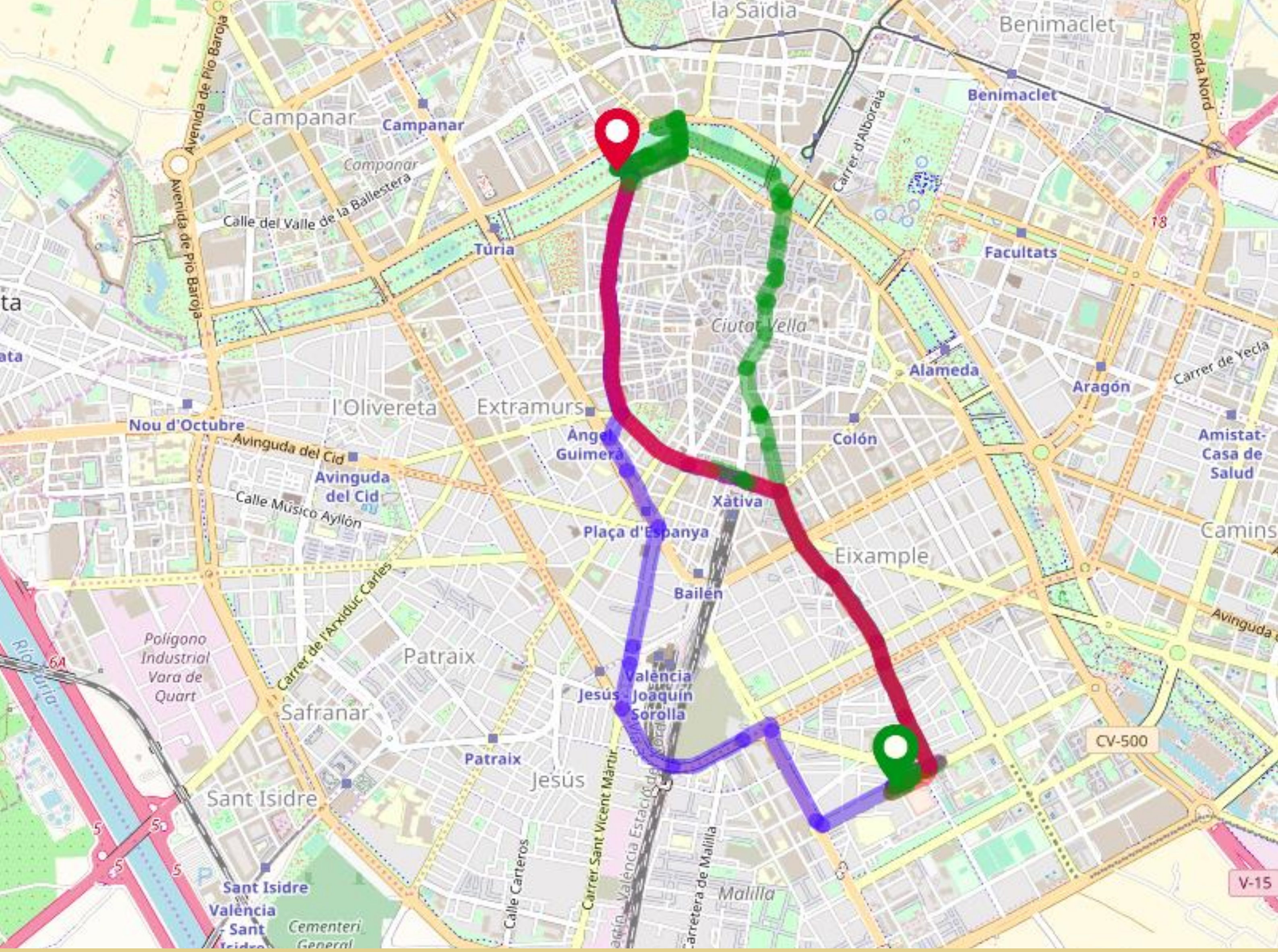


Gonçalo Daniel Almeida Santos

Door-to-Door Mobility as a Service

Dissertation/Internship Report
Master in Informatics Engineering
advised by Carlos Bento
and presented to the Departament Informatics Engineering
of the Faculty of Sciences and Technology of the University of Coimbra

July 2018

· U C ·

UNIVERSIDADE DE COIMBRA

Acronyms

BaaS Backend-as-a-Service.

DtD Door-to-Door.

IoT Internet of Things.

JWT JSON Web Token.

MaaS Mobility-as-a-Service.

OTP Open Trip Planner.

Abstract

As cities evolve, the need for mobility efficient solutions increases. The idea of better mobility is getting more attention from both transport users and city planners, with factors such as price, speed, comfort, sustainability and even health getting different priorities depending on the user. Therefore, a better awareness of the options should be provided by the cities administration, to increase the satisfaction of the community but also to increase the public transportation usage. This trend can help reduce traffic, reduce expenses on public transportation but also decrease the environmental impact of the transport network (both public and private means of transportation) on the city.

With this motivation, Ubiwhere considers itself capable of developing a Mobility-as-a-Service project, establishing connections with specific cities with innovation programs, gathering information on the public transport network and bikesharing programs, and developing a mobility service capable of aiding commuters, newcomers and tourists.

The objective of this project is to start the development process of the said product, implementing its core functionalities and provide proof of concept of the service, as well as to study what features should be implemented in the future, in order to certify that the final product is a service useful for transport users, city administration and transport companies.

In this document are documented both the steps took throughout the internship and the final analysis on the developed product, as well as the acknowledgements regarding fulfilled objectives and the future of the project.

The present report documents the study and product developed by the student Gonçalo Daniel Almeida Santos, in the context of the Internship in Software Engineering of the Department of Informatics Engineering of FCTUC.

Keywords

Mobility-as-a-Service, Door-to-Door, Transportation Services, Backend-as-a-Service, Public Transportation, Private Transportation, Last-Mile Transport

Resumo

À medida que as cidades evoluem, a necessidade por soluções de mobilidade eficientes cresce. A ideia de melhor mobilidade está cada vez mais a ganhar atenção tanto de utilizadores dos transportes como dos planeadores das cidades, com fatores como o preço, tempo dispendido, conforto, sustentabilidade e até saúde a ganharem diferentes prioridades, para diferentes utilizadores. Assim, e com vista a aumentar a satisfação da comunidade, mas também para aumentar a utilização dos transportes públicos, as cidades devem providenciar meios para os seus cidadãos conhecerem melhor as suas opções. Esta tendência pode ajudar a reduzir o trânsito nas cidades e as despesas com as empresas de transportes públicos, mas também reduzir o impato ambiental da rede de transportes (transportes públicos e privados) na cidade.

Com esta motivação, a Ubiwhere considera-se capaz de desenvolver um projeto de *Mobility-as-a-Service*, estabelecendo ligações com cidades especificas que têm programas de inovação, recolhendo dados da rede de transportes públicos e programas de *bikesharing*, e desenvolvendo uma serviço de mobilidade capaz de ajudar turistas, recém-chegados e pessoas que viajam diariamente na cidade.

O objetivo deste projeto é começar o processo de desenvolvimento do produto, implementando as suas funcionalidades principais e providenciar a prova de conceito do serviço, assim como estudar quais funcionalidades devem ser implementadas no futuro, de modo a certificar que o produto final é útil tanto para utilizadores de transportes, administração das cidades e para as companhias de transportes.

Neste documento está documentado tanto os passos seguidos durante o estágio como a análise do produto desenvolvido, assim como os pontos a reconhecer em relação aos objetivos cumpridos e o futuro do projeto.

O presente relatório documenta o estudo e o produto desenvolvido pelo estudante Gonçalo Daniel Almeida Santos, no contexto do estágio em Engenharia de Software do Departamento de Engenharia Informática da FCTUC.

Acknowledgements

The past year had a staggering amount of work, stress, and sleepless nights. Without any doubt, it also was more fulfilling that I could imagine, both for taking such an interesting project and for being surrounded by exceptional people.

Firstly, I must thank all the help and support I got at Ubiwhere. I couldn't ever desire more from a hosting company, as everyone proved helpful, patient and friendly, with incredible work conditions and understanding mindset. Without any doubt, this team shaped the kind of future team I want to be part of.

I would like to thank Professor Carlos Bento, for providing precious input on the project, being as much available as one would want and aiding with his great experience in multiple areas.

To André Duarte, an outstanding coworker, advisor and friend, I would like to express my sincere thanks. The impressive readiness, spirit, knowledge and dedication you show, even in the most tedious tasks, could encourage anyone.

I must also thank my friends for always being there and, on a special note, I must thank all my colleagues at NEI for the outstanding companionship.

