflet.touch.events:** This library is adapted from an implementation of touch events, although it was heavily improved to fire all the touch events on the map properly.

- **leaflet-areaselect**[12] Areaselect is used for the *Square Fixed* selection option in the selection process, allowing the creation of an overlay square in the centre of the map with the goal of the Hub selection. This library was customized to allow touch input and adapt the usage to be more fluent on mobile devices.

[12] leaflet-areaselect - https://github.com/heyman/leaflet-areaselect

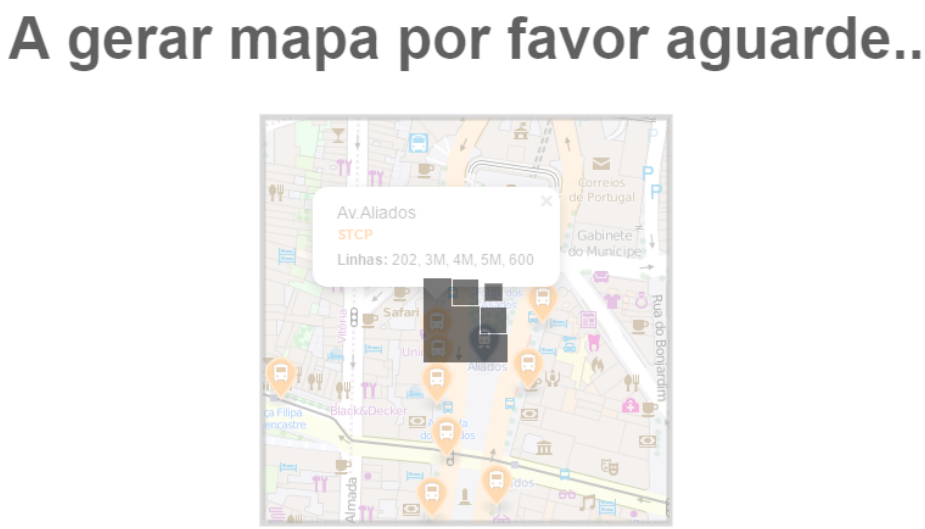


Figure 4.11: Generation Map Feature follow up Animation

- **leaflet-locationfilter**[13] locationfilter is a library with the same purpose of the Areaselect, but the filtered area is fixed on a given map position and can be resized/moved. The extension of this library allowed the creation of the editing feature, enhancing its overall use and adapting its clickable elements size to be compatible with mobile devices. A keep aspect ratio functionality was also added.

- **leaflet-boundary-canvas**[14] Boundary canvas is the main library used to hide the remainder of the leaflet map. It also enables the definition of more complex boundaries to restrict the map usage to areas such as districts or countries, given that the corresponding polygon is defined.

- **Leaflet.Fullscreen**[15] Leaflet Fullscreen adds a control interface to the chosen position which implements the fullscreen feature for multiple browsers, although some will ask for the user permission first. It also allows to switch back to normal mode.

- **leaflet-search**[16] Leaflet Search module is a plugin which allows for the implementation of a search feature in the control interface, allowing for custom or external data sources and panning the map to the found location.

[13] leaflet-locationfilter - https://github.com/kajic/leaflet-locationfilter
[14] leaflet-boundary-canvas - https://github.com/aparshin/leaflet-boundary-canvas
[15] Leaflet.Fullscreen - https://github.com/Leaflet/Leaflet.fullscreen
[16] leaflet-search - https://github.com/stefanocudini/leaflet-search

Figure 4.12: Converge on Hub Feature

- **turf.js**[17] Turf is a library which supplies multiple methods for calculations using geographic elements such as points lines and polygons. It is used to calculate the intersections between the lines and the Hub.
- **geojson-js-utils**[18] Geojson is a library that, similarly to the turf library, allows for calculations between GeoJSON [BDD+08] objects. It supplies complementary features to turf as it allows to calculate if a point falls within a given polygon, allowing for calculations concerning stops inside the Hub.
- **Leaflet.label** Leaflet label[19] adds better labels in comparison to Leaflet's standard text indications, allowing for the definition of cleared labels on important elements such as lines.

4.9 Requirements assessment

To be able to validate the CobWeb Solution in the following chapter, the CobWeb 1.0 application was developed. However, it is important to check if the implementation follows the guidelines proposed by the solution, so that the test results can give some insight in what concerns the solution. To validate the implementation, Table 4.3 assesses the solution requirements present in CobWeb 1.0.

[17] Turfjs - http://turfjs.org
[18] geojson-js-utils - https://github.com/maxogden/geojson-js-utils
[19] Leaflet.label - https://github.com/Leaflet/Leaflet.label