To you, Jéssica, I must thank for your support, care, love and for being the best person I ever knew. Your efforts and patience allowed me to suppress all the obstacles, in hopes of a better future for both of us.

To my parents, I must deeply thank for being the cornerstone of everything I am. Your strength, understanding and teachings have always enlightened my life.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Nowadays a great part of the world population resides in city areas, due to the greater quality of life; this affects all branches of services to the city, from infrastructure to services provided. One of the most affected services, the transport one, scales in a different way from most: as cities get larger, their roads and routes can't get larger or provide additional throughput. This proves a problem, as most cities are split into suburbs and city centre, and therefore rely on the constant daily transit between those areas. In another scenario, one can notice the lack of effective transport services in the less urban areas, where transport routes are scarce and not suitable for the users' needs. The company Ubiwhere, as a smart city endorser and developer, aims to better solve the rupturing mobility problem in the cities and then enlarge the solution coverage for the suburban and rural areas.

In this work are presented the problems in the Mobility-as-a-Service (MaaS) area, the solution proposed by Ubiwhere, the architecture supporting this solution and the description of the supporting system. The system developed during the internship, a Backend-as-a-Service (BaaS), has the core features of the Ubiwhere's final solution, providing the proof-of-concept. The system, with its route planner, has the ability to suggest the user the best options based on the preferences (for comfort, speed and/or price), his/her preferred transport types, and available private transports (car and bike). The system is a truly Door-to-Door (DtD) system, taking into consideration the beginning and final parts of the trip (where the user usually walks or rides a bike). To fulfil its objective, the developed system uses the data of public transport, bike sharing and parking areas in the cities of Cologne (Germany) and Valencia (Spain), both cities where Ubiwhere has information on these systems, and a business connection to allow data exchange and a future product deployment.

This document details presents all the work done during the internship which occurred during the academic year 2017/2018, in the context of the Internship/Dissertation of Masters Degree in Informatics Engineering in the Software Engineering branch. The internship hosting company, Ubiwhere, is founded in Aveiro that focus on the development of intelligent systems for cities and state-of-the-art technology for the cities of the future. The internship took place at Ubiwhere's office at IPN, in Coimbra.

1.1 - Scope, Motivation and Opportunities

Mobility has always been present as a defining success factor in civilization, from the need to better transport systems and tools, like the wheel and the usage of steam locomotives, to the expansion of the low-cost airlines. Even today we can correlate the prosperity of a city to its transport systems, possibilities and opportunities provided to citizens. The transportation services became a huge part of the economic system, as the volume of merchandise and the number of people needing an efficient transport solution grew over the past de decades.

Some challenges have risen over the past years, thanks to this growth: according to World Health Organization, it is expected that, by 2020, at least 63% of the world population is located in urban areas, due to the greater quality of life and opportunities (job offers, social integration, etc.) compared to the rural ones. Therefore, the number of people needing transport in the cities (and even outside them) is constantly increasing, although cities can't restructure their roads or revolutionize their means of transportation without deeply changing their city planning.

Another challenge that appeared recently was the common ineffectiveness of the public transport in some cities. Some of the plans in action nowadays were made years ago, with minor changes through the years, and as city changes, the needs for each zone often drastically change too. This leads to a standoff situation: the transportation users need to get the service provided to them adapted, but the transport company needs to make a business valuable decision; one great example of this is the increasing number of smaller shuttles in cities, especially serving routes with fewer users, being more comfortable (increasing ride quality and satisfaction) and providing a more efficient service. Without an adapting, efficient and valuable (in terms of comfort and user satisfaction) transport network, the user could resort to private transportation, as seen in most of the medium cities, as a faster and cheaper option.

It should be kept in mind that the target of transportation services is not uniform, and discrepancies have increased: different users, travelling the same route should be able to choose between a low price trip, and a comfort-focused one (with consequently higher monetary value). To better understand this premise, and truthfully provide a solution service, one should understand the user needs and context. Taking the trips provided by aviation companies as an example, it is noticeable the market stratification between the users searching for a low-cost option, and those who prefer the high price option, due to its higher travel comfort, internet service provided, and food options. This stratification is also observable in longer train trips.

One potential misleading premise is that the time spent travelling is considered a waste of time and, keeping that in mind, trips should be as fast as possible, but recent advancements in transport comfort and options provided prove that, depending on the users point-of-view, the time spent travelling may not be completely useless. An important indicator is a relation between the perceived travel time and the total travel time. The perceived travel time is the amount of time the user can't use as if he wasn't on a trip, or, in other words, the amount of time actually wasted during the trip. The indicator is smaller on route options that meet the conditions for remote work (as a silent ambient and internet access), and proves handy comparing two route options with the same total travel time: the option with a perceived travel time is generally more expensive. Although this indicator can be assessed with extensive surveying work, it should prove useful when the user chooses between route option he's not acquainted with.

Taking a look at smaller trips occurring in a everyday basis, it is noticeable the increasing quantity of people ditching the usual means of transport and commuting in a bike: although with increasing effort, this option provides a healthier commute and diminishes the loss of value of the trip, as this may be considered a way to execute their daily exercise. So, the health beneficent presents as a motivator for walking or biking in the daily commute, gaining importance over the years in cities that make an effort to provide conditions for that to happen, like Perth, in Australia, which has a reported growth of biker commuters of 28% from 2006 to 2011, and a growth of 30% from 2011 to 2013 [1].

Last but not least, increasing the transportation system effectiveness is a step in the right direction to better pollution emission levels, considering it can reduce the number of private car ownership and usage. The sustainable mobility concept, while gaining increasing awareness during this decade and surely in the future, relies not only on public transport acceptance and vehicle pollution ban rules (like the currently being planned or introduced in several European cities), but also the investment and innovation in the zero-emission means of transport, like walking, bike sharing or private bike (and similar vehicles), as well as in the promotion of car sharing solutions (for both intracity or intercity trips). This mindset, of promoting the multiple means of environmentally friendly transport is considered one of the most effective ways to reduce the carbon footprint and promote environmental awareness.

Its hard to delivery a single transport solution that can fit all of those needs. Therefore, the user should always be able to choose the best suitable option for him, when possible. On this premise, Ubiwhere's solution to the people transportation problem is a multi-modal transport platform, that could prove suitable for everyone's needs, both for day-to-day trips and longer ones. This solution gives users the possibility to choose between different transports and changing between them on a trip, so they can travel in the best suiting way to reach his destination.

1.2 - Objectives

The internship project is divided in two phases:

- The first phase, took place during the first months, consisting mostly on **concept study** and **product concept establishment**.
- The second phase lasts the remaining of the internship duration and is reserved for the **product development**. The project consists of a **backend** application capable of suggesting the user a list of the best possible routes, based on his preferences and in the transport systems data available, taking in consideration the parameters refereed and explained later on this document.

Should be duly noted that the report elaboration took place during all the duration of both phases.

1.3 - Challenges

One of the biggest challenges of this project is the lack of data to effectively test and use the application. Both Ubiwhere and respective competitors struggle to find the information data with clearly specified routes, schedules and options of most transportation systems. It's observable the lack of disposition from transport companies to provide their data as application or servers, and that contributes to the lack of information. This constitutes a smaller problem in the Northern Europe, due to an earlier awareness and increasing concern about the ambient condition and mobility systems effectiveness, from both the government and transport operators in the countries of that area. Although this constitutes a problem for deploying the application as a product, Ubiwhere readily provided the information and data necessary for the application development and system testing.

The other concern, although not as defining, is the need for application components reusability, documentation and standardization, coming from the fact that, as a *backend* application, the product shall be integrated with other products, and is to be developed with great concerns on maintainability, quality and resilience.

1.4 - Document Structure

The Chapter 3 of this document describes the internship methodology, the development and work processes and the respective techniques, as well as the planning for the project for the whole timetable.

In Chapter 2, the results state of the art study are presented, with the explanation of the current developments of the MaaS concept in the global environment, as well as the direct and indirect competitors, their *modus operandi*, and their features. After this reflection, the Ubiwhere's concept is described, the product of the alignment of the desired features, the market needs and the viability concerns.

The Chapter 4 contains the product of the requirements specification process, using personas and *user stories*, to clearly specify the product features and modules associated with each.

In Chapter 5 are presented the results of the architecture study and decisions, and the way they provide a better solution. Also, in the same chapter, the technology study, results and decisions are presented.

Presented in Chapter 6 are the implementation decisions and most remarkable characteristics of the project. There can be consulted, in this chapter, both the system usage flow, both internal and external, as well as data models implemented, and their explanation. There is also given a brief overview of the frontend developed for testing and showcasing the project.