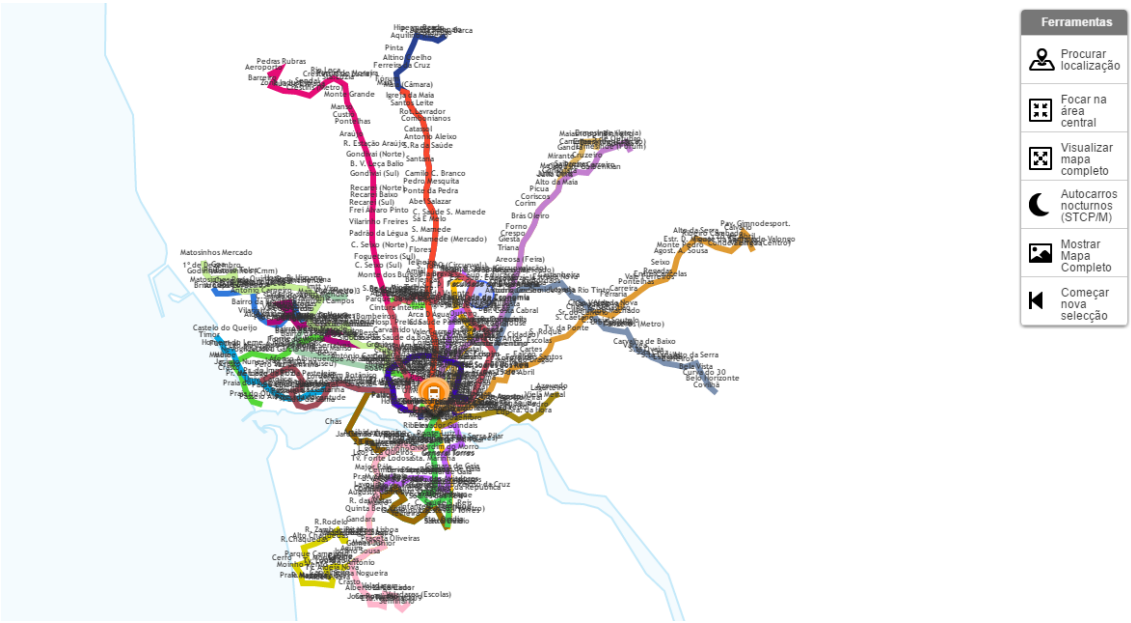


Figure 4.13: Diverge from Hub Feature

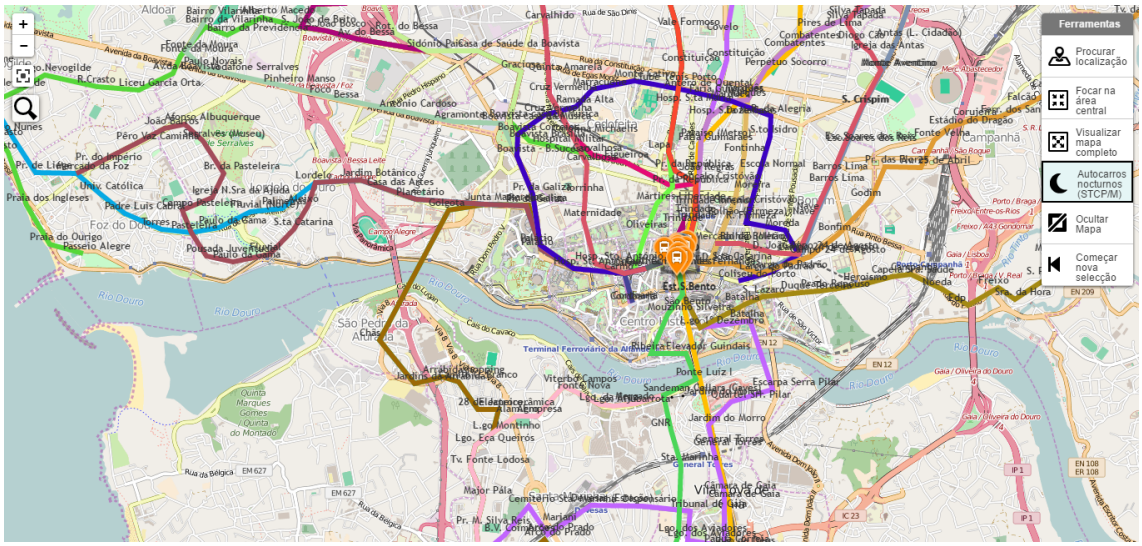


Figure 4.14: Show Background Map Feature

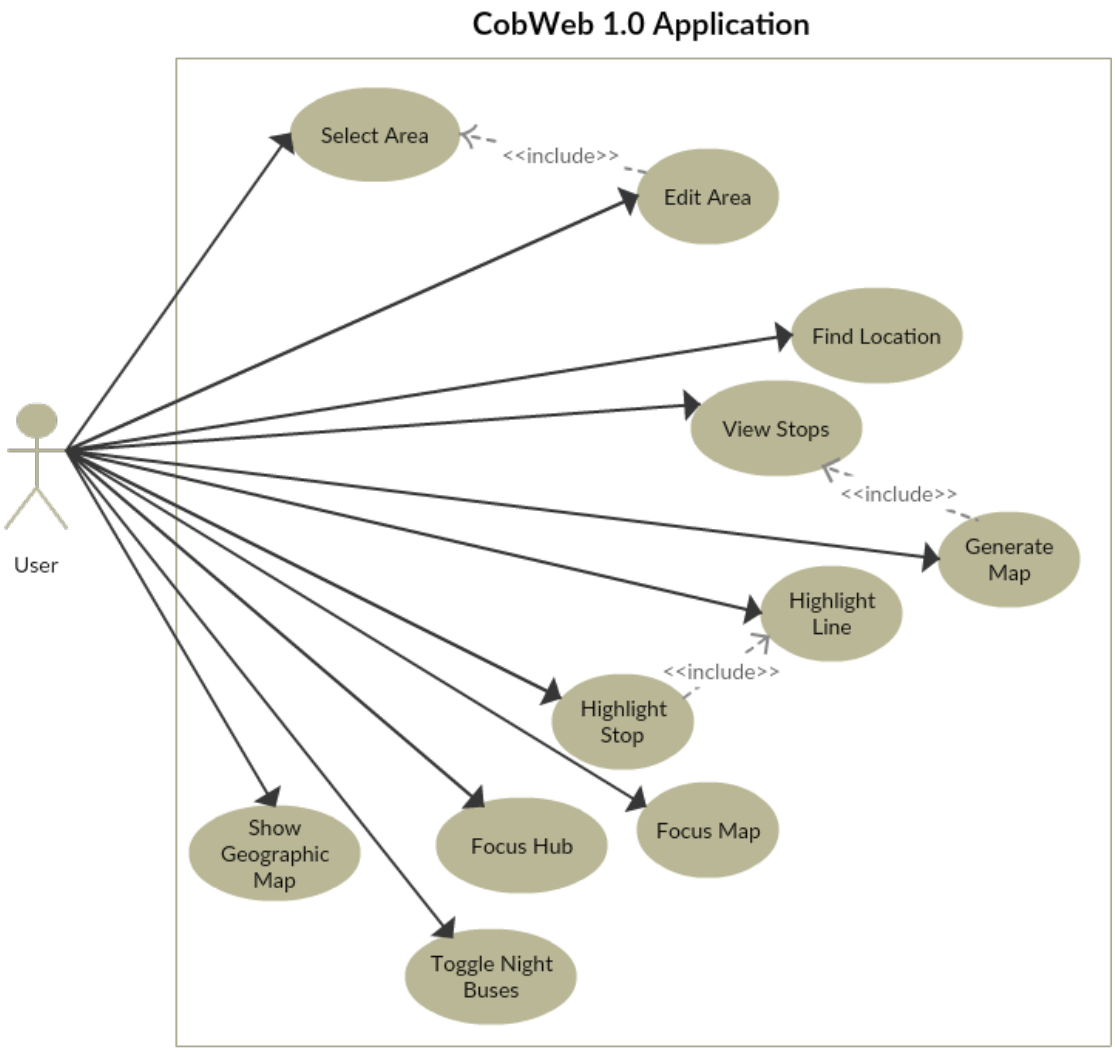


Figure 4.15: Use Case Diagram

Figure 4.16: Plexus Testing Module

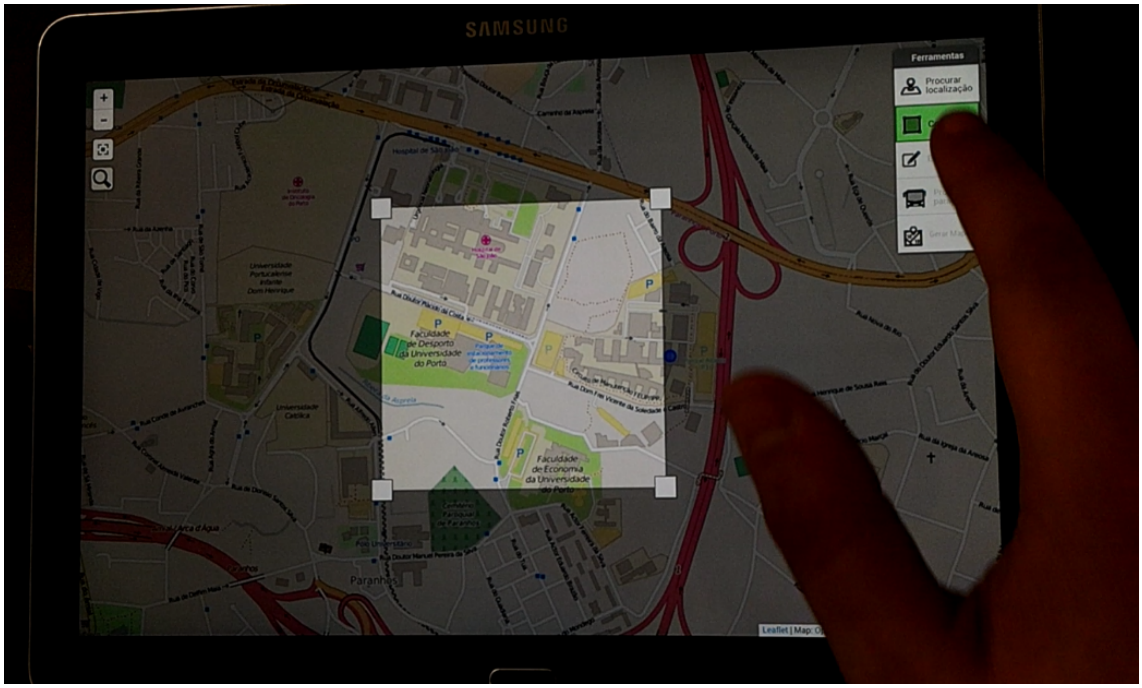


Figure 4.17: Square Selection Alternative 1: Square Fixed

Figure 4.18: Square Selection Alternative 2: Hold to Fill

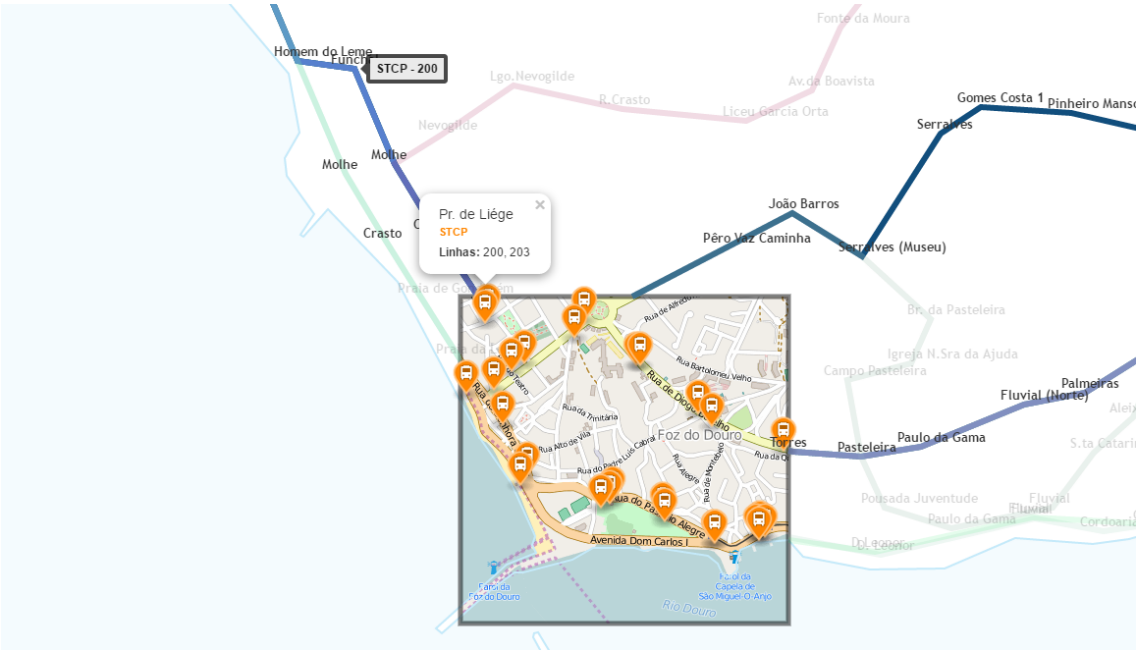


Figure 4.19: Review Type Alternative 1: Complete

Figure 4.20: Review Type Alternative 2: Minified

ID	Priority	Implementation	Details
Browsing	Required	Yes	
Selection	Required	Yes	
Edition	High	Yes	
Location	Crucial	Yes	
Stops Display	Optional	Yes	
Hub Shapes	Optional	No	Only square shape was implemented
Generate	Required	Yes	
Transform	Required	Maybe	Transformation has a loading animation, but it is yet to be tested if it is understood
Zoom	Ideal	Yes	
Gradual	Optional	No	Transformation is sudden
Browse	Required	Yes	
Highlight route	Crucial	Yes	
Schedule	Ideal	No	OPT Data had trips and schedules information, however the adaptation to a GTFS like format to easily process data should be implemented in future iterations of this application
Time Range	Optional	Maybe	Night bus lines can be toggled, however a specific time range is not clearly defined
Save & Share	Optional	No	Not implemented
Background Map	Ideal	Yes	

Table 4.3: Solution Requirements Proposal

Chapter 5

Evaluation and Validation

5.1 Introduction

Taking into account that the final goal of the application is to provide benefits to real users, it is important that the validation of the solution is focused on user feedback and user usage metrics. To achieve this goal, user tests were designed and applied. The methodology and procedures used are described in this chapter, followed by an analysis of the results.

5.2 Methodology

To evaluate the solution, an implementation was developed, as described in chapter 4. To validate the solution, user testing were created, aiming to evaluate the application in the following terms:

- The application provides user with relevant information regarding the Public Transportation services of Porto, in tasks where a user wants to know where he or she can travel to, from a given area.
- Given the different interaction alternatives, which are preferred by the users.
- The developed application provides an improvement regarding standard Spider Map.

Users participated on these tests willingly and with no promise of a reward. They were invited into a quiet room, with a desk and one chair. The test guide was also sitting on another chair next to them taking notes and noting times, while providing the minimal required explanation to start the test, making sure that everything works according to the plan. The test guide also asked the users, between the first and second task, to access the tablet device to alternate the application between the two interaction alternatives (using Plexus module - chapter 4.7).

In front of the users, there as a laptop and a tablet device (Samsung Galaxy NOTE 10.1 2014 Edition [gal16]). The laptop contained an open browser with the form (full form in Annex B) providing the instructions, and another tab with a spider map image (figure 5.1). The tablet was placed in front of them, on a comfortable position, with the screen locked but with the browser

already open. The application was launched, on the Opera [ope16] browser, where all the features were previously tested to check if all of them worked according to specification. The tablet was connected to the website using a reliable wi-fi connection, and the tablet had GPS activated for the location of the user.

In order to test different interaction alternatives, two alternatives were prepared using the Plexus module for switching between them, as described bellow:

- **Alternative A:** The Selection process used the **Square Fixed** (chapter 4.8.1), the generation used the **Converge** option(chapter 4.8.2) and the review was **Complete** (chapter 4.8.3). In general, alternative A provides a more complete interface with all the information available. The Selection uses an overlay which lets the user have full control of the area, clearly defining the bounds. In the Generation, the converge alternative shows the main map, highlighting the geography first to present all available stops and the start of the routes. In the Review, the user can see all the stop names of all the routes at once, with no need to highlight other than to emphasize the selected elements above the others.

- **Alternative B:** The Selection process used the **Hold to Fill** (chapter 4.8.1), the generation used the **Diverge** option (chapter 4.8.2) and the review was **Minified** (chapter 4.8.3). This approach is more minimalist and presents only the base interface elements required for basic interaction. In the Selection, the user must only hold the screen to fill an area, needing no input anywhere else. The generation shows the full map, consisting of an overall view, which does not emphasize any particular element. The Review is simplistic as it does not present the names of any stop besides the final destination unless the user requires so.

To test these different alternatives without being subject to the different ordering of the two scenarios, all users tested both alternatives, but in different orders. This means that some users started with A in Scenario 1 and followed up with B in Scenario 2, while the remaining users did the opposite. We can refer to these groups as AB group and BA group.

5.3 Tasks Preparation

The test consisted of two scenarios followed by a simple questionnaire. These tasks were prepared taking into account the target sample. Given that the CobWeb 1.0 solution could give very different results to different users, the first task was focused on a place familiar to all (the Asprela Campus area). As such, all users were students of a faculty that was at most 1000 meters away from the general campus area, so that the first approach to application could be similar to all.

In the beginning, users were given the following text for reading, to understand the purpose of the application:

> *This experiment consists in testing a web application which main goal is to explore the public transportation network on the city of Porto, and the area around it. This application has the main advantage the exploration of the coverage of the network around a set area.*

> *The main question to which the application aims to answer is: "From the area I am, (or where I plan to be) where can I go to, using the Public Transportation services?"*
>
> The Public Transportation services present are STCP, Metro do Porto, CP (Portugal Trains), Resende and EDG (Empresa de Transportes Gondomarense).

The text was followed by a first question, asking users the degree of familiarity they had with each of the transportation services in the application, ranging from 0 (no familiarity) to 4 (great familiarity).

Then, the test users were presented to an image of a Spider Map in the Area of "Hospital de São João", an area close to the Asprela Campus, which was the area to be used on the first scenario. This image (figure 5.1) was shown to users for approximately 15 seconds, and they were asked to try to interpret what they saw based on the goal of test, explaining that the image shown corresponds to a Spider Map that is usually posted on a physical location with a large size, showing the available transportation alternatives. After reviewing this image, the users were directed to continue to the following page of the form.

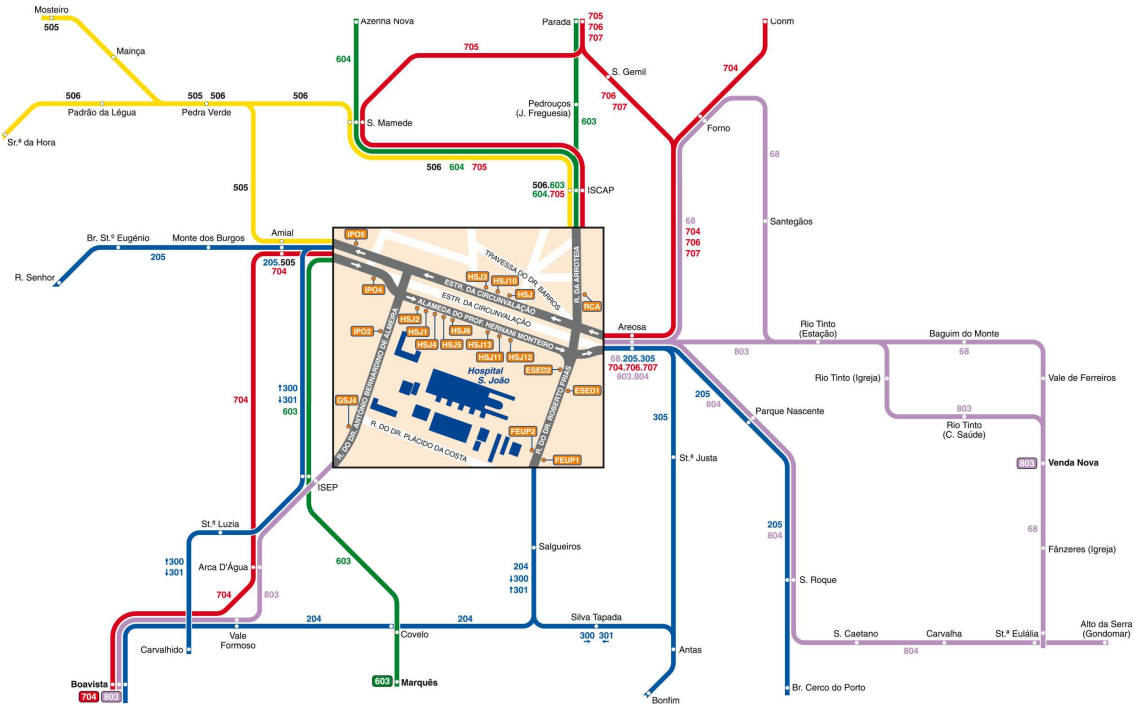


Figure 5.1: Standard Hospital de S.João/Areaosa Area Spider Map

5.3.1 Scenario 1

Scenario 1, involves the area of Area Campus and the generation of an Interactive Spider Map around it, or nearby. This area was chosen for its hight degree of familiarity to all users. As this is the first interaction task with the application it is expected that a user takes some time to

understand how the CobWeb 1.0 Application works, so it is considered important that at least the area is familiar, so that in future tasks the challenge with the application is reduced. Therefore, this task is, overall, a task based on comprehension while the second one is exploratory.

The following description of the task was given to users:

> **Scenario 1 - University Student**
>
> *In Scenario 1, embody the roll of a student who will enter for the first time in the University of Porto, being enrolled in one of the several faculties in the Asprela Campus. Your goal as a student is to try to understand what destinations you can reach using the several Public Transportation services in this area, having the possibility to move on foot on a radius of 1000 to 1500 meters, from the chosen faculty.*
>
> *You can also assume that you are going to rent a place in this area and desire to explore the city in your leisure time, or even travel back home.*
>
> To evaluate this exploratory activity indicate (when you are finished) three or more destinations that you can reach, and the way you are going to reach them (including travelling to the stop and from the stop to the destination).
>
> Note: You can explore several areas around Asprela Campus.

After reading this scenario, users were incentivized to unlock the tablet, put the application on fullscreen and start using the application. After completing the task, the users had 5 questions to fill-in (3 were mandatory) with the found destinations:

- Destination $1-5$
- Way and/or lines utilized to reach destination $1-5$

5.3.2 Scenario 2

In this scenario, users are expected to be a bit more familiar with the general functionalities of the application. As such, this task is more exploratory in the sense that, while in previous task they had only to generate one map, this time they are asked to repeat the process three additional times. In this task, users are proposed to embody the role of a tourist who wishes to visit the city or Porto area, and is aiming to choose a location based on the public transportation network services available. The goal of the task is to study at least three different areas, either from the suggestions or any other area that the user desires to explore, and to perform a generic analysis of each one, grading the best and the worst for the tourist. The task description is presented as follows bellow:

> **Scenario 2 - Tourist**
>
> *In this scenario, embody the role of a tourist that is coming to visit the city of Porto and is looking for a place to stay. As you do not plan to rent a car, you are worried about the coverage of Public Transportation in the area where you are renting a place. You have several alternatives available to choose and price is not a relevant factor.*

You are quite indecisive, and you have several choices: Trindade, Baixa do Porto, Senhora da Hora, São Benta, ... You are also pondering exploring other peripheral areas, you can chose whatever area you like or is familiar to you.

Explore at least three possibilities and indicate, based on the access to Public Transportation, which one do you think that you be the best and the worst.

While completing the task, the user could enumerate on the form the areas they explored. The following form was then presented to the user, in order to grade the areas explored:

- Enumerate the areas you have explored:
- Best Area Found (and justification for the choice).
- Worst Area Found (and justification for the choice).

5.3.3 Questionnaire

To finalize the test, the following questionnaire was given to the users in order to evaluate the overall application:

- Concerning the goal of the application: (1 - I did not understand, 4 - I understood)
- In the case of not understanding the goal of the application, what were the main challenges?
- How do you evaluate the the usability in the Scenario 1 - University Student? (1 - Very Poor, 4 - Very Good)
- How to you evaluate the quality of the information in Scenario 1 - University Student? (1 - Barely Relevant, 4 - Very Relevant)
- Do you think that this tool was useful to provide answers to the goals of the Scenario 1 - University Student? (1 - Not Useful, 4 - Very Useful)
- How do you evaluate the the usability in the Scenario 2 - Tourist? (1 - Very Poor, 4 - Very Good)
- How to you evaluate the quality of the information in Scenario 2 - Tourist? (1 - Barely Relevant, 4 - Very Relevant)
- Do you think that this tool was useful to provide answers to the goals of the Scenario 2 - Tourist? (1 - Not Useful, 4 - Very Useful)
- Which interaction alternative did you prefer? (A - Scenario 1 - University Student, B - Scenario 2 - Tourist)
- In concerns the functionality of "Showing Background Map", how to you grade the importance of this alternative for the comprehension of the context? (1 - Not important, 4 - Very important)

- In which scenario was the "Show Background Map" feature most important? (A - Scenario 1 - University Student, B - Scenario 2 - Tourist, C- None)
- In a general sense, how useful is this application? (1 - Not useful, 4 - Very useful)
- In comparison with the standard spider map shown in the beginning, how advantageous is this application (namely, but only restricted to, Scenario 1)? (1 - No Advantageous, 4 - Very Advantageous)
- If you feel the need in the future similar to one of the scenarios presented, would you use this application? (A - Yes, B - No)
- What are the main reasons why you would or would not this application?
- What would you add to this application in the future to make it better?