In Chapter 7, are presented the multiple tests implemented, to assure all the quality attributes are met. There can be consulted the results of the multiple tests implemented, and how they were idealised.

The Chapter 8 serves the purpose of explaining how the solution developed is applied in both testing and real-world scenarios, and how it performed in each one of them.

The Chapter 9 serves as an overview over the product developed, as it contains the internship product's correspondence to the personas and quality attributes, as well as a brief overview over the hosting company's expectations.

The conclusions of the internship are expressed in Chapter 10, where the intern expresses the word done in the internship, obstacles encountered, lessons learned and the future of the project.

Chapter 2

State of the Art

The results of the study conducted on the MaaS concept are expressed in this section, including the progress of the mobility-as-a-service concept, it's evolution through the years and future expectations. A small description of the concept of Backend as a Service is also provided, as the project stands as a product in the service category. In this section its also presented the available projects (being them competitors or products with another market in mind), their features, systems and operation modes. Both this researches allowed the posterior project conception, resulting in the desired product idea and the features to develop.

2.1 - Concept Development

The concept of intelligent information assistant, to deliver an integrating and intelligent solution containing tourism and travel systems, was first introduced in 1996, at an ENTER Conference, by Nico Tschanz and Hans-Dieter Zimmermann. Since then, the idea became popular, with the first concept being implemented in Gothenburg, Sweden, and gathering more attention as countries like Finland manifested the support from the government to companies working on the concept. Many more examples have arisen, with various base ideas, especially in more developed countries with a lack of space in cities or high environmental awareness.

One of the great examples is Japan: during the great expansion of cities, the parking space scarcity and increasing traffic jams increased, leaving no choice but to use public transportation; to answer the increasing affluence, JR East, the country's greatest railway company, implemented a "wallet card", allowing ticket-less public transport travelling. This option not only emphasized the use of public transport but also brought more comfort into the daily commute.

The concept lacked not only the governmental and company support but also the technological; there was a need for standardization for the transit and route information. As different companies and cities used various data formats, the usage of these was challenging. On 27 September 2006, Chris Harrelson (initial side project owner), Google's employee, released the format GTFS as a standard way of storing and save route information, allowing for the first time reasonable dataset usage.

Similarly, information exchange between sensors and devices, with the rise of Internet of Things (IoT), made even more important the standardization; across multiple makers, brands and models, the need for a common data modelling was notable. One of the initiatives, FIWARE Data Models, aims to model the data exchange from a wide variety of devices, like environment quality sensors, transport information, weather and parking spaces. This effort contributed not only to standardize the data but also as an incentive to open it to the public.

2.2 - European Organisations

This concept and its increasing awareness during the past years have contributed to the creation of multiple platforms and agencies, whose main purpose is to research the progress and study the viability of the MaaS concept. It should be duly noted that these agencies/platforms have been a great source of articles and researches to achieve the conclusions presented in this chapter. Those entities can be enunciated, along with their purpose and area, as follows:

- **MaaS Alliance** - A integration of multiple transport data into a mobility service. This company also provides multiple projects and documents on the MaaS subject, as well as supporting other multiple solution providers in multiple cities across Europe and North America;
- **ERTRAC** - The European Road Transport Research Advisory Council platform aims to bring together the transport stakeholders across Europe, in order to better suit their needs and wills in a common vision;
- **ERTICO ITS** - Europe's Intelligent Transportation System is an organization aiming to connect all the interventionists in the transport service, from authorities to private organizations, in order to promote research on the transport subject, as well as defining the industries standards;
- **POLIS** - This network, composed of multiple European cities, has the goal of developing technologies and policies for local transportation, addressing the multiple dimensions around the concept, being them economic, social, and environmental.

2.3 - Backend as a Service

The Backend as a Service is a Cloud Computing Service, which allows the development of applications and systems to be run in a different way from normal architectures. The system, acting as a middleware, allows the user applications to interact with the system using application programming interfaces (APIs) and software developers' kits (SDKs). This provides many advantages:

- The system communication (both request and response) are standardized and documented, leading to a better and easier front-end and user-level applications implementation (e.g. can be developed by a completely different team);
- The business logic can be encapsulated in the middleware;
- The services used by the middleware can be blocked from the user-level applications;
- The middleware independence from both external services and from the user-level applications leads to a better practice, providing higher cohesion and lower coupling.

The best examples of this cloud computing service model, are Parse, owned by Facebook, and Firebase, owned by Google. Both of them, oriented for mobile development of applications, specializing in different areas.

As for Parse, whose focus on allowing the user to implement custom code, and providing a real-time data sync using a real-time queries system, it is better suitable for real-time applications. Firebase's main purpose, on another hand, is to support the application with a model-observer scheme, hence being better used as a support for structured data.

2.4 - Golden Standards and Ubiwhere's solution

The solution to be developed should be based on the public interest, worldwide necessities and business needs. While those themes and analysis are too long and hard to be made within an internship, the internship's hosting company prepared some of the outlines for the project. Therefore, while the internships encapsulation is maintained, allowing a healthy research phase during the internship.

With all alternatives presented before kept in mind, there are many questions to be asked:

- What degree of innovation each one represents?
- What business model each one follows?
- Should the area of action be considered, or should be expected that all solutions will grow over time to a worldwide/high area action?

To provide a layer of stratification, the multiple competitors analysed were divided into five groups:

- **Community Based** - Blablacar
- **High Level** - Google Transit, Waynaut, eu-spirit, Goeuro
- **Region Focused** - Transport Direct, IDOS, Rejseplanen
- **Focused on Public Transport** - Moovit, Transit, CityMapper, Moovel
- **Full Products** - FluidHub, Whim

A full analysis of the multiple competitors is available in the Annex A. A comprehensive table, summing its features, can be consulted Annex B.

Based on those questions, Ubiwhere's approach is consistent: the developed solution should always be cemented with good development and stability, with enough interest from cities to be deployed and used by the public. With that in mind, the choice of the most interesting available solutions is based on how the solution approached the market, how the solution is growing (in terms of area of operation and public interest) and how the solution reaches the mobility interested community.

After the study on the available products/platforms, three golden standards were chosen, and below is presented an outline of them:

- **FluidHub** - The possibility to integrate all services in a city/region, to better suit its users, makes this platform a more effective product thanks to the commitment from all the transport entities available. Although not yet visible the importance and the usefulness of this platform, it is expected to grow in the upcoming years.
- **Google Transit** - This project, although not set for commercial viability, has the most available information from all analyzed competitors, with gathered data from the main transport companies from the most developed countries, allowing for a product more focused on bigger (and even international) trips.
- **Whim** - This region focused product, focus on a seamless travel inside a city. Although many other competitors achieve that, no other provides nowadays a product so well integrated along all levels, from single ticket multi-modal transport to the billing system.

A synthesis of the features of the available products/platforms considered golden standards is presented in the Table 2.1.

With both the concept and available products/platforms studied, with a greater focus on those considered golden standards, there could be assessed the developed product scope and features, with support from Ubiwhere's advisors, resulting in the conclusions presented in this section.

		FluidHub	Google Transit	Whim	Ubiwhere's Solution
Region		Worldwide	Worldwide	Helsinki	Worldwide^1
Transit and Delay Alerts		X	X		X^2
Routing Choice		X	X		X
Payment		X		X	X^2
Multimodal Transport	Foot	X	X		X
	Metro	X	X	X	X
	Train	X	X	X	X^2
	Bus	X	X	X	X
	Plane	X			
	Bike	X			X
	Private Car	X			X
	Car Sharing	X			X^2
	Taxi/Uber	X			X^2
	Rental Car	X		X	X^2
	Bike Sharing	X			X
Technological Support		Custom software for each city	Routing system using GTFS	Adaptative solution for a small number of cities	

Table 2.1: Comparison of the considered Golden Standards and Ubiwhere's concept

The Ubiwhere's solution column follows the following terminology:

- **Worldwide^1** - Although the solution is meant to be expandable one city at a time, with a process of gathering data, the internship's product development will focus on Cologne and Valencia, which provides a opportunity for a proof of concept, with the data gathering process already completed (to support the features focused on the internship).
- **X** - Feature to be implemented in the internship final product.
- **X^2** - Feature to be implemented in the future Ubiwhere's solution.

The internship's final product aim is to be developed in the city of Cologne, to provide better integration of their different solutions, as Ubiwhere already has projects developed in this urban centre. This allows for an easier gathering of data, allowing for a focused approach to providing a *proof of concept*.

The internship's final product features are the following:

- Routing System
- Multimodal Transport with the following transports:
  - Foot
  - Metro
  - Bus
- Private Car
- Private Bike
- Bike Sharing
- Car Sharing

One of the most important features implemented is the use of user-defined constraints to the route provided by the platform; it is important for the user to get the route with the most correspondence to his desires. This is the main feature that makes this system stand out: while the alternatives don't change their behaviour based on the user, our solution allows for high customisation and user satisfaction.

The complete features and requirements are available in the forthcoming Chapter 4.

Chapter 3

Methodology

In this section presents the methods followed and planning made during the internship.

3.1 - Process Management

Since the internship implied biweekly meetings with the advisors in the first semester, the methodology adopted was Scrum, providing a steady sequence in market and concept study, report writing, architectural and technology study, and product requirements specification, with a cyclic nature based on work done and feedback. After validating the results of each phase, there could be initiated the next one, if all the requirements are met.

In the second semester, mostly occupied with the development process, the adopted methodology was also Scrum, with sprints on a weekly basis. This week cycle allows for the intern to gain autonomy but sustains the controlled environment, based on the feedback from the advisors, and the Scrum meetings at the end of each cycle allow for the intern to report on work done, his needs and the problems encountered. The Figure 3.1 briefly explains the process.

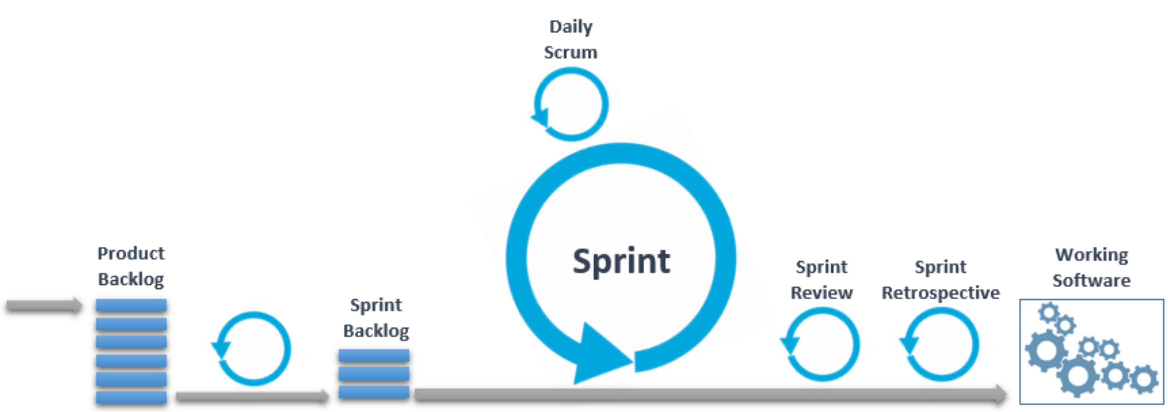


Figure 3.1: Scrum Diagram

Another advantage of this sprint-based approach is to more easily steer the development according to the User Stories prioritization (functional requirements), which is explained in Chapter 4.

To manage the tasks and log the times, the platforms Redmine and UIS were used. Those platforms, hosted by Ubiwhere, are used by both employees and interns.

During the development process, the weekly meetings with the company's advisor allowed the intern to adjust the expectations, gather feedback on implementation decisions (being them simple or complex and project defining ones), as well to as plan the upcoming weeks' work.

3.2 - First Semester Plan

Following is presented the planning for the first semester, which is mainly dedicated to concept study, architecture analysis and requirements specification.

- **Intermediate Report:**
  - **Concept Study** - Study of MaaS concept and developments: the products and platforms available, the worldwide progress in the mobility area and the opportunities to take.
  - **Report Writing** - Writing the intermediate report with orientation from both advisors.
  - **Intermediate Report Delivery**: 2 January 2018
- **Requirements Specification:**
  - **User Stories** - The work conducted specifying the requirements and producing the Chapter 4.
  - **Requirements Approval**- Approval from the requirements specification, from both advisors and necessary entities in Ubiwhere.
  - **Requirements Milestone**: 22 November 2017
- **Architecture e Technology:**
  - **Architecture Study** - Consisted of the study of the available architectural options, and choosing process.
  - **Technology Research** - Consisted of the study of the available technological options, and choosing process.
- **Overview:**
  - **Technological Overview** - This phase was dedicated to adopting to the technology, allowing better a better awareness of good coding practices.

3.3 - Second Semester Plan

Following is presented the planning for the second semester, which is dedicated to the solution development and testing.

- **Product Development**
  - **Development** - Phase dedicated for development of the product.
    - **Pollers** - Development of the components that retrieve information from cities APIs and external components.
    - **Achitecture Setup** - Establishment of the communication between the multiple components of the architecture.
    - **Realtime Tracing** - The component of real-time route tracing, that was integrated with another project.
    - **Car Sharing** - The component of car sharing, that was integrated with another project.
    - **Route planner** - The main component developed in the project, and the bigger time syphon.
    - **Showcase Frontend** - A simple web page to allow better testing and concept proof, in a later project phase.
    - **Other components** - A collection of simpler components that support the project, with information like projects and so on.
  - **Refactor** - The project reiteration phase should include alterations to the developed product, with both experience gained from the development as well as the feedback from Ubiwhere's responsible employees.
  - **Testing** - To test the product developed, including solving problems that may be revealed.
- **Final Report**
  - **Report Writing** - Writing the final report with orientation from both advisors, including the refactoring of the intermediate report.
  - **Final Report Delivery:** 2 July 2018

3.4 - Gantt Diagram

In Figure 3.2 and Figure 3.3, are presented the Gantt Diagrams, resuming the plan of the internship. It is presented with a basic time period of a week, for a broader overview, and also include the milestones stipulated above. As the diagram was updated over the time, to allow both the intern and advisors the assessment of the state of the internship process, as well as to better illustrate the completed and remaining tasks.

Over time, the Expected Gantt evolved into the Updated Gantt Diagram, and the differences can easily be noticed, specially regarding the duration of the development of the components.