All questions were mandatory, excepting the one to explain the reason why the user did not understand the goal of the application (second question) and the last one for future improvements.

5.3.4 Additional Metrics

Besides the questionnaire, additional metrics were retrieved during the test by the test guide, using a pen and a sheet next to the user, and a chronometer.

Concerning the general metrics of the test, the following data was noted down:

- User Age
- User Gender
- Student Institution in Asprela
- Student Course
- Time when test started
- Time when the user finished Scenario 1
- Time when the user finished Scenario 2
- Time when the test ended
- Additional user feedback

Moreover, a table for measuring specific details of the test was prepared, filling a row for each participant, with the following data:

- User Test identification number
- Scenario 1 - Interaction Alternative
- Scenario 2 - Interaction Alternative
- Scenario 1 - Time in selection phase
- Scenario 1 - Time in review phase
- Scenario 1 - Number of functionalities utilized
- Scenario 1 - Number of interaction frustrations
- Scenario 2 - Area 1 - Time in selection phase

- Scenario 2 - Area 1 - Time in review phase
- Scenario 2 - Area 2 - Time in selection phase
- Scenario 2 - Area 2 - Time in review phase
- Scenario 2 - Area 3 - Time in selection phase
- Scenario 2 - Area 3 - Time in review phase
- Scenario 2 - Number of functionalities utilized
- Scenario 2 - Number of interaction frustrations

By interaction frustrations, we consider the occurrences where a user is clearly frustrated by the interface in the sense of an expectation not met when performing an action and being incapable of obtaining the expected response from the application. Examples of behaviours considered frustrations are; when a tap is made and nothing happens; when the user fails to select an element of the interface; when the interface takes too long to answer and the user feels the need to touch again.

The functionalities used count the number of tools utilized from the available set of the CobWeb 1.0. These tools include the side interface tools but also search and highlight.

5.4 User Sample

This test was applied to 16 users, with ages with an average of 23 years old, ranging from 20 to 28. From the participants group, 7 were female and 9 were male. All users were students from the Asprela Campus, 5 from Educational Sciences, 4 from Informatic Engineering, 3 from Multimedia, 1 from Psychology, 1 from Environmental Economics and Management, 1 from Tourist Activities Management and 1 from Physiotherapy.

The test was conducted in portuguese language, both the form and the application. All 16 users were portuguese speakers, 15 of which were native Portuguese.

5.5 Results

The results can be analysed from three points of view: overall results, AB results and BA results. The overall results can be used to extract generic conclusions regarding the tests, while AB and BA results show be compared with each other in an attempt to understand which of the alternatives is, in general, better.

The first data to be analysed is summarized in figure 5.2, concerns the overall time metrics, in minutes. As we can see, in the Test Time Comparison chart, the tests took on average 31 minutes. In an Overall/Scenario1/Scenario2 time comparison, the BA test alternative was faster, averaging 29/11/10 minutes respectively, against AB which took 33/14/12 on the same phases. Note that the data on AB and BA is provided by different users, being the reasonable sample size sufficient to make the comparison.

It is important to analyse each of the interaction perspectives, distinguishing between each phase of the interaction, the selection and review processes, for both Scenario 1 and Scenario 2

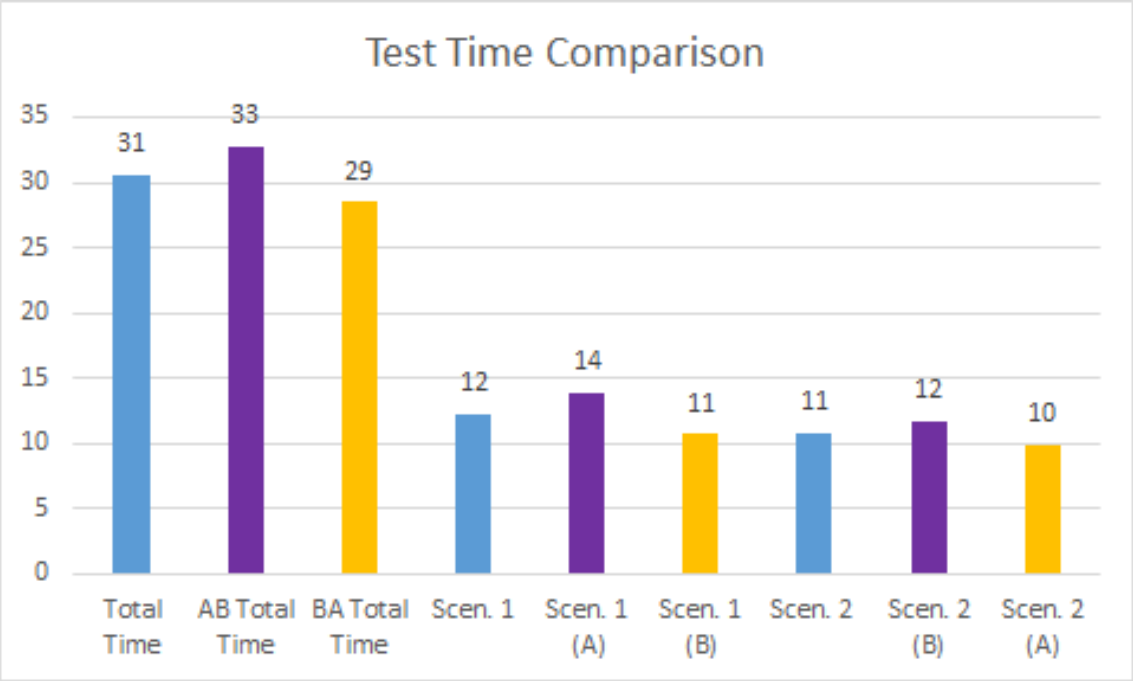


Figure 5.2: Test Time Comparison (min)

separately. For each one, interaction alternative A and alternative B are displayed in figures 5.3 to 5.6. The correlation between frustrations and tools was also analysed, being visible a dependency between both variables. This means that users who use more tools also present more frustrations, possibly being a consequence of more usage. The frustration average also reduces from Scenario 1 to Scenario 2, going from 4.25 to 3.5, a natural evolution for a user who is starting to better understand the application.

Starting with Scenario 1, in figure 5.3, we can see that option A takes, on average more time both on the Selection and Review. In figure 5.4, we can see however that both Tools and frustration is higher on alternative A in comparison to B.

In Scenario 2 however the time results are opposite to Scenario 1. Although not much difference is noted on the Review process, the Selection time is favoured for the A alternative (figure 5.5). Similarly, the opposite is also verified on the tools and frustrations, which are both higher for the option B(figure 5.6).

These results may suggest that option A is harder to learn, but easier to master, as the users who have option A on the first scenario showed that they more time with the interaction. However, option A on the Scenario 2 results in faster times and less frustration, indicating a possible preference on the long run.

Concerning the user responses to the questionnaire, we shall focus on the general feedback, without analysing the differences between AB and BA tests, which are explained further ahead. In general, the responses follow the trends described bellow:

- **Transportation Knowledge:** In relation to Public Transportation information, most users

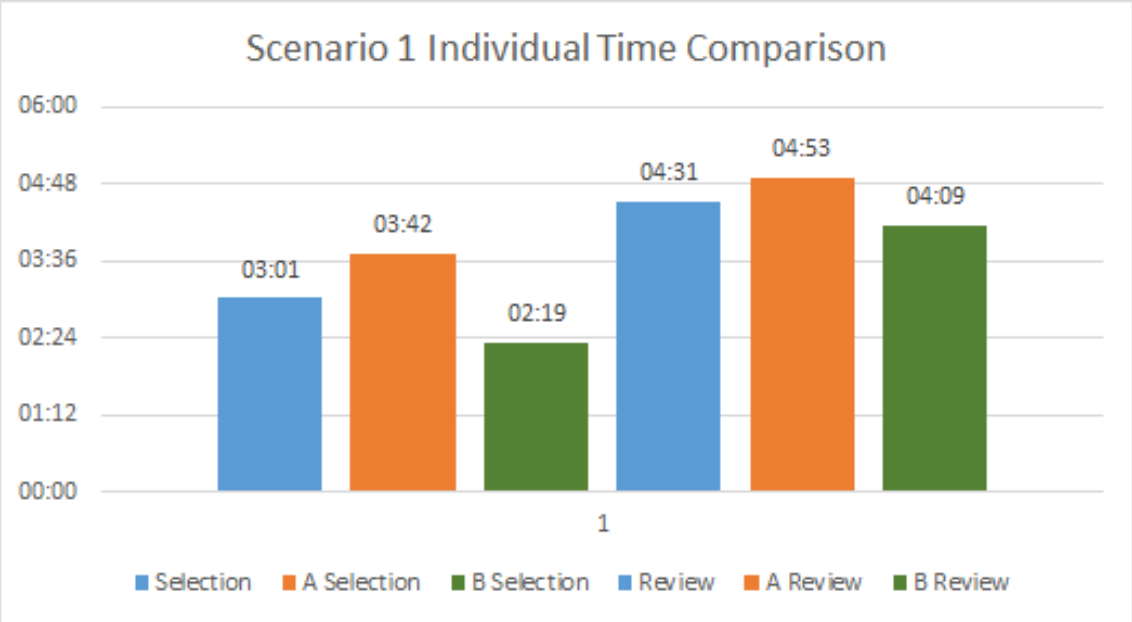


Figure 5.3: Scenario 1 Time Comparison (min)

showed knowledge of at least one provider, being most of the them familiar with Metro do Porto and STCP. CP is on a middle ground while Resende and ETG are less known by the average user.

- **Goal of the Application:** over 93% of users understood well the application goals, with nearly 70% having understood them completely. This indicates that the description of the test was clear and the overall usage clearly reflects the purpose of the application.

- **Scenario 1 - Usability, Information and Usefulness:** The feedback to the Scenario 1 of the user testing was positive. The usability was considered by 75% to be good or above. The information feedback was positive, 87.5% of users finding it to have a quality relevant or above. In terms of usefulness the positive responses were also positive, having 93.8% found it Useful, but only 25% of users considered it to be very useful.

- **Scenario 2 - Usability, Information and Usefulness:** The feedback for Scenario 2 was favourable, being a little bit superior in terms of Usability and Information to in comparison to Scenario 1. This difference can be caused by the increased knowledge the user acquires on the system, perceiving a better experience after having the appropriate knowledge to take full advantage of the application. The usability is graded as good by 62.5% of users and very good by 18.8%. The quality of information is probably were the difference in knowledge has the higher impact, with 93.8% of users considering it above relevant. The usefulness for Scenario 2 falls a bit short against the Scenario 1 as 87.6% of users consider it useful or above. 43.8% of the users found it very useful, superior to Scenario 1 (25%).

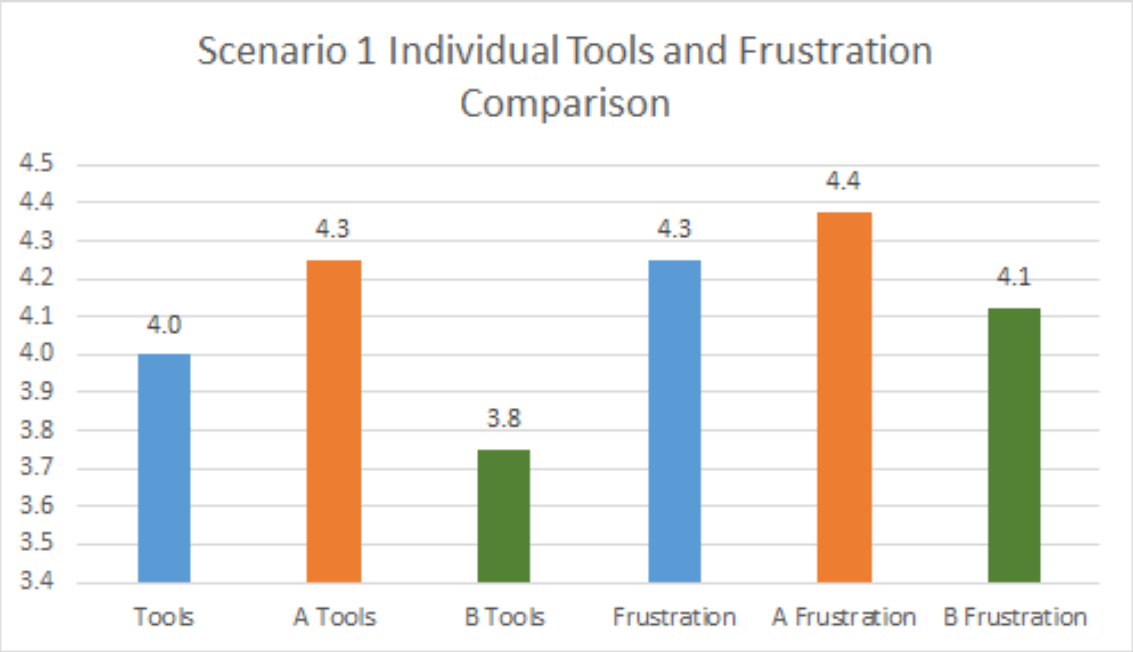


Figure 5.4: Scenario 1 Tools and Frustration Comparison

- **Preferred Interaction Sequence in scenario:** As these responses mix both A and B interaction alternatives, it is expected that the percentage falls within the 50% mark (figure 5.7) as the preference for scenarios is still biased by the alternative. The amount of users that preferred Scenario 2 is 56.3%, leaving the remainder 43.8% to the Scenario 1. Once again, the scenario that is presented first might have caused the slight tilt in favour of the one which benefited by the additional user acquired knowledge and experience. This metric is split in AB and BA interaction alternatives interpretation further ahead.

- **Show Background Map:** This feature was implemented after following up an additional feature described in the Cobweb Solution, and received an astounding feedback from users. 50% of the users considered it somewhat important while 43.8% found it very important, adding up to 93.8%. The general feedback received by the test guide also highly favoured this feature. It is important to note that 56.3% of users found it most important in Scenario 2, being the main justification for this choice the more important applicability on the Tourist's goals who does not know the map and can better understand the schematic context if it is shown.

- **Application Usefulness:** In terms of usefulness, the application scored positively, having 56.3% of users considered it reasonably useful and 37.5% very useful. This matches the following question, having 93.8% of users replied that they would use this application in the future, if they felt a need similar to the described scenarios.