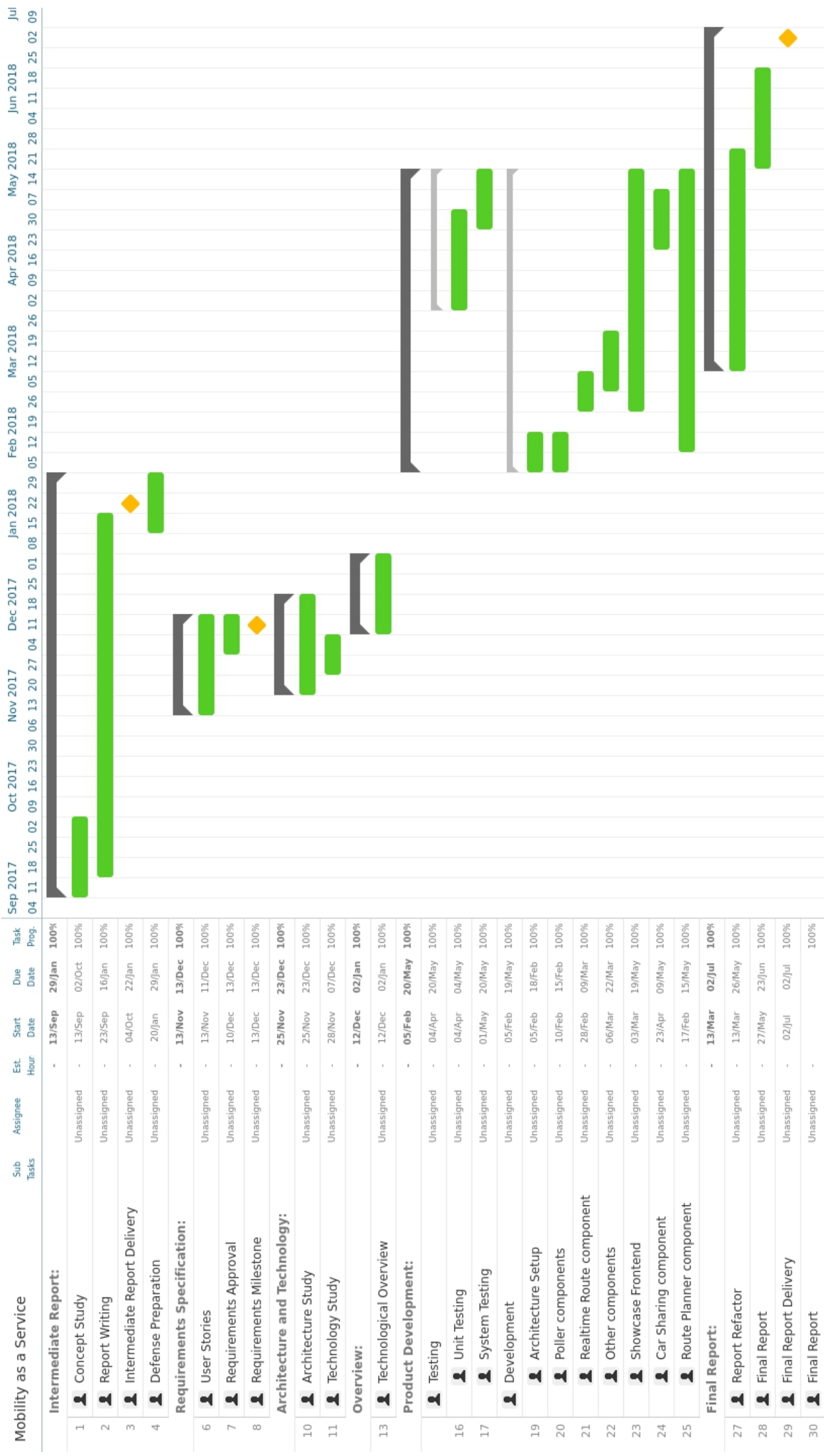


Figure 3.2: Expected Gantt Chart of the internship

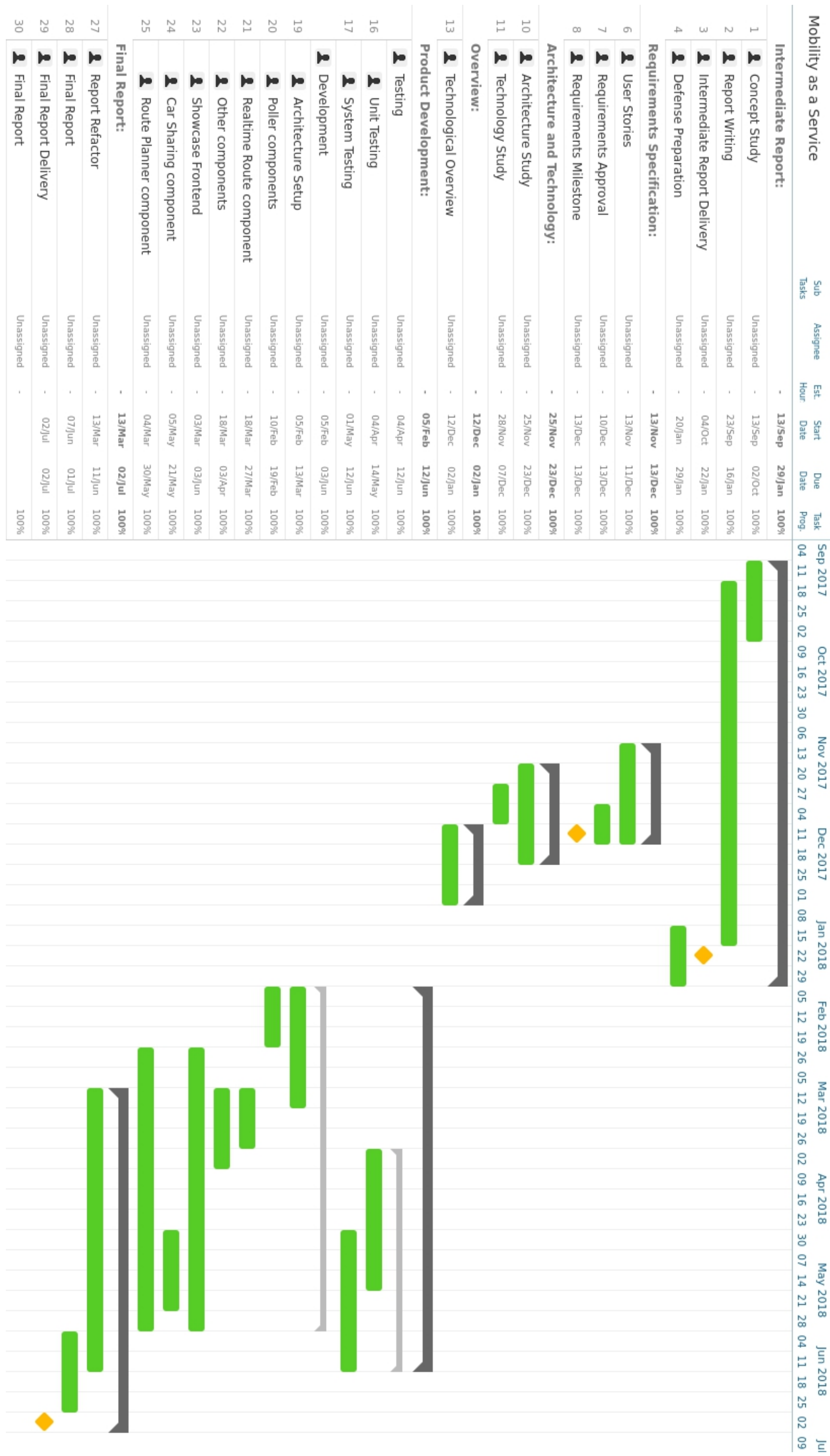


Figure 3.3: Updated Gantt Chart of the internship

3.5 - Internship Success Criteria

The success criteria represents the agreement between the intern and the hosting company intervenients, respecting the guidelines imposed for the internship. Therefore, the following points represent this agreement:

- All architecture and requirements are approved by the entities responsible at Ubiwhere.
- The requirements classified as must-have, that can be consulted in Chapter 4, must be implemented.
- The developed product satisfies all the imposed quality attributes, stated in Chapter 4.
- Both milestones for the project (intermediate and final) are delivered on time, without requiring spending more than the proposed hours.

3.6 - Risks Management

In order to increase the success rate of the project, there must be given attention to all factors that could affect its course in a negative way. By making a risk analysis, selecting each possible risk, as well as its own probability and impact in the project, one can also create a mitigation plan to be put in action when and if necessary.

In Table 3.1 is displayed the syntheses of the risk management process. It should be noted that this process is iterative, and should be complemented during the internship. The probability range is from 1 to 5, being 1 the least probable and 5 the most probable (considered almost certain). The impact range is from 1 to 5, behind 1 of minimal impact, and 5 of catastrophic proportions to the project.

Type	Risk	Probability	Impact	Mitigation Plan
Technical	Internal or external services unavailability	1	3	Application of error reporting system and redundancy techniques. When possible, promote use of internal components.
Technical	Development Delay/Setback	2	4	Revise the plan, evaluate the estimatives and, as a last resort, adapt requirements and/or scope.
Technical	Difficulty in adoptance of the technologies or usage of necessary components for the project.	1	3	Dedicate more time to the training, and require more advisoment in the development process.
Technical	Requirements chang/reavaluation	2	3	Reavaluate the requirements, keeping the final product both valuable for the hosting company and suitable for the available timeframe.
Technical	No time available for the completion of all the internship required processes	3	5	Reavaluate the intership plan, and ensure that all the compromises of the internship are met.
Technical/Business	Insuficient data for development of the project	2	3	Require from Ubiwhere the data necessary to implement and test the product. As a backup plan, mock the data so that the product can be implemented and tested.
Business	No cooperation from transport entities in a given city	2	3	Pressure the local management the pressure to ensure that transport information is available in a suitable way.
Business	No interest from cities in the final product	2	5	Change the scope of the project, to better reflect the interests of the city. Iterate the requirements and the development process.

Table 3.1: Risk Management Analysis

From a business point of view, the associated risks are also pointed in the table. It should be noted that, as a curricular internship and as agreed on by the hosting company, the project should be encapsulated from external factors.

3.7 - Project and Development Tools

To support both internship related processes and development, a collection of tools was chosen. That collection includes all the applications, websites, and processes used in the internship, from the support to the report writing to the used application to provide the communication with the deployment machine.

- **Overleaf** - This Online Latex Editor, allowed the writing of the internship report.
- **Docker and Docker-Compose** - This virtualization tool gives the ability to run and host different projects in a controlled environment. More info on this usage is available in 5.7.
- **Visual Studio Code** - This code editor allowed all the necessary actions to develop all the system.
- **Instagantt/Asana** - This project management tool, was used to plan the project and the different tasks over time.
- **GitLab** - This only platform, provided support for version control and code review.
- **GitLab Issues** - This feature of GitLab, was the main communication in technical questions, as well as reports, feature requests and documentation.

Chapter 4

Requirements Specification

In this chapter are specified the platform functional requirements, as well as its quality attributes. In order to best track the different possibilities of usage of the platform, the requirements are presented both in the form of user stories, to explain the functional usage as a whole, and the form of personas, to better exemplify the different possibilities of usage, according to different types of user.

4.1 - Terminology

To allow for a better understanding of the requirements through the User Stories Specification, the concepts related to the Ubiwhere's solution are indexed and are defined for each user by default. This indexes can be altered by the user at any time, saving their personal preferences and values.