- **Standard Spider Map:** It is a clear opinion of 100% of the users that the CobWeb 1.0

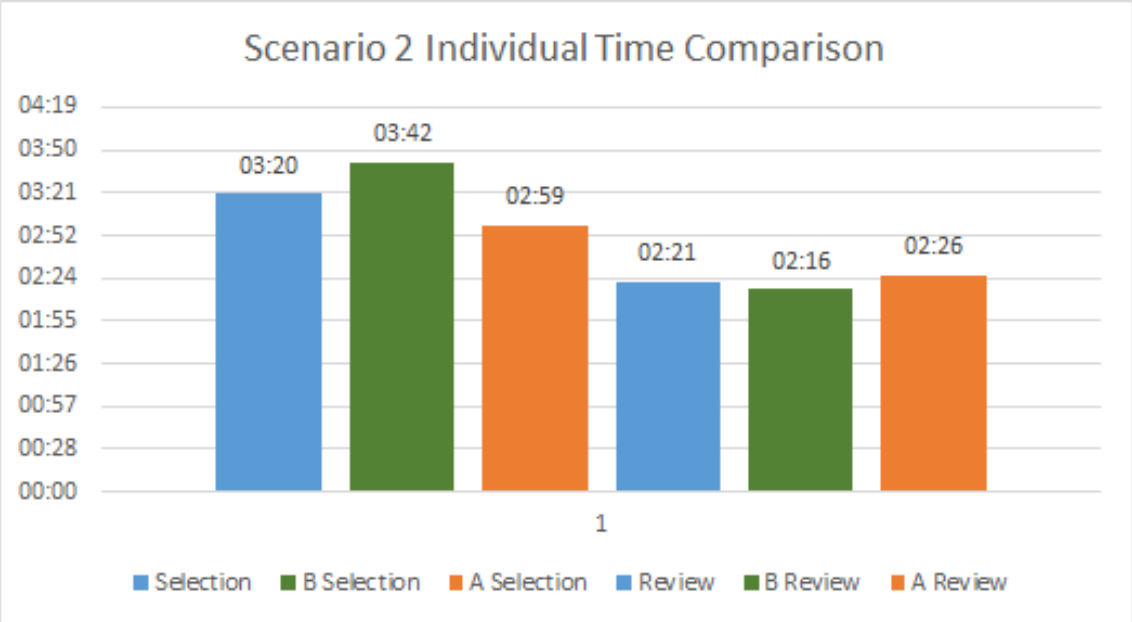


Figure 5.5: Scenario 2 Time Comparison (min)

Application is more advantageous than the Spider Map image that they were presented to, having 50% considered it very advantageous and 50% considered it simply advantageous.

It is important to consider, if any of the of A or B alternatives had a clear advantage over the other on the usability question by Scenario. However, the result show the same score for either scenario, in an interesting coincidence, with an average evaluation of 3.0, equivalent to "good". This means that, in this scenario, there was no clear influence in the outcome of this question depending on the interaction alternative the user was faced with.

However, the most interesting analysis comes with the preferred interaction alternative asked directly to the user. As it was explained previously, this statistic neared the 50% mark, giving no obvious advantage to either scenario. However, after separating the user groups and replacing the scenario selected by the interaction alternative (A and B), the results favour the alternative A, with 68.75% of users selecting it as the favourite. The B alternative was left with the remaining 31.25%. This statistic suggests that A was the best interaction, as this interpretation is now independent of the Scenario.

5.6 User Feedback

The user feedback to the application was overall very positive, being most of the test participants excited to interact with the application and to find out what it could do, as well as exploring the map as much as possible. One user mentions "I was fascinated, I had no ideal this bus went through here!", being pleased with the additional information gain. Another user was able to identify a familiar Public Transportation, saying "Oh, I usually go on this bus!". The experience

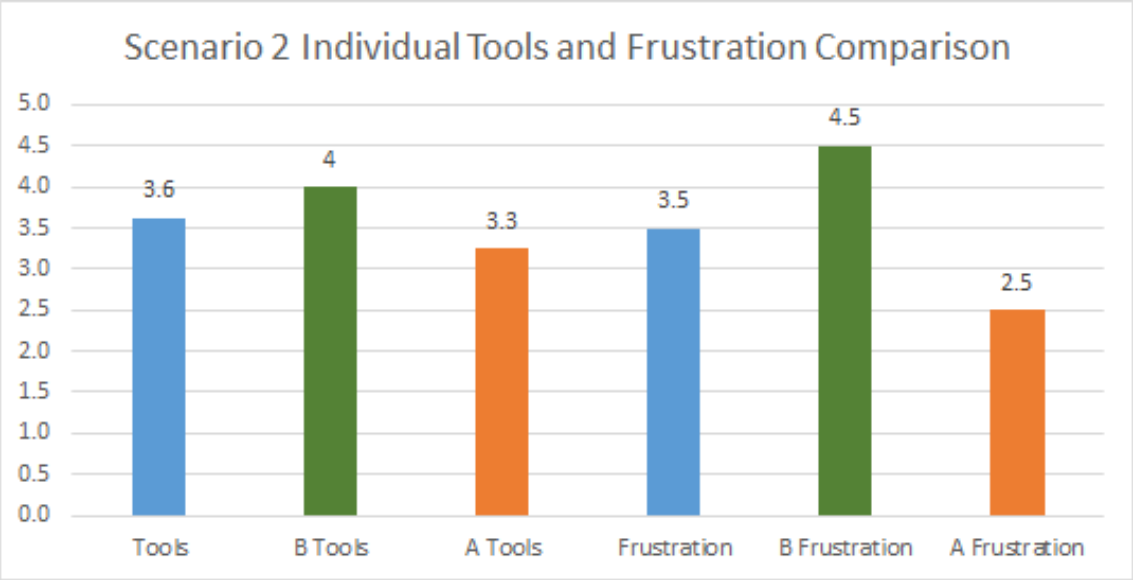


Figure 5.6: Scenario 2 Tools and Frustration Comparison

was challenging, but some users where pleased with the result, as one user mentioned "Complex, yet fun, amusing!".

However some aspects of the application can be improved. Some users experience frustration with some elements in the interface, namely on the "Hold to Fill" alternative, accidentally selecting an area when they were trying to drag the map. Moreover, a few users were unable to find the highlight feature on a stop, and only 43.75% use the search function. Some users dislike of the performance of the application on a complex map, saying "This can be a bit slow!".

Some interesting remarks also mention possible improvements to the application, further explained in the future work question, such as "The background map has to be behind to be able to understand." and "I would like to change the bus".

Analysing the importance of the application and why users would use it in the future, five users would mainly use it to find out where they can go from a given location, five others would use it to find a new location and three others to simply access and to know more about the transportation network around them.

There were many suggestions for the future work, some of the being related to unimplemented features from the Cobweb Solutions. The suggestions can be grouped by some categories, as enumerated bellow, ordered by number of suggesters:

- More services information (taxis, travel zones) - 5 Users
- Schedules - 4 Users
- Performance - 4 Users
- Points of interest 3 Users
- Additional side interface for more information - 3 users
- Ability to toggle lines inside the hub - 2 Users

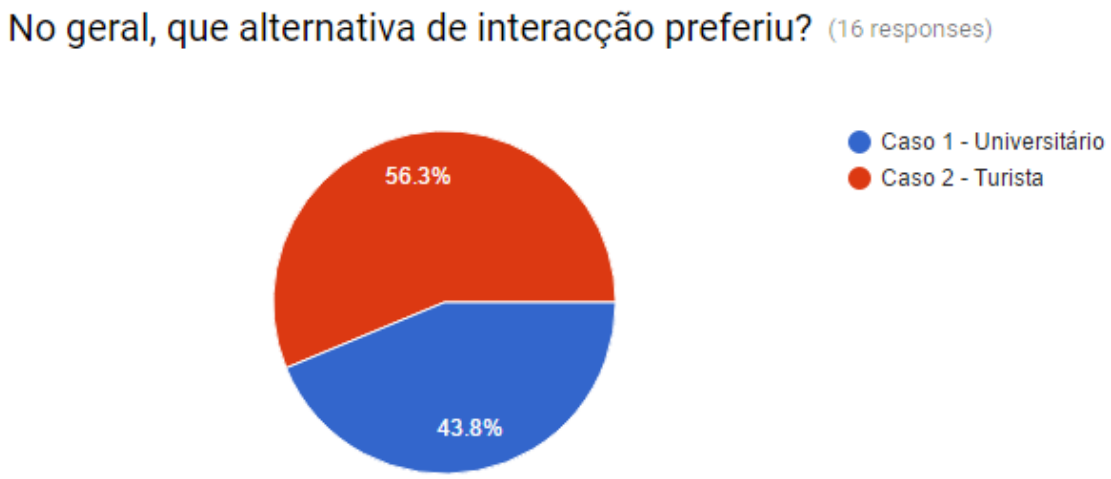


Figure 5.7: Preferred Interactions by Scenario

- Event location in real time - 1 User
- Line crossing (transferring from one bus to another) - 1 User
- Real time map generation by dragging finger - 1 User
- Different hub shape - 1 User
- Spider map at start and destination - 1 User
- Ability to share the map - 1 User

5.7 Conclusions

The test results are clear in what concerns to the general usefulness of the CobWeb 1.0 application and improvement in relation to the the Spider Map alternative.

On a metric level, the alternative A needs more time than alternative B, in both the Selection and Revision process. There was also more frustration in alternative A, but it correlates the usage of more features and exploration of the tools, which might also be the reason why the interaction took longer periods of time.

The user feedback concerning interactivity for each alternative was similar, while the information and usefulness favoured Scenario 2 for the context of the scenario. The background map feature was overwhelmingly found to be crucial for the application usage, providing a quick switch between the Schematic and Geographic map. There was a clear interaction preference in users, who chose the alternative A. User feedback was very positive, providing important ideas for future work, coincidently being some of them already contemplated in the Cobweb solution specified earlier.

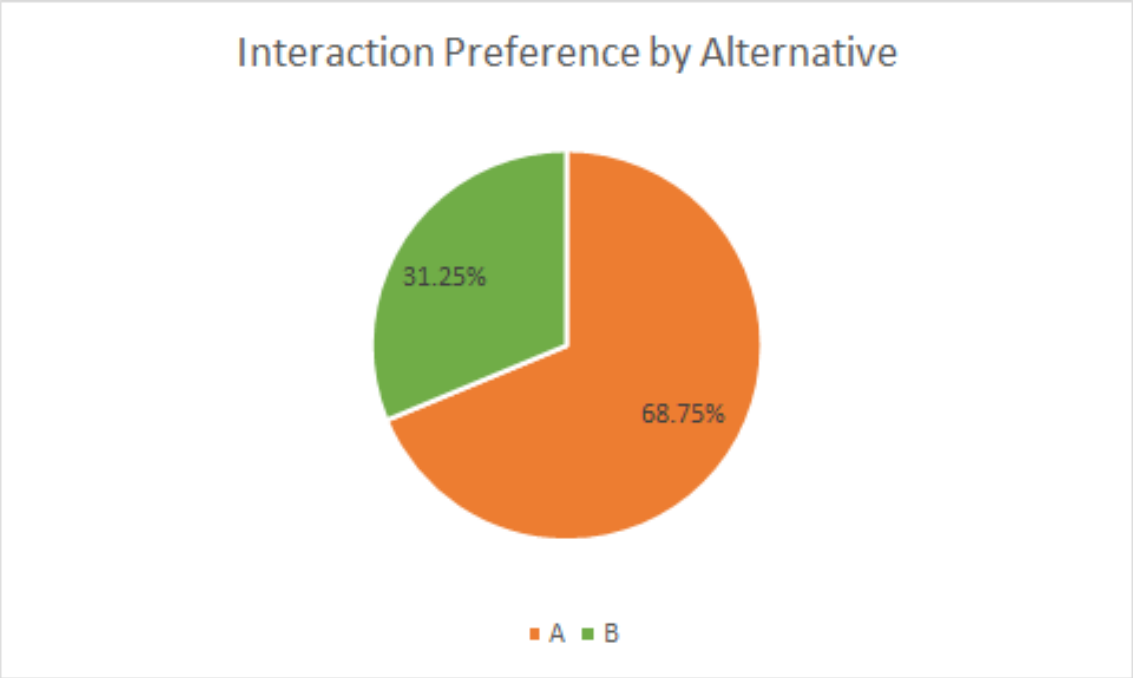


Figure 5.8: Preferred Interaction by Alternative

Chapter 6

Conclusions

Interactive Spider Maps look forward to change the way the traditional solution approaches the user, enhancing the interface and adding valuable, dynamic, customized context-aware information. The recent work developed around automated Spider Map generation and interaction opened room for the creation of a concrete solution and further implementation. The final research contribution of this adaptation creates innovation in the way technology approaches Public Transportation systems, proposing, implementing and validating a innovative solution.

The state revision of the art studies different approaches to the maps and Public Transportation systems, fuelled with interaction possibilities in mobile and touch devices, that allow the enhancement of such systems. The intertwining of the map concepts with technology translate to great exploration potentials, as is the case of Spider Maps.

Throughout this research, the proposal of the CobWeb solution and the underlying details proved to be important when taking into account the necessary adaptations to make to the traditional map, defining a solid concept to be implemented. This solution aims to solve a problem a step higher than the implementation, as it defines a baseline for further interactive maps and provides guidelines on architecture standards and data handling. The interaction process is divided in selection of the area of the map, generation of the map output and review of the resulting map. System scalability is also taken into account, as well as other requirements which followed different priorities.

The implementation developed provided a challenging experience, in order to adapt to the Solution, creating the CobWeb 1.0 web application, which includes a variety of features to answer the problem proposed. It was concluded that some alternatives of the interaction required further testing and the best way to do so was to dynamically implement different interactions, being modular enough to switch to others. The Porto area revealed to be a great testing location, having a plethora of Public Transportation services, although only some of them were chosen, to simplify the implementation. Open source map libraries were ideal for adaptation to this project, as well as other libraries that helped with additional tasks, further developed to achieve all the necessary

features. Highlighting features on the resulting map provided an important exploratory development, having the informational features of the Public Transportation services switch between the user focus in order to maximize the information gain against information noise in the map environment. Additional features which were not possible in a standard Spider Map, such as changing the geographic and schematic map enabled the contextual awareness of the application to increase dramatically. The modular nature of the application follows the core centres of data information, interaction, selection and review processes.

The testing process intends to evaluate the developed solution, defining a methodology for testing the application with users. The testing aims to minimize the impact of user lack of knowledge of the application and disparities in ability to interact with information by selecting a diverse group of 16 users, yet diversified enough to form a good sample. Two testing alternatives are proposed, related to the implementation differences, intercalated in order between two groups of users. The developed test contains two scenarios to compare the outputs using different metrics, as well as user feedback from questionnaire answers. A time measurement analysis favours the alternative B (simplistic features and touch related selection) in terms of speed, while also showing that users that take more time also use more features and struggle more with the interaction. The user feedback is very positive, favouring the application greatly against it's traditional counterpart, finding it to be a major improvement. The innovative feature of showing the background map was deemed as essential for the understanding of the context. Unknowingly, users ended up slightly favouring the second scenario of the tests possibly for being more comfortable to the overall application, but more importantly favoured the alternative A (complete information and customized selection process) of the interaction. User feedback shows concern on performance, but emphasizes the usefulness of the application information, opening a wide range of future improvement in different areas of the application.

In the future this work can be improved by testing the implementation of additional requirements contemplated in the Solution, which remained unhandled. Additional interaction alternatives described in the solution can be added and tested, as well as additional features such as information regarding the map and transportation services, times and schedules, points of interest and dynamic events around the map. The map generation can also be improved to real time using caching, improving performance. The barrier between the geographic and schematic map can also be thinned further by showing routes inside the Hub and improving highlight features to increase information gain. The solution can also be expanded by challenging the proposed solutions with new features which involve multiple map locations, and more intertwined route connections.

The work produced for this research has motivated the writing of a paper for the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC 2016) [its16], to be integrated in the 2nd International Workshop on Intelligent Public Transports under the topic of advanced traveller information systems using homogeneous/heterogeneous data sources.

References

[AB15] Michela Bertolotto Andrea Ballatore. Digital maps can be engineered to adapt to a person's unique interests and experience in geographic space. *COMMUNICATIONS OF THE ACM*, DOI:10.1145/2756546, December 2015.

[and16] Android - mobile operating system developed by google. https://www.android.com, June 2016.

[BDD+08] Howard Butler, Martin Daly, Allan Doyle, Sean Gillies, Tim Schaub, and Christopher Schmidt. The geojson format specification. *Rapport technique*, page 67, 2008.

[BRB+97] Bettina Berendt, Reinhold Rauh, Thomas Barkowsky, et al. Spatial thinking with geographic maps: An empirical sstudy. In *Herausforderungen an die Wissensorganisation (ISKO Tagung)*, pages 63–74. Citeseer, 1997.

[ÇHGF09] Arzu Çöltekin, Benedikt Heil, Simone Garlandini, and Sara Irina Fabrikant. Evaluating the effectiveness of interactive map interface designs: a case study integrating usability metrics with eye-movement analysis. *Cartography and Geographic Information Science*, 36(1):5–17, 2009.

[chr16] Google chrome browser. http://www.google.com/chrome, June 2016.

[CL11] Andre Charland and Brian Leroux. Mobile application development: web vs. native. *Communications of the ACM*, 54(5):49–53, 2011.

[dig16] Digitalocean - cloud hosting. https://www.digitalocean.com, June 2016.

[fL15] Transport for London. Bus spider maps, Novemeber 2015. https://tfl.gov.uk/maps_/bus-spider-maps.

[gal16] Samsung galaxy note 10.1 2014 edition. http://www.gsmarena.com/samsung_galaxy_note_10_1_(2
5677.php, June 2016.

[ges15] Touch gestures. http://maxcdn.thedesigninspiration.com/wp-content/uploads/2012/10/gesture-icons-004.png, Novemeber 2015.

[Goo15] Google. General transit feed specification. https://developers.google.com/transit/gtfs, Novemeber 2015.

[HS05] Mark Harrower and Benjamin Sheesley. Designing better map interfaces: A framework for panning and zooming. *Transactions in GIS*, 9(2):77–89, 2005.

[HW08] Mordechai Haklay and Patrick Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18, 2008.

[HWG03] Anders Hejlsberg, Scott Wiltamuth, and Peter Golde. *C# language specification*. Addison-Wesley Longman Publishing Co., Inc., 2003.

[its16] 2016 ieee 19th international conference on intelligent transportation systems. http://www.ieee-itsc2016.org, November 2016.

[Jen06] Bernhard Jenny. Geometric distortion of schematic network maps. *Bulletin of the Society of Cartographers*, 40(1):15–18, 2006.

[Joh10] David Johnson. Integrated passenger information: Delivering the rail end to end journey. Technical report, AECOM Limited, January 2010.

[Joh13] Jeff Johnson. *Designing with the mind in mind: simple guide to understanding user interface design guidelines*. Elsevier, 2013.

[KM99] Barry Keepence and Mike Mannion. Using patterns to model variability in product families. *IEEE software*, 16(4):102, 1999.

[KRR12] Jens Kolb, Benjamin Rudner, and Manfred Reichert. Towards gesture-based process modeling on multi-touch devices. In *Advanced Information Systems Engineering Workshops*, pages 280–293. Springer, 2012.

[lea16] Leaflet - an open-source javascript library for mobile-friendly interactive maps. http://leafletjs.com, June 2016.

[LM12] Jon Loeliger and Matthew McCullough. *Version Control with Git: Powerful tools and techniques for collaborative software development*. " O'Reilly Media, Inc.", 2012.

[LS87] Jill H Larkin and Herbert A Simon. Why a diagram is (sometimes) worth ten thousand words. *Cognitive science*, 11(1):65–100, 1987.

[map05] The age of exploration map. http://ranicaverzosa.50webs.com/age.jpg, 1705.

[Mas11] Mark Masse. *REST API design rulebook*. " O'Reilly Media, Inc.", 2011.

[MMWW07] Craig McMurtry, Marc Mercuri, Nigel Watling, and Matt Winkler. Windows communication foundation unleashed (wcf)(unleashed). 2007.

[Mon90] Mark Monmonier. Strategies for the visualization of geographic time-series data. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 27(1):30–45, 1990.

[Mou15] João Mourinho. *Automated Generation of Context-Aware Schematic Maps: Design, Modeling and Interaction*. PhD thesis, Faculty of Engineering of the University of Porto, February 2015.

[new16] Json.net - high-performance json framework for .net. http://www.newtonsoft.com/json, June 2016.

[NGJ+00] Fredrik Nyberg, Per Gustavsson, Lars Järup, Tom Bellander, Niklas Berglind, Robert Jakobsson, and Göran Pershagen. Urban air pollution and lung cancer in stockholm. *Epidemiology*, 11(5):487–495, 2000.

[ope16] opera browser. http://www.opera.com, June 2016.

[OPT16] Opt - optimização e planeamento de transportes, s.a. http://www.opt.pt, June 2016.

[Por07] Thomas Porathe. User-centered map design. In *usability professionals' association conference*, 2007.

[PRS02] J. Preece, Y. Rogers, and H. Sharp. *Interaction Design: Beyond Human-Computer Interaction*. Wiley, 2002.

[Ree08] Will Reese. Nginx: the high-performance web server and reverse proxy. *Linux Journal*, 2008(173):2, 2008.

[Rei04] Tumasch Reichenbacher. Adaptative visualisation for mobile users, 2004.

[RLHJ+99] Dave Raggett, Arnaud Le Hors, Ian Jacobs, et al. Html 4.01 specification. *W3C recommendation*, 24, 1999.

[SDKR09] Johannes Schöning, Florian Daiber, Antonio Krüger, and Michael Rohs. Using hands and feet to navigate and manipulate spatial data. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, pages 4663–4668. ACM, 2009.

[Shi62] Moshe Shimrat. Algorithm 112: position of point relative to polygon. *Communications of the ACM*, 5(8):434, 1962.

[SIR15] Service interface for real time information. http://www.siri.org.uk, November 2015.

[Smi15a] Aaron Smith. U.s. smartphone use in 2015. http://www.pewinternet.org/2015/04/01/us-smartphone-use-in-2015, April 2015.

[Smi15b] Patrick Smith. This physical accurate tube map will change the way you think about london. http://www.buzzfeed.com/patricksmith, September 2015.

[SS84] BP Shumaker and RW Sinnott. Astronomical computing: 1. computing under the open sky. 2. virtues of the haversine. *Sky and telescope*, 68:158–159, 1984.

[Sug69] Kohei Sugiura. Travel-time map. Shukan Asahi, 1969. Retrieved from http://3.bp.blogspot.com/_Fh2aCkgfh6Q/Sds5w-jaJZI/AAAAAAAAAN0/Gk0dF5b6eyM/s1600/Japan.jpg.

[TK05] Mike Lowry Taryn Kristof. Assessing the benefits of traveler and transportation information systems. Research Report 29, University of Washington, March 2005.

[Tor13] Marta Filipa Batista Torneiro. Study of the adaptation of spider maps to digital media (such as smartphones), from the interaction point of view. Master's thesis, Faculdade de Engenharia da Universidade do Porto, July 2013.

[tra97] Transmodel. http://www.transmodel.org/, 1997.

[VCW02] Michael P Verdi, Steven M Crooks, and David R White. Learning effects of print and digital geographic maps. *Journal of Research on Computing in Education*, 35(2):290–302, 2002.

[vis16] Visual studio - microsoft developer tools. https://www.visualstudio.com, June 2016.

[wam16] Wamp - a windows development environment. http://www.wampserver.com, June 2016.

REFERENCES

[WJH+02] S. Winter, T. Jung, B. Hessing, W. Nordsiek, O. Binnewies, T. Fabian, B. Rentel, D. Otte, and D. Tiemann. Method for operating a navigation system, October 29 2002. US Patent 6,473,691.

Appendix A

CobWeb 1.0 Screenshots

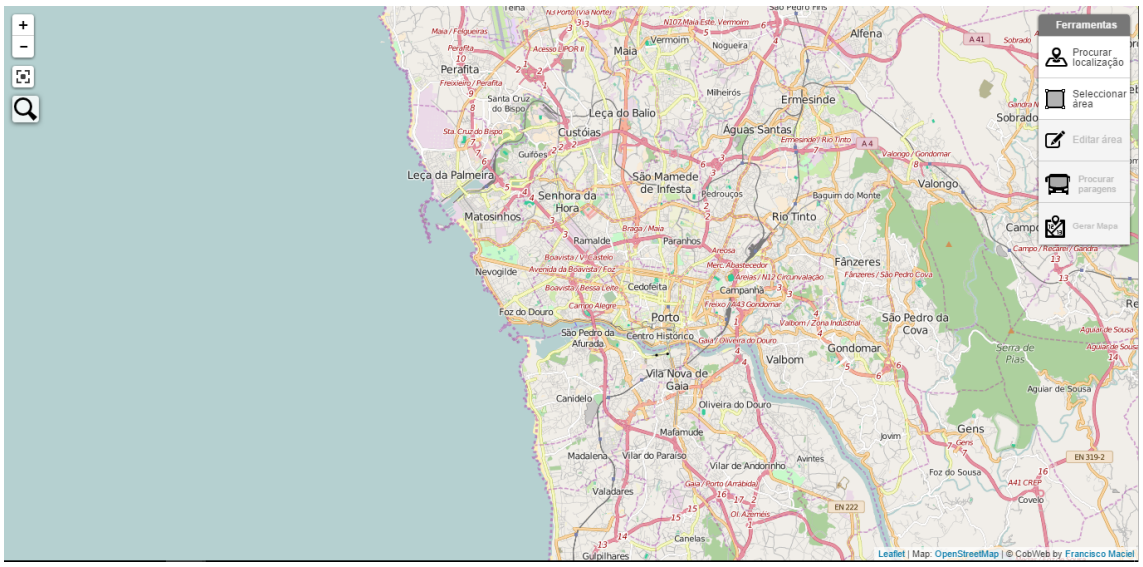


Figure A.1: Selection process example

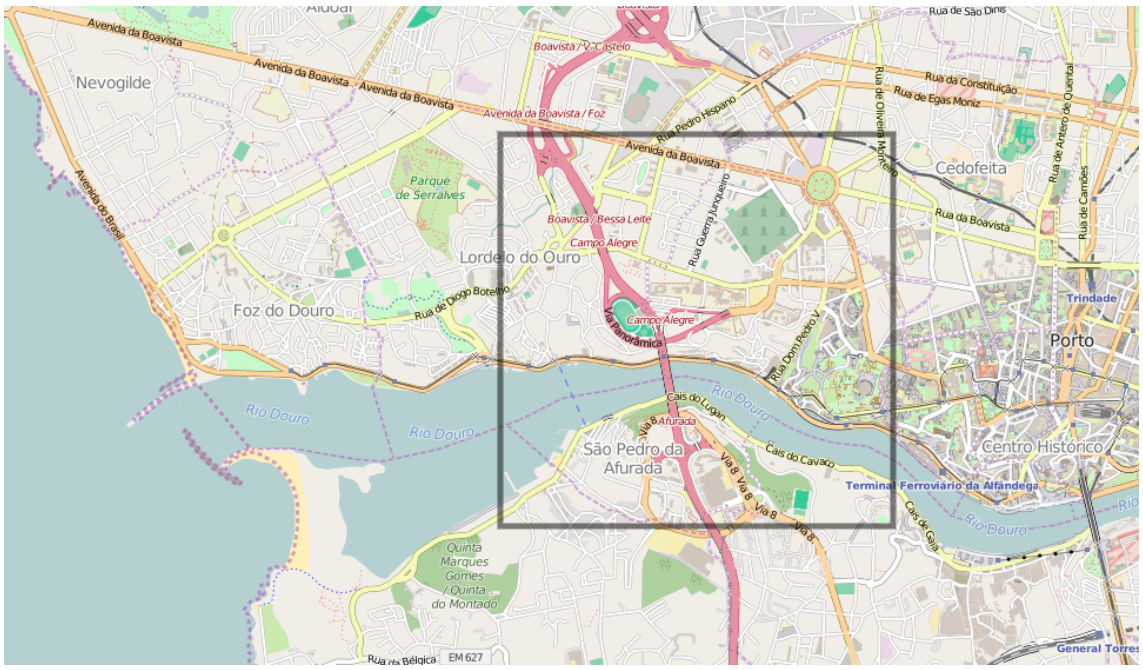


Figure A.2: Drawn Hub example

Figure A.3: Generation/Review Process Example

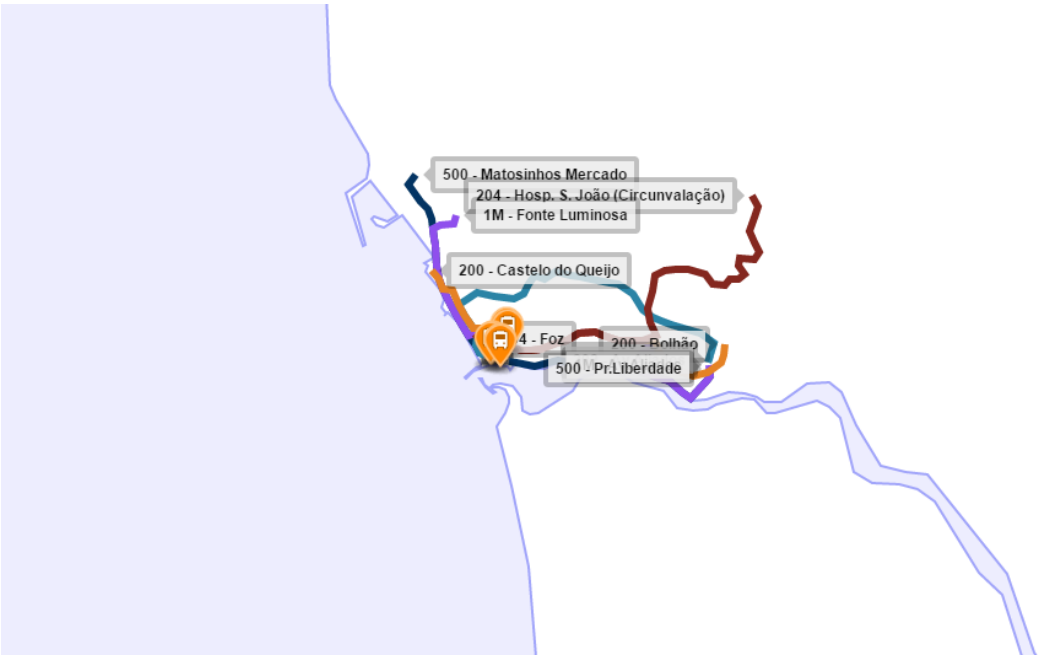


Figure A.4: River display example (darkened colour for visibility)