4.1.1 - User Preferences Indexes

This indexes, express the user's preferences, factors that will be used to provide the most adequate route to the user. The fact that those range from 0 to 100 allow great customisation for each user, and provide data to better compare the different options, suiting it best to the user. In the upcoming Personas section, one can verify how different values attributed to the indexes can result in a different route suggestion.

General Indexes:

- Distance
- Value
- Speed

Those allow the user to specify how great is its concern about the different topics (in a range from 0 to 100), maximising or minimising the suggested trips accordingly.

Transport Mode Preferences:

- Foot
- Public Transport
- Bike
- Car

Those allow the user to specify how willing the user is to use a said transportation mode (in a range from 0 to 100).

Resources:

- Bike
- Private Car

Those allow the user to specify if it has and is willing to use a private bike or a private car.

Limit Constraints:

- Max Walk
- Max Bike

Those allow the user to specify how much distance he is willing to walk or pedal (in metres). Those constraints, although taken into consideration, can also be ignored if no accordingly option is generated.

Additionally, the user can also specify where its car is parked, so it can be considered in the route suggestion algorithm.

The user can also specify whether to use its last 10 chosen routes history in the route processing algorithm. In the history processing, the only chosen trips properties used are the percentage of the different transport modes.

All the above transports are considered at each trip request, if available in the area of operation, therefore the indexes are used to provide the most suitable suggestions to the user.

4.2 - User Personas

To better understand how users can interact with the platform, the best way is to define the different user characteristics and desires. The definition of personas is a greatly used technique, allowing better requirement specification and, in certain phases of the project, to test if the developed system suppresses satisfies its user base.

Persona 1

Name:	**Maria**
Characteristics:	• 18 years. • Has a bicycle and loves using it to go around town. • His workplace is within a few kilometres from his home.
Goals:	Wants to assess if his new school is within a distance she considers viable to make using a bicycle, as well as comparing it to public transportation, both in terms of travelling time, speed and price.

Persona 2

Name:	**Pedro**
Characteristics:	• 24 years. • Does not have a bicycle. • Has a car and licence but, on his daily commute, prefers using ambient-friendly ways of transportation. • His workplace is within a few kilometres from his home.
Goals:	Wants to assess if there is a way to make his daily commute using bikesharing, otherwise, he will take public transportation to do so, and would like to know which routes to take and the hours to do so.

Persona 3

Name:	**José**
Characteristics:	• 70 years. • Has a car and licence but does not like to drive inside cities. • Likes to use public transportation. • His little remaining mobility prevents him from walking and riding a bike.
Goals:	Would like to know not only the best way to reach the old part of the city using public transportation. If possible, he would also like to know if there a car park as close as possible, within the city range, so it can have a less stressful visit to the city.

Persona 4

Name:	**Ricardo**
Characteristics:	• 29 years • He's a traveller; enjoys visiting cities in Europe for the culture, and spend some days at each one. • Usually, he uses to travel by plane, and use the multiple available transport modes (free and paid) in a city. • Has no real preference for his transport modes.
Goals:	As he is travelling to a city soon, would like to know the best way to reach the hotel and different cultural sites around the city, as well as how to reach the airport.

Persona 5

Name:	Helena
Characteristics:	• 32 years • Has a car licence. • Due to her new position in her company, she must travel across the country, having multiple meetings in various cities. • Her company provides her with a car.
Goals:	As she is travelling across the country, and many cities have no indication about car parks, she would like to know both the way and the closest car park to her destination. Because she drives the car multiple times per day, she would like to keep track of her car in a map and use a system that considers the updated position whenever she wants to get directions for the next destination.

Persona 6

Name:	Rogério
Characteristics:	• 32 years • Has a car license. • His coworkers and him, working in multiple offices in the region, make several trips to each other's offices.
Goals:	As Rogério is about to make a trip between cities, he would like to know if any other coworker is in need of a carpool to a city in its path or the other way around.

Persona 7

Name:	**Rita**
Characteristics:	• 17 years • Makes a daily commute, using at least two buses and some walking time in between.
Goals:	Wants to check the time and details of its commute trip, using public transportation, at a different hour and day of the week that she is used to, allowing her to better prepare her day in advance.

Since the User Stories technique was also used in the internship, as a more strict requirement specification technique, the document can be consulted in Page 105.

4.3 - Quality Attributes

With the objective of integrating the developed backend product into the wide range of Ubiwhere services, it shall meet multiple standards to prove itself as stable, reliable and testable. Therefore, its quality should be kept in mind, with the following quality attributes accessed, taken into account during the development process, and measured.

- **Maintainability** - The ease with which the system can be fixed, expanded and modified to add features, increase its security, efficiency and speed.
- **Reliability** - Is can be expressed as the capacity of the system to remain in full functions over an interval of time.
- **Availability** - The proportion of time the system is available with all features available for use. It can be measured using up-time, as a per cent value, and there is a strong connection between the system availability and its reliability.
- **Interoperability** - The ability of the system to communicate with external systems, both for input and output, and when well managed, can help reduce information processing times and effectiveness.
- **Performance** - Can be expressed as the time the system takes to execute a given action and may vary with the action. It can be measured as latency (amount of time to respond to an action) and throughput (the number of actions in a given interval of time).
- **Scalability** - The capacity of the system to be submitted to varying workloads (and especially large amounts of it).
- **Security** - The ability of the system to deny access, modification or loss of private information, deny operations from unauthorised users, and prevent generally malicious actions.
- **Testability** - The ease to create the tests for the system, including defining the criteria, the interface of testing, and the general ability to identify the problems that arise in testing.

The system behaviour according to each of the scenarios can be consulted in the annexes, in Page 4, in the form of scenarios, allowing the reader to gather a better understanding on the different qualities of the developed product, regarding both architectural conception and development techniques.

Chapter 5

Architecture and Technology

In this chapter are explained the project constraints, the architectural decisions that were taken, the architecture of the product developed in the internship and the technological decisions made along the whole planning and development process. The system architecture is expressed in detail, as well as the technologies used to implement it.

As the developed system aims to integrate multiple components, both internal and external, this decision process proves itself important to better suit both the requirements and the quality attributes.

5.1 - Project Background

As said in previous chapters, Ubiwhere's aim with the final product is to deliver a user-friendly final application using all its services available and using the information provided by the cities. Those services and information can be divided into two categories:

- **Internally stored information** - This data is stored on the servers at Ubiwhere. It can be accessed faster, but its access must be controlled. This data should be updated regularly, gathering the information from the external components and entities (like public transport companies).
- **Information provided by external components on demand** - External components, outside Ubiwhere's control, whose data fetched follows FIWARE Data Models. This allows real-time information to be used in the system processing and responses.

Therefore, the architecture conceived must comport the possibilities for both types of integrations.

5.2 - Open Source Projects to Integrate

To support most of the complex tasks in the project, are useful already tested, deployed and open source projects, instead of building a proprietary one, reducing the necessary time to develop the project. With that in mind, the following components are or have already been used by Ubiwhere and, analysing their characteristics, are suitable for the necessities.

5.2.1 - Open Trip Planner

To be able to provide the user with a routing system considering public transportation routes, the decision was to use Open Trip Planner (OTP), an open source platform capable of using the GTFS data to provide journeys based on transit information. This project can be run in an internal machine, and thanks to its client-server model and REST API implementation, is suitable to be used as the base routing system.

5.2.2 - Open Source Routing Machine

While OTP is used to generate the public transportation routing, is the chosen counterpart to support the routing management of the private transportation. Its flexible usage of Open Street Maps data, using OSM files, allow usage of data from the cities and countries that provide them free to download, as well as the data provided to Ubiwhere. While being open source, it was chosen for the role in this project due to the wide experience of Ubiwhere's employees on this tool.

5.2.3 - Celery

To better distribute the backend processing, as well as to schedule the needed tasks to keep information updated in the system, there must be used a distributed task and queue job tool. Since Django was chosen as the technology to use, the open source project Celery is used to suppress those needs, due to its simplicity but also high possible customisation.

5.2.4 - Orion Context Broker

To manage and use the data provided by the transport companies, the Orion Context Broker is used when necessary, as a context to store and serve the information. Since Ubiwhere has great experience on its usage, is expected that, in conjunction with data harmonisation following Fiware Data Models paradigm, the information usage is stable and easy using this project instances.

5.2.5 - Fiware Data Models

It's also important to make sure both information follows the standards on quality and standardisation. With that in mind, the usage of Fiware Data Models can prove helpful, as this allows for information standards to be applied to information from multiple sources. Ubiwhere is a contributor to the development of Fiware Data Models concept, which aims to standardise the models of data provided by different sources.

5.3 - Data and Information from Cities

To develop the project, Ubiwhere promoted relations with specific cities, in order to get information and data on the multiple transport systems in place. The cities on which the information collection had greater success were Cologne and Valencia, both with great emphasis on city planning and efficiency, predisposition for innovation, and resident satisfaction policies. The data will be accessed through cities open data portals and harmonised on the middleware to be developed, as described in chapter 5.

Relative to the city of Cologne, the following information/data is available:

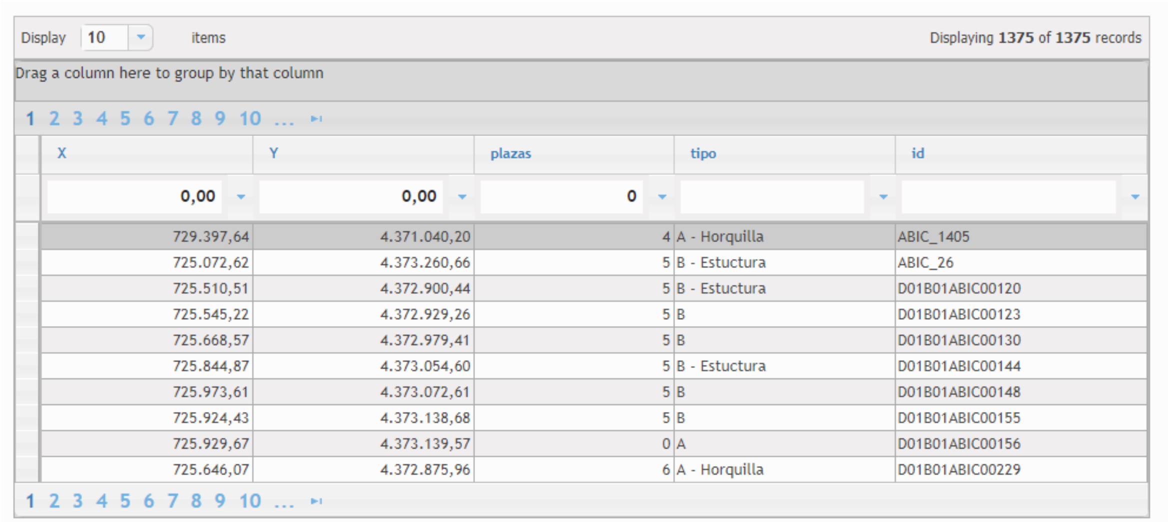
- Routing Software - Available for deployment with OTP and OSRM.
- Bicycles and Bicycle Parks - Available in a city hosted website, as a set of information, as seen in Figure 5.1.


```
▼<markers>
  ▼<country lat="50.9364" lng="6.96053" zoom="11" name="KVB Rad Germany" hotline="+493069205046" domain="kg" country="DE"
   country_name="Germany" terms="https://www.kvb-rad.de/de/agb/" policy="" website="https://www.kvb-rad.de/" booked_bikes="0"
   set_point_bikes="1447" available_bikes="815" capped_available_bikes="1">
    ▼<city uid="14" lat="50.9429" lng="6.95649" zoom="12" maps_icon="norisbike" alias="koeln" break="0" name="Köln" num_places="854"
     refresh_rate="462" bounds="{"south_west":{"lat":50.8636,"lng":6.8158},"north_east":{"lat":51.0237,"lng":7.0284}}"
     booked_bikes="0" set_point_bikes="1447" available_bikes="815">
       <place uid="22756" lat="50.957410536847" lng="7.0119494627215" name="Bahnhof Mühlheim " spot="1" number="4812" bikes="0"
       free_racks="0" terminal_type="free" place_type="0"/>
     </city>
   </country>
   <country lat="46.0275" lng="25.0269" zoom="6" name="nextbike World" hotline="+443300109343" domain="nw" country="RO"
   country_name="Romania" terms="https://nextbike.com/index.php?aam_media=1321" policy="" website="http://www.nextbike.com/"
   booked_bikes="0" set_point_bikes="8" available_bikes="0"/>
 </markers>
```


Figure 5.1: Bicycle Parks in Cologne

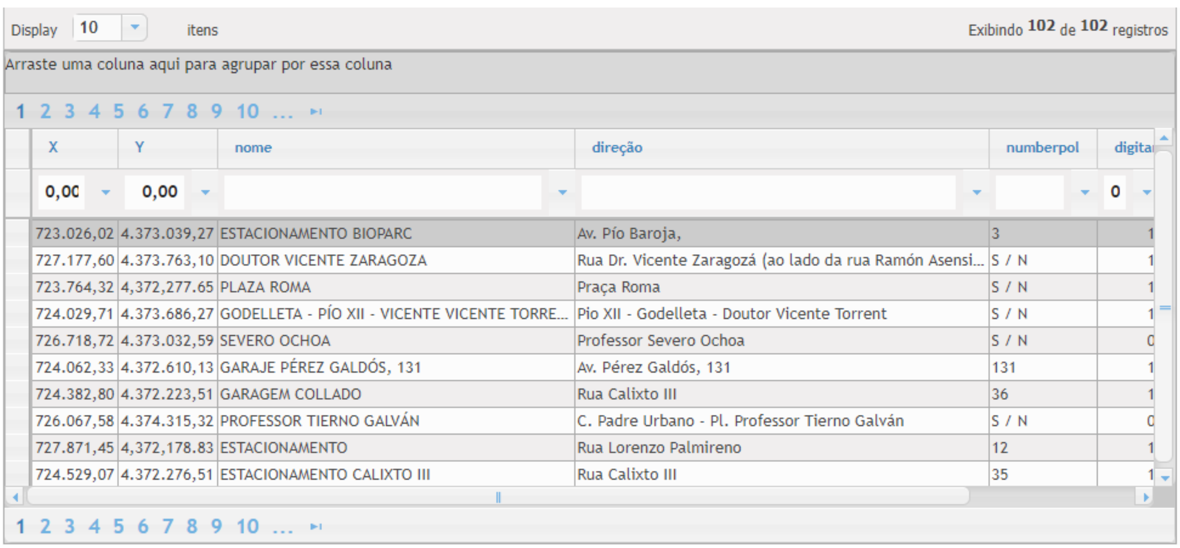
Relative to the city of Valencia, the following information/data is available:

- Routing Software - Available for deployment with OTP and OSRM.
- Bicycles and Bicycle Parks - Available in a city hosted API, as seen in Figure 5.2.
- Park and Go locations - Available in a city hosted API, as seen in Figure 5.3.



X	Y	plazas	tipo	id
0,00	0,00	0		
729.397,64	4.371.040,20	4	A - Horquilla	ABIC_1405
725.072,62	4.373.260,66	5	B - Estuctura	ABIC_26
725.510,51	4.372.900,44	5	B - Estuctura	D01B01ABIC00120
725.545,22	4.372.929,26	5	B	D01B01ABIC00123
725.668,57	4.372.979,41	5	B	D01B01ABIC00130
725.844,87	4.373.054,60	5	B - Estuctura	D01B01ABIC00144
725.973,61	4.373.072,61	5	B	D01B01ABIC00148
725.924,43	4.373.138,68	5	B	D01B01ABIC00155
725.929,67	4.373.139,57	0	A	D01B01ABIC00156
725.646,07	4.372.875,96	6	A - Horquilla	D01B01ABIC00229

Figure 5.2: Bicycle Parks in Valencia



X	Y	nome	direção	numberpol	digita
0,00	0,00				0
723.026,02	4.373.039,27	ESTACIONAMENTO BIOPARC	Av. Pío Baroja,	3	1
727.177,60	4.373.763,10	DOUTOR VICENTE ZARAGOZA	Rua Dr. Vicente Zaragozá (ao lado da rua Ramón Asensi...	S / N	1
723.764,32	4,372,277.65	PLAZA ROMA	Praça Roma	S / N	1
724.029,71	4.373.686,27	GODELLETA - PÍO XII - VICENTE VICENTE TORRE...	Pio XII - Godelleta - Doutor Vicente Torrent	S / N	1
726.718,72	4.373.032,59	SEVERO OCHOA	Professor Severo Ochoa	S / N	0
724.062,33	4.372.610,13	GARAJE PÉREZ GALDÓS, 131	Av. Pérez Galdós, 131	131	1
724.382,80	4.372.223,51	GARAGEM COLLADO	Rua Calixto III	36	1
726.067,58	4.374.315,32	PROFESSOR TIERNO GALVÁN	C. Padre Urbano - Pl. Professor Tierno Galván	S / N	0
727.871,45	4,372,178.83	ESTACIONAMENTO	Rua Lorenzo Palmireno	12	1
724.529,07	4.372.276,51	ESTACIONAMENTO CALIXTO III	Rua Calixto III	35	1