(a) Selection Process Interface

(b) Review Process Interface

Figure A.5: Side Interface

Figure A.6: Find My Location Feature

Figure A.7: Editing Hub Feature

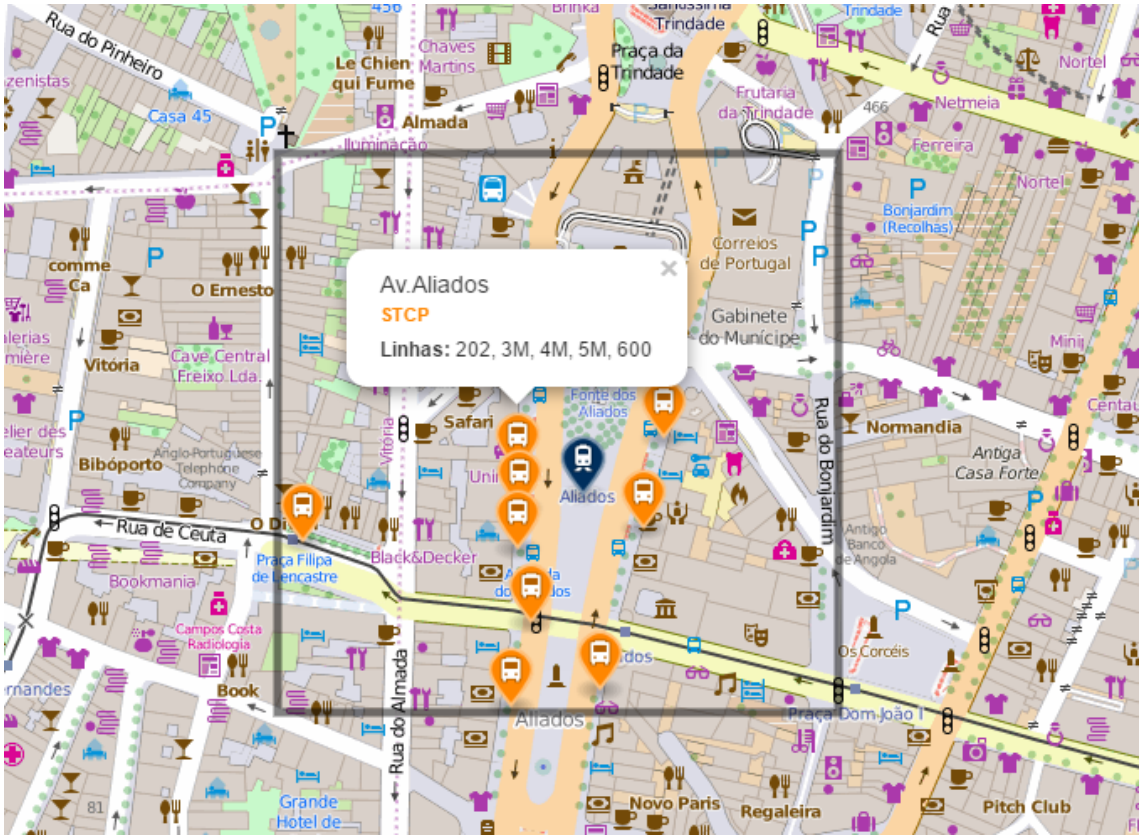


Figure A.8: Search for Stops Feature

Figure A.9: Generation Map Feature follow up Animation

Figure A.10: Converge on Hub Feature

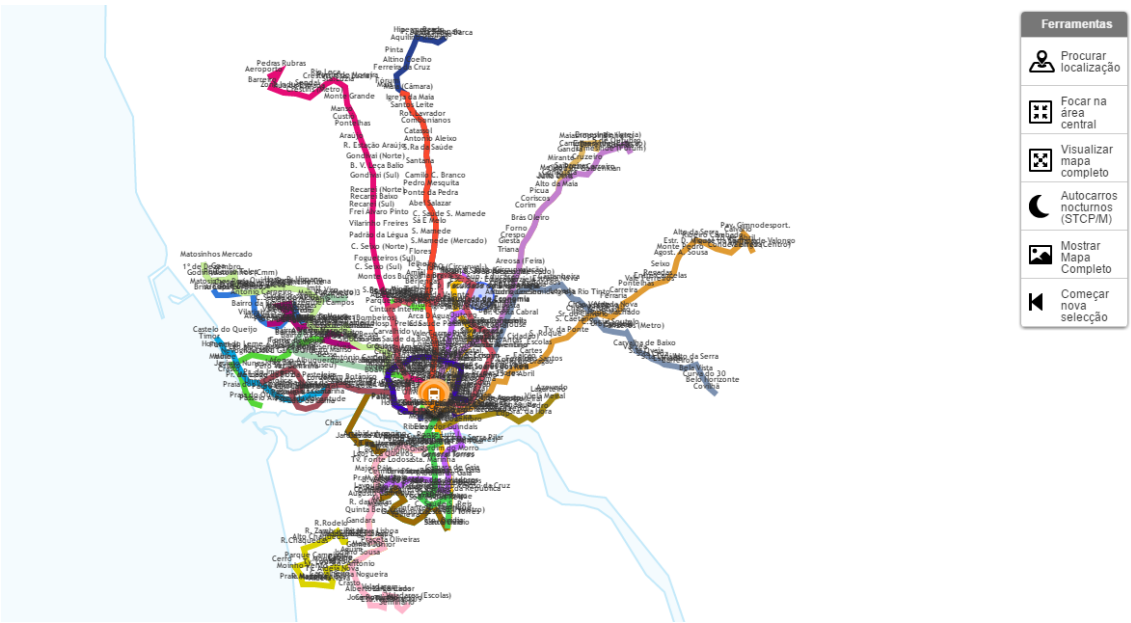


Figure A.11: Diverge from Hub Feature

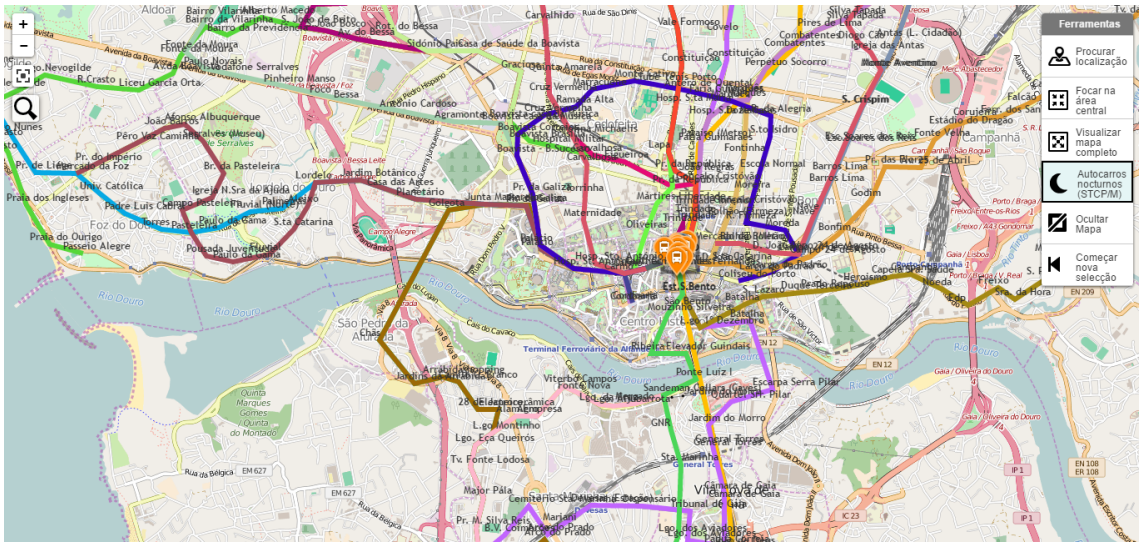


Figure A.12: Show Background Map Feature

Figure A.13: Plexus Testing Module

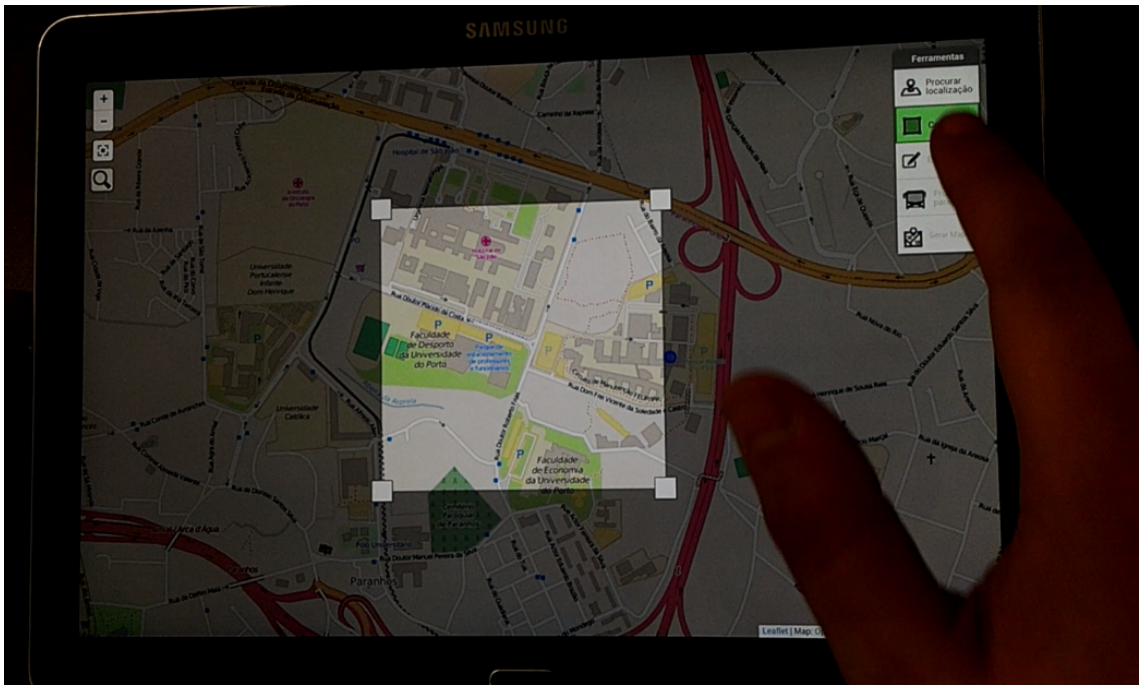


Figure A.14: Square Selection Alternative 1: Square Fixed

Figure A.15: Square Selection Alternative 2: Hold to Fill

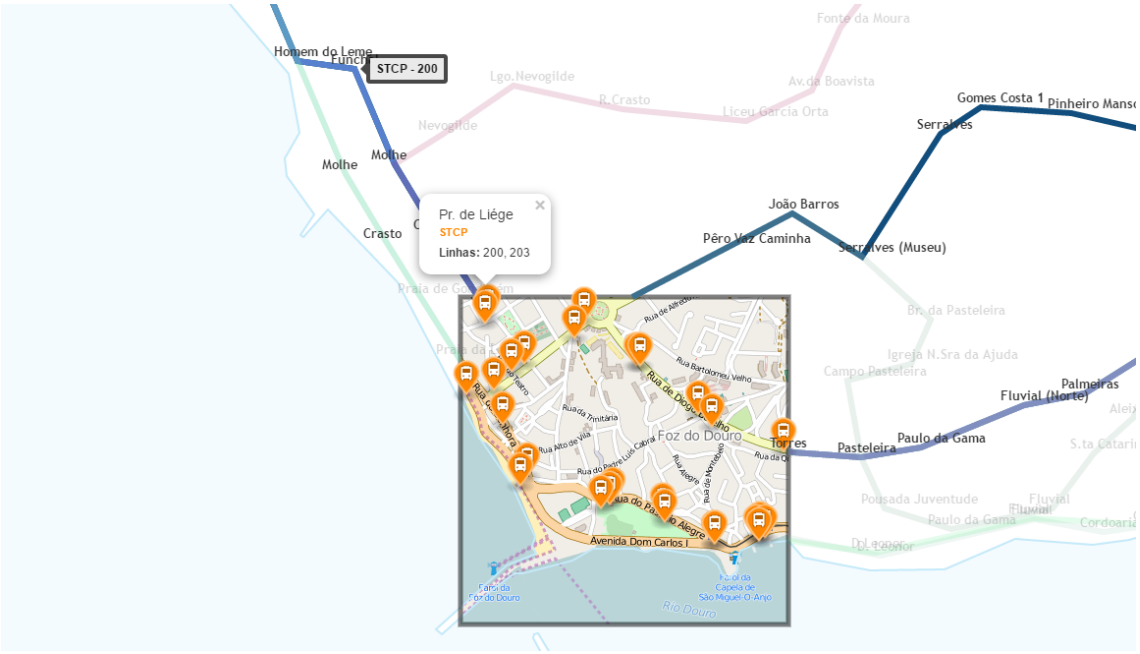


Figure A.16: Review Type Alternative 1: Complete

Figure A.17: Review Type Alternative 2: Minified

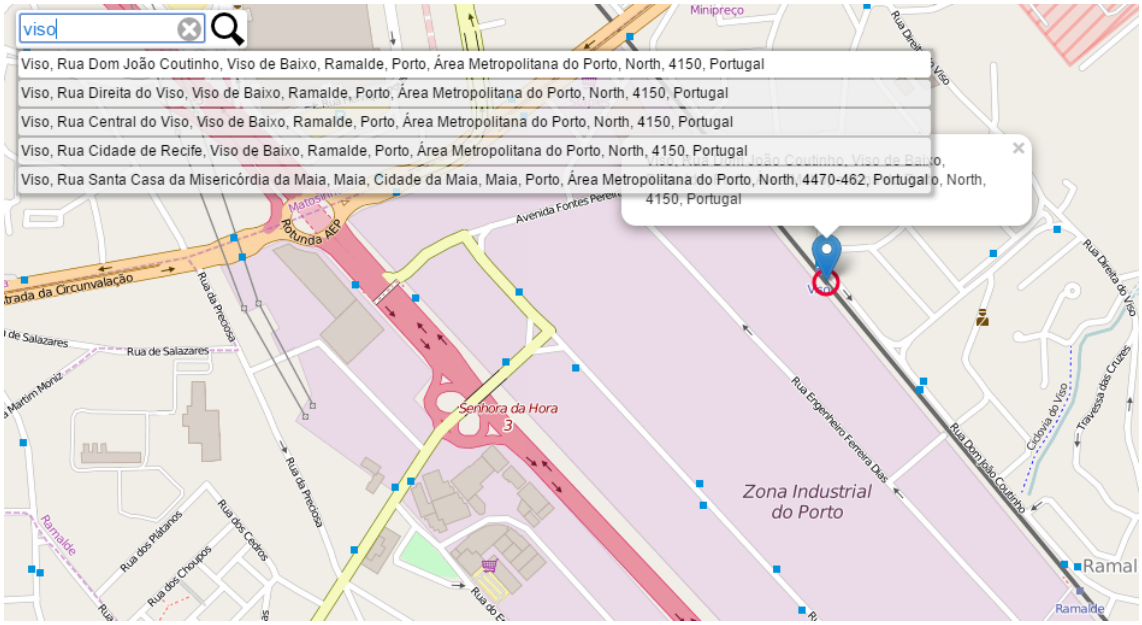


Figure A.18: Search Feature

Figure A.19: Night Lines feature (on)

Figure A.20: Night Lines feature (off)

Appendix B

Questionnaires

Teste da aplicação CobWeb

*Required

Descrição da aplicação CobWeb

Esta experiência consiste em testar uma aplicação web cujo objectivo principal é explorar a rede de transportes públicos da cidade do Porto e zona que a rodeia. Esta aplicação tem como principal vantagem a exploração da cobertura da rede em torno de uma determinada zona. A pergunta principal à qual se pretende dar resposta é: "A partir da zona onde estou (ou onde planeio estar) para onde posso ir, utilizando a Rede de Transportes Públicos?"

Os serviços de transporte presentes são a STCP, o Metro do Porto, a CP (Comboios de Portugal), a Resende e a EDG (Empresa de Transportes Gondomarense).

*

Por favor assinale o grau de familiaridade que detém com cada um destes serviços, e prossiga para começar.

	Nenhuma	Pouca	Alguma	Muita
CP (Comboios de Portugal)	○	○	○	○
Resende	○	○	○	○
STCP	○	○	○	○
Metro do Porto	○	○	○	○
EDG (Gondomarense)	○	○	○	○

BACK NEXT 40% complete

Never submit passwords through Google Forms.

Figure B.1: Form Part 1

Teste da aplicação CobWeb

*Required

Caso 1 - Universitário

No caso 1, encare o papel de um/uma estudante que irá entrar pela primeira vez na Universidade do Porto, ficando inscrito/a em uma das várias faculdades do Pólo da Asprela. O seu objectivo como estudante é perceber até onde poderá chegar utilizando os diversos transportes Públicos na área do Hospital de São João, tendo a possibilidade de se deslocar a pé num raio de cerca 1000 a 1500 metros a partir da zona da faculdade escolhida.
Poderá também assumir que vai alugar uma habitação nesta zona e deseja poder explorar a cidade nos tempos livres, ou mesmo voltar poder viajar até à localidade de origem.

Para avaliar esta actividade exploratória indique (quando terminar) três ou mais destinos que descobriu que pode alcançar, e o modo como se irá deslocar para os mesmos (incluindo o trajecto até à paragem e da paragem até ao destino).

Nota: pode explorar várias zonas perto do Pólo da Asprela

Destino 1 *

Your answer

*
Modo e/ou linhas utilizadas para chegar ao destino 1

Your answer

Destino 2 *

Your answer

*
Modo e/ou linhas utilizadas para chegar ao destino 2

Your answer

Destino 3 *

Your answer

*
Modo e/ou linhas utilizadas para chegar ao destino 3

Your answer

Destino 4

Figure B.2: Form Part 2

Teste da aplicação CobWeb

*Required

Caso 2 - Turista

Neste caso, encare o papel de um/uma turista que vem visitar a cidade do Porto e está à procura de alojamento. Como não planeia alugar carro, está preocupado/a com a cobertura de transportes Públicos na zona onde vai alugar casa. Dispõe de várias opções à sua escolha, não sendo o custo da habitação limitativo.

Está bastante indeciso e possui várias escolhas: Trindade, Baixa do Porto, Senhora da Hora, São Bento, etc ... . Está também a ponderar explorar outras zonas da periferia pelo que pode escolher outra zona qualquer do mapa que lhe seja familiar.

Explore pelo menos três possibilidades e indique , com base nas condições de acessibilidade de transportes públicos, qual acha que seria a melhor e a pior zona para alugar.

Enumere as zonas que explorou. *

Your answer

Melhor zona encontrada (em comparação) *
(E justificação)

Your answer

Pior zona encontrada (em comparação) *
(E justificação)

Your answer

BACK NEXT 80% complete

Never submit passwords through Google Forms.

Figure B.3: Form Part 3

Teste da aplicação CobWeb

*Required

Questionário

Terminado o teste, segue-se um pequeno questionário para avaliar a aplicação..

Em relação ao objectivo da aplicação: *

- Não compreendi
- Compreendi pouco
- Compreendi razoavelmente
- Compreendi

No caso de ter respondido "Não compreendi" ou "Compreendi pouco", quais foram as principais dificuldades?

Your answer

Como avalia a usabilidade da aplicação no Caso 1 - Universitário *

- Muito fraca
- Fraca
- Boa
- Muito boa

Como avalia a qualidade da informação no Caso 1 - Universitário? *

- Muito pouco relevante
- Pouco relevante
- Relevante
- Muito relevante

Figure B.4 Form Part 4

Pensa que a ferramenta foi útil para dar resposta aos objectivos do Caso 1 - Universitário? *

- Nada útil
- Pouco útil
- Útil
- Muito útil

Como avalia a usabilidade da aplicação no Caso 2 - Turista *

- Muito fraca
- Fraca
- Boa
- Muito boa

Como avalia a qualidade da informação no Caso 2 - Turista? *

- Muito pouco relevante
- Pouco relevante
- Relevante
- Muito relevante

Pensa que a ferramenta foi útil para dar resposta aos objectivos do Caso 2 - Turista? *

- Nada útil
- Pouco útil
- Útil
- Muito útil

No geral, que alternativa de interacção preferiu? *

- Caso 1 - Universitário
- Caso 2 - Turista

Figure B.5: Form Part 5

Em relação à funcionalidade de "Mostrar Mapa Completo" como avalia a importância desta alternativa para a compreensão do contexto? *

- Nada importante
- Pouco importante
- Algo importante
- Muito importante

Em qual dos casos foi a funcionalidade "Mostrar Mapa Completo" mais importante? *

- Caso 1 - Universitário
- Caso 2 - Turista
- Nenhum deles

Quão útil é, no geral, esta aplicação? * *

- Nada útil
- Pouco útil
- Razoávelmente útil
- Muito útil

Figure B.6: Form Part 6

Relativamente ao "Spider Map" que foi apresentado, quão vantajosa é a aplicação interactiva (nomeadamente no Caso 1 - Asprela)?

- ○ Nada vantajosa
- ○ Pouco vantajosa
- ○ Vantajosa
- ○ Muito vantajosa

Se sentir uma necessidade no futuro semelhante a um dos casos apresentados, utilizaria esta aplicação? *

- ○ Sim
- ○ Não

Qual o/s motivo/s principais pelos quais utilizaria/não utilizaria esta aplicação? *

Your answer

O que acrescentaria no futuro para tornar esta ferramenta melhor?

Figure B.7: Form Part 7

Appendix C

Additional Result Graphs

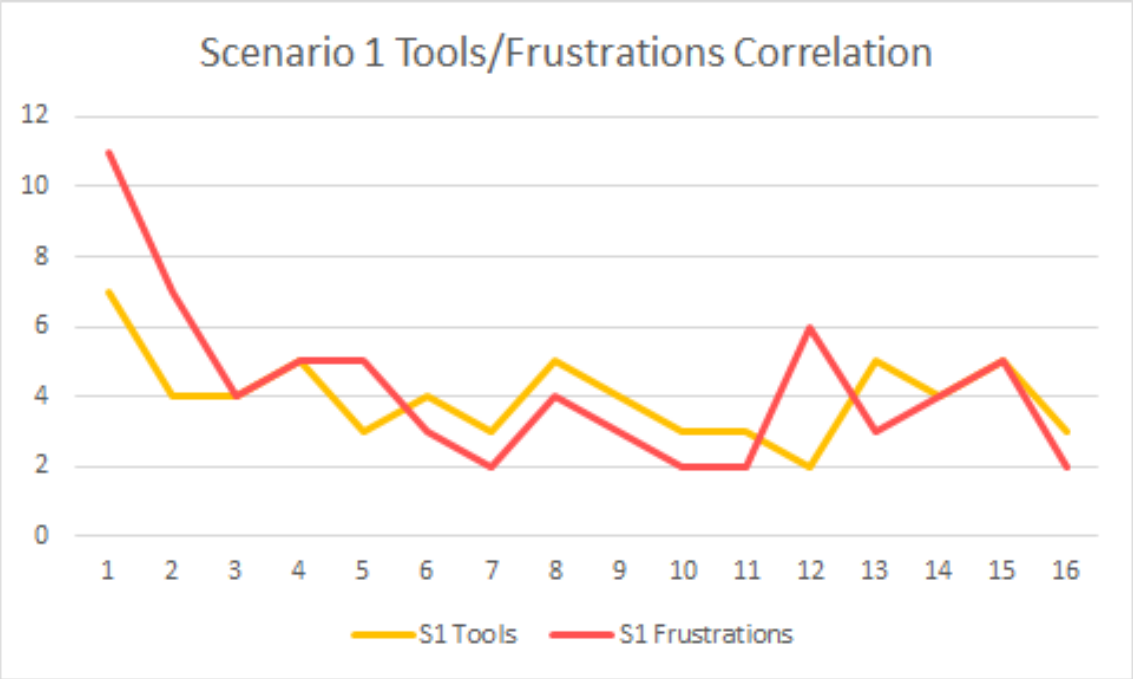


Figure C.1: Scenario 1 Frustrations and Correlations

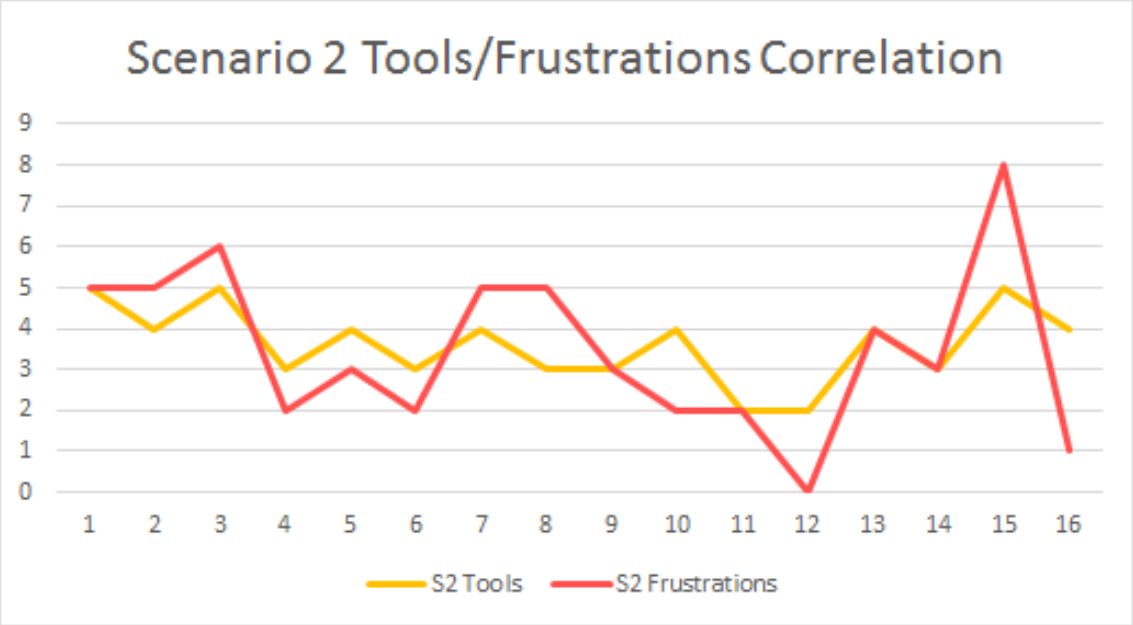


Figure C.2: Scenario 2 Frustrations and Correlations

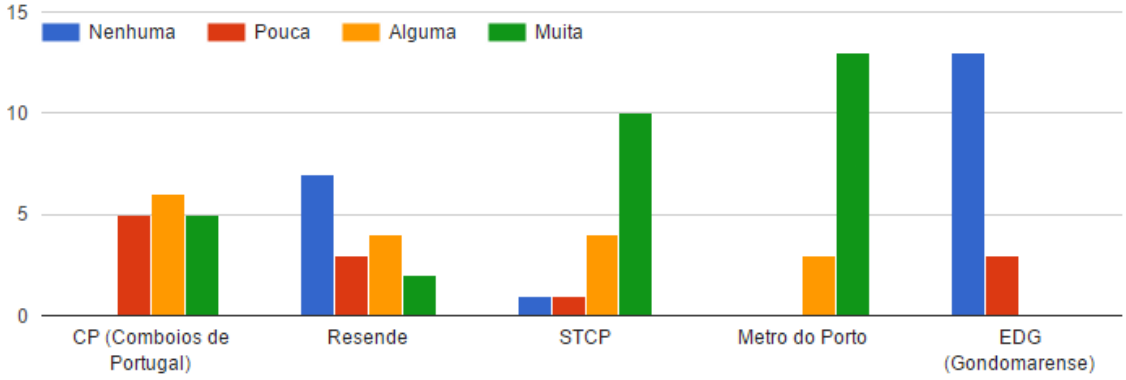


Figure C.3: Transportation Knowledge Responses

Figure C.4: Goal of the application Responses

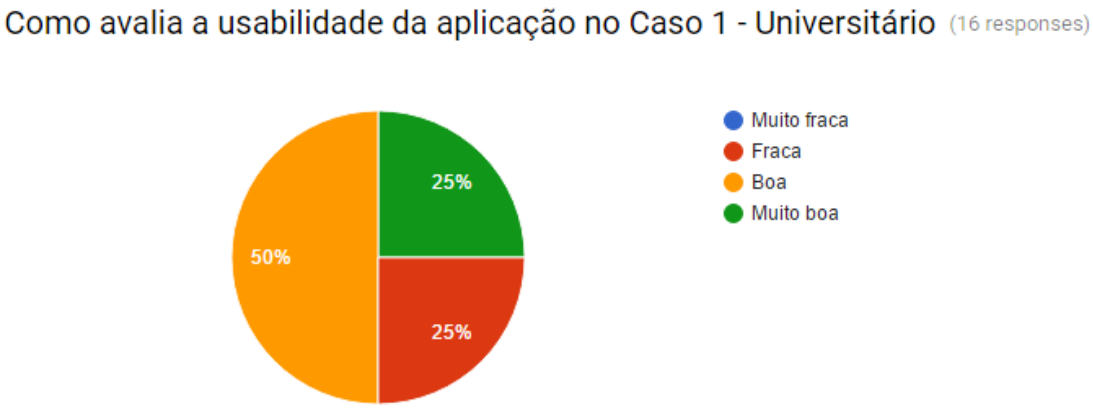


Figure C.5: Scenario 1 Usability Responses

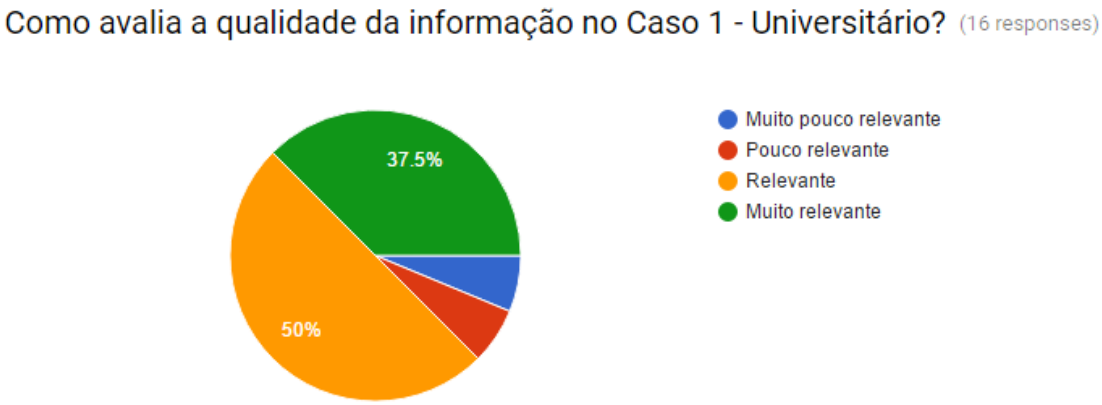


Figure C.6: Scenario 1 Information Responses

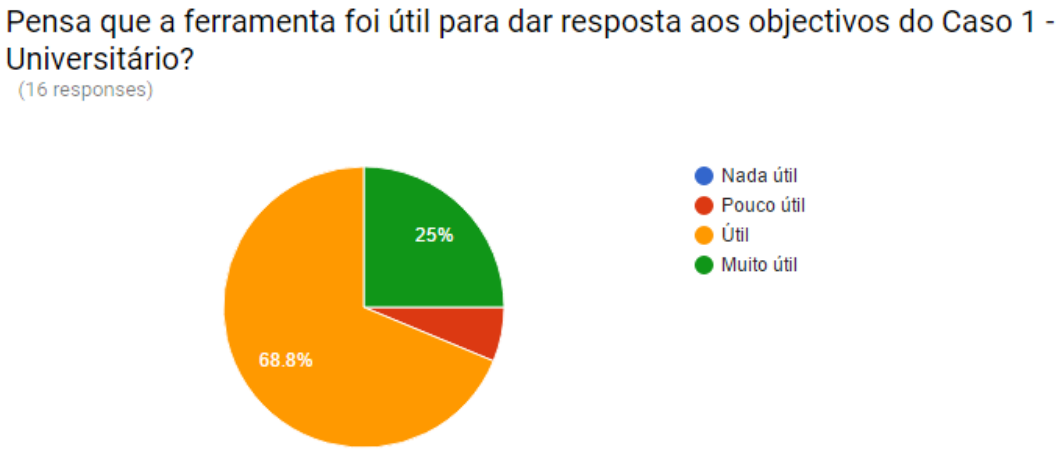


Figure C.7: Scenario 1 Usefulness Responses

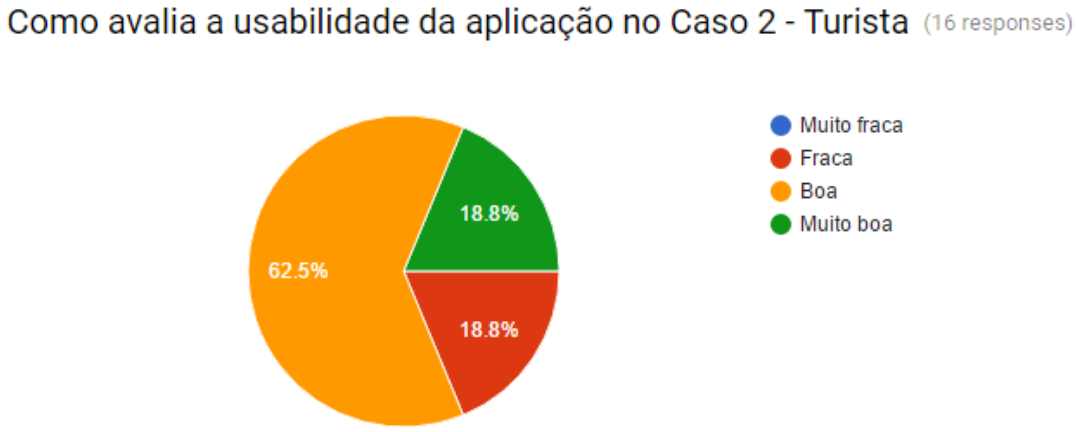


Figure C.8: Scenario 2 Usability Responses

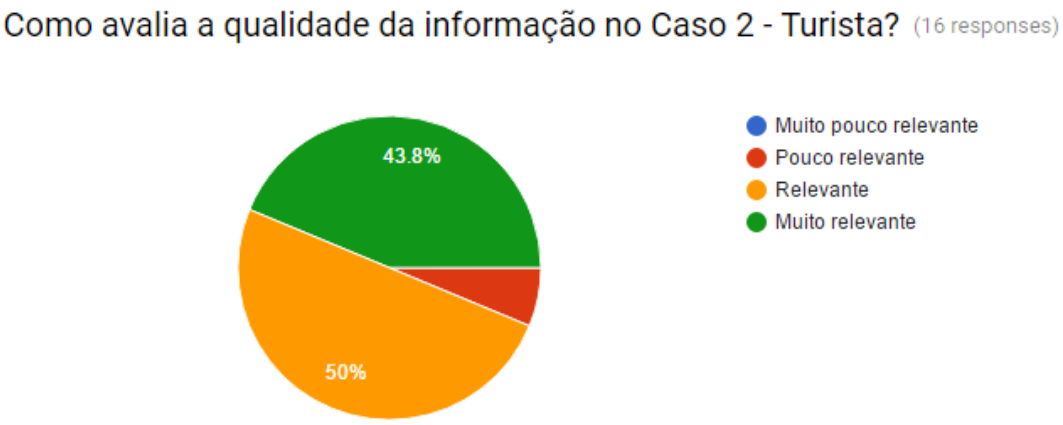


Figure C.9: Scenario 2 Information Responses

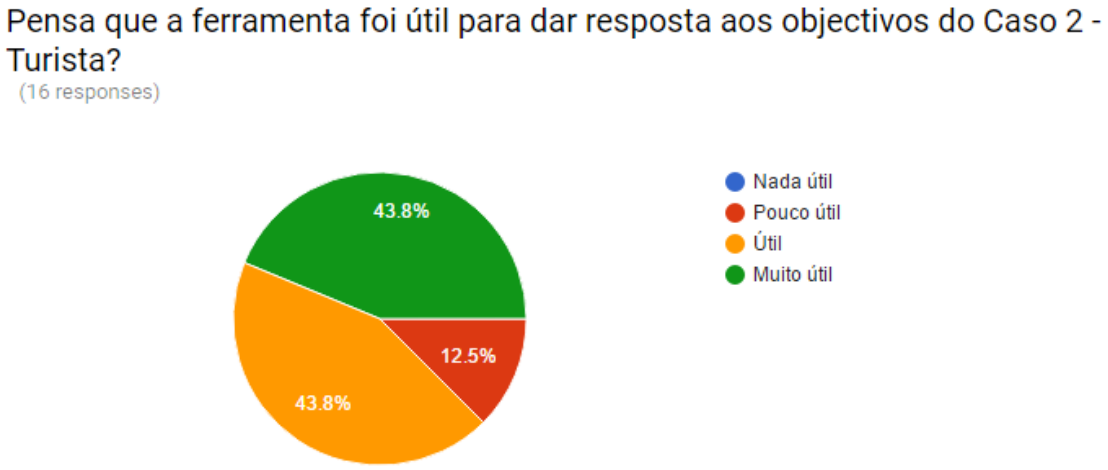


Figure C.10: Scenario 2 Usefulness Responses

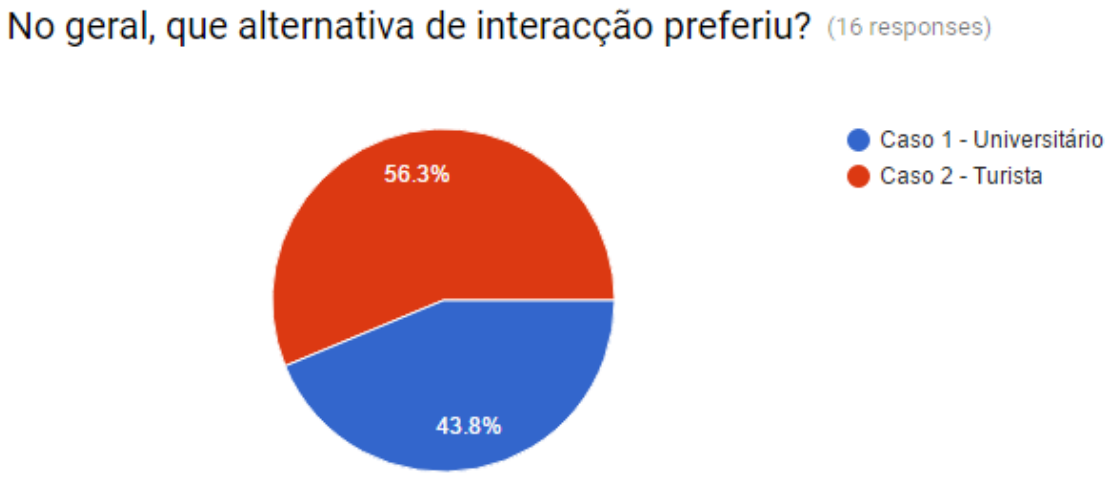


Figure C.11: Preferred Interactions Responses

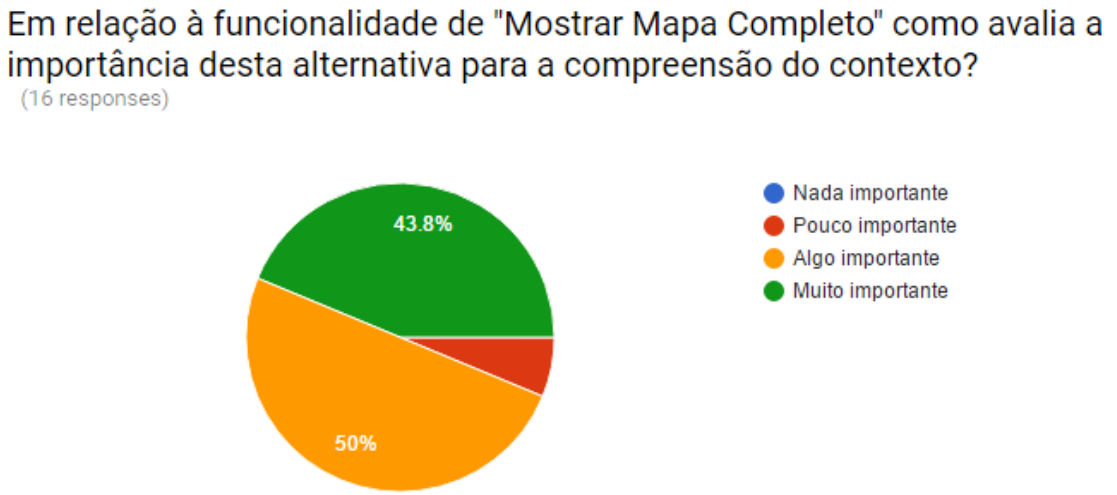


Figure C.12: Show Background Map Importance Responses

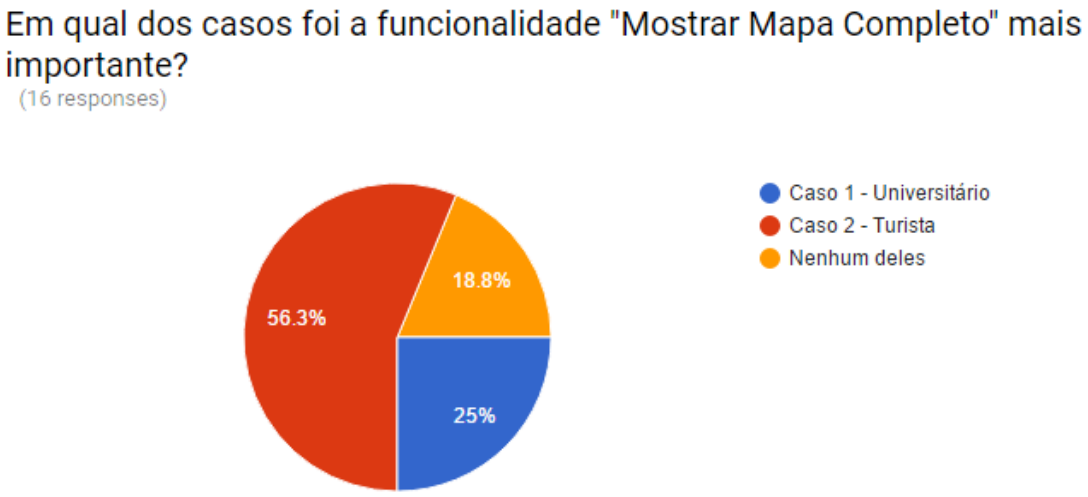


Figure C.13: Show Background Map Importance by Scenario Responses

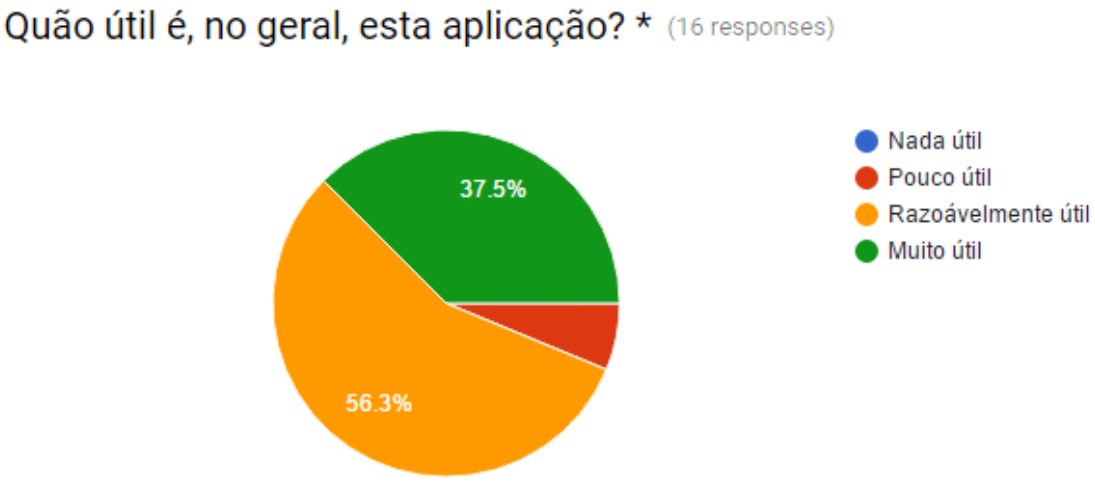


Figure C.14: Application Usefulness Responses

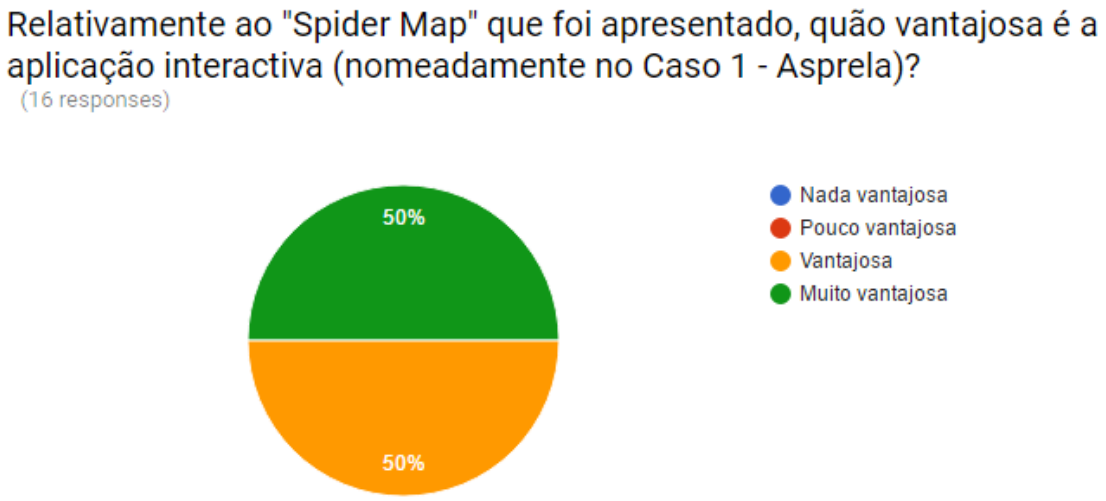


Figure C.15: Default Spider Map Comparison Responses

Figure C.16: Use in the Future Responses