Figure 5.3: Valencia Car Parks information

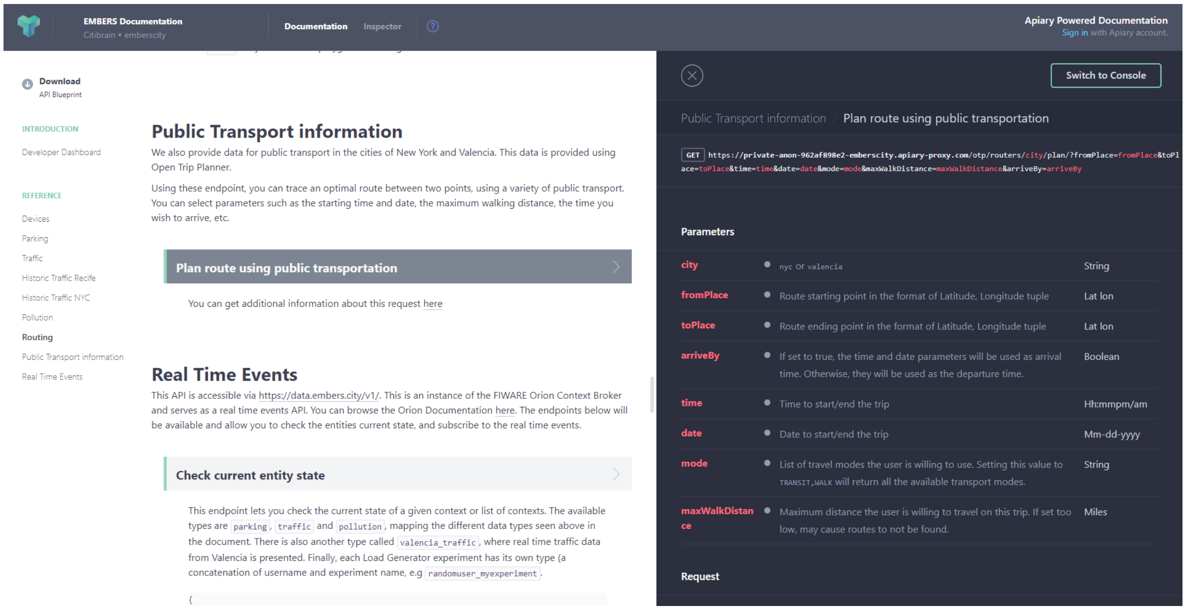


Figure 5.4: Routing Engine

The bikesharing and parking information sets, available from different sources, are to be used in the system when processing a route request. In order to keep the response time of the route suggestion system as little as possible, this information is not gathered on demand but instead saved and read from a reserved database. This database is updated at configurable intervals of time, using Celery schedule tasks.

5.4 - Architecture Overview and Decisions

In the architecture specification, it is important to decide if the system architecture should be monolithic or a microservices oriented one. Below, these are described for better understanding:

- **Monolithic Oriented** - The system is a single component, with the option for multiple parts within the unit. The testing phase (an important one, regarding Ubiwhere's standards) is slower when compared to microservices oriented, as the system must be tested and developed as a whole.
- **Microservices Oriented** - The system is split into different components, independent from each other, but communicating between themselves when needed. This approach allows for independent testing and reduces the coupling in the system, also improving scalability and reusability.

To better make the decisions regarding the system architecture, one should also understand the cohesion and coupling definitions.

- **Cohesion** - A indication of how much fit are the responsibilities of a component. A good component responsibility designation (what it should do) increases cohesion.
- **Coupling**- A indication of the dependence between components. A higher coupling between two components means that they need each other to function.

With both concepts in mind, and in order to develop the most suitable system, considering as main factors its expandability, testability and scalability, the following architecture decisions:

- The system architecture must be microservices oriented, for the following reasons:
  - The system must be expandable, and Ubiwhere's final system is going to use the product developed in the internship as a base. With that in mind, splitting it into multiple sub-systems will allow for better maintenance, easier and faster testing and easier improvement.
  - Integrating different components allow for technological differences between them, like different programming languages, frameworks and databases.
  - As different components have different responsibilities, they also need to access and/or edit different parts of information in databases. As an example, a service dealing with bike-sharing information doesn't need to access user travel history.
  - A modular architecture can also mean easier replicability and scalability.
- With increased awareness on the components cohesion, its possible for them to be used in different projects, being them internal or external to Ubiwhere (if it is decided to open its usage), allowing for both code and component reuse.

5.5 - Technology Overview and Decisions

To choose the technologies and frameworks of the developed system, the following factors were taken into account, in order of importance:

- Suitability for the subsystem to implement;
- Ubiwhere's knowledge of the technology/framework;
- General community knowledge available on the technology/framework;
- Ramp-up time when starting using the technology/framework. The ramp-up time is defined as the initial development phase, in which the developers' not-ideal knowledge level on the frameworks and tools lowers the development performance;
- Time to implement using the technology/framework;

With all those factors in mind, eliminating some options altogether, the study conducted in the multiple available technologies/frameworks resulted in the following options:

- **Slim** - A PHP microframework simple enough to rapidly deploy an API.
- **Ruby on Rails** - Ruby on Rails is an open source web MVC framework written in Ruby.
- **Django** - A Python Framework with a lot of community knowledge available, embracing the MVC pattern.
- **Flask** - Flask is a simple and lightweight Python microframework.
- **Spring** - A Open-Source Java Framework gaining popularity as a full-stack development framework.
- **Play Framework** - A lightweight Java framework.
- **Express.js** - Express is a minimal and flexible node.js framework.

The above options have been considered, having fulfilled the first requirement, that is suitable for the implementation objective - constructing a *JSON REST API*.

The pros of the usage of Django, as well as its intrinsic benefactor characteristics, are as follows:

- With Django REST Framework, the development of the REST API can be shortened without loss of quality and supports all necessary securities and good procedures.
- Using Python for the processing part of the project, while not being the faster performance option, represents the best compromise between performance, development and debugging ease and testability.
- The Django ORM (Object-Relational Mapping) provides not only an easy way to interact with the database but also a powerful solution to geographic queries, sparing a lot of time that simple actions would take, otherwise; a good example is the ”Retrieve the closest point to a given point” query. Django is also held as a technology with remarkable support for systems with geographic components.

- Its typical app stratification and organisation allow for a modularly designed system. With this in mind, and being modularity one of the main quality attributes, the organisation is an important factor.
- The amount of open source and licensed packages available is one of the greatest in this area, and its use allows to better and more systematic development.
- The documentation of the framework, as well as the used secondary technologies, is widely available in both official docs and community forums/tutorials.

The greatest factor to keep in mind is the fact that Ubiwhere can provide great support for the development process, thanks to a huge company level knowledge on it. It should be duly noted that, as the internship product is the core of the final product, it's important to ease the transition process and provide a quality developed product to serve as a base for the oncoming iterations.

The hosting company also provides information, tools and a knowledge base to diminish the ramp-up time that, when allied with the intern's knowledge on Python and the Django Framework, can provide a suitable environment for better programming practices, product quality and higher development speed.

The database technology used across the system is Postgres, and it's used to store all the information in the system, including account details (emails, usernames and passwords) and related user preferences, as well as the routes information. Postgres is a relational database, with great community knowledge that, when allied to its reliability and simplicity, provides a suitable solution for the product of the internship. As the amount of data to be stored per user is not considerably large, the scalability is assured using the Postgres replication mechanisms.

This technology also proves itself the correct choice when many other factors are considered: it's considered by the development community to be one of the most compatible database's system in geography data support and, when combined with Django default ORM, provides a successful match between operation speed, compatibility with used technologies, replicability (in the future) and compatibility with the other development techniques and tools (like Docker).

5.6 - System Architecture

With all quality attributes kept in mind, and on par with the technology decisions process, the system architecture was defined. This iterative process, based on meetings and brainstorm sessions between the intern, the advisor and other employees at Ubiwhere, resulted in an architecture suitable for the quality attributes requested both for the internship and the future product.

It should be noted that, during the internship, there was never the objective of a fully scalable deployed project; this involves a lot of time, effort and knowledge; the main focus of the internship was based to prove functionality of the system. With that in mind, the architecture and its implementation were approved and considered better than expected, within the bounds of the required for the internship.

To be considered scalable, this architecture is absolutely stateless, in every developed component. This factor is important to allow the replication of the different components (both for scalability reasons and also for modularity ones) without additional development time.

5.6.1 - Architecture Diagram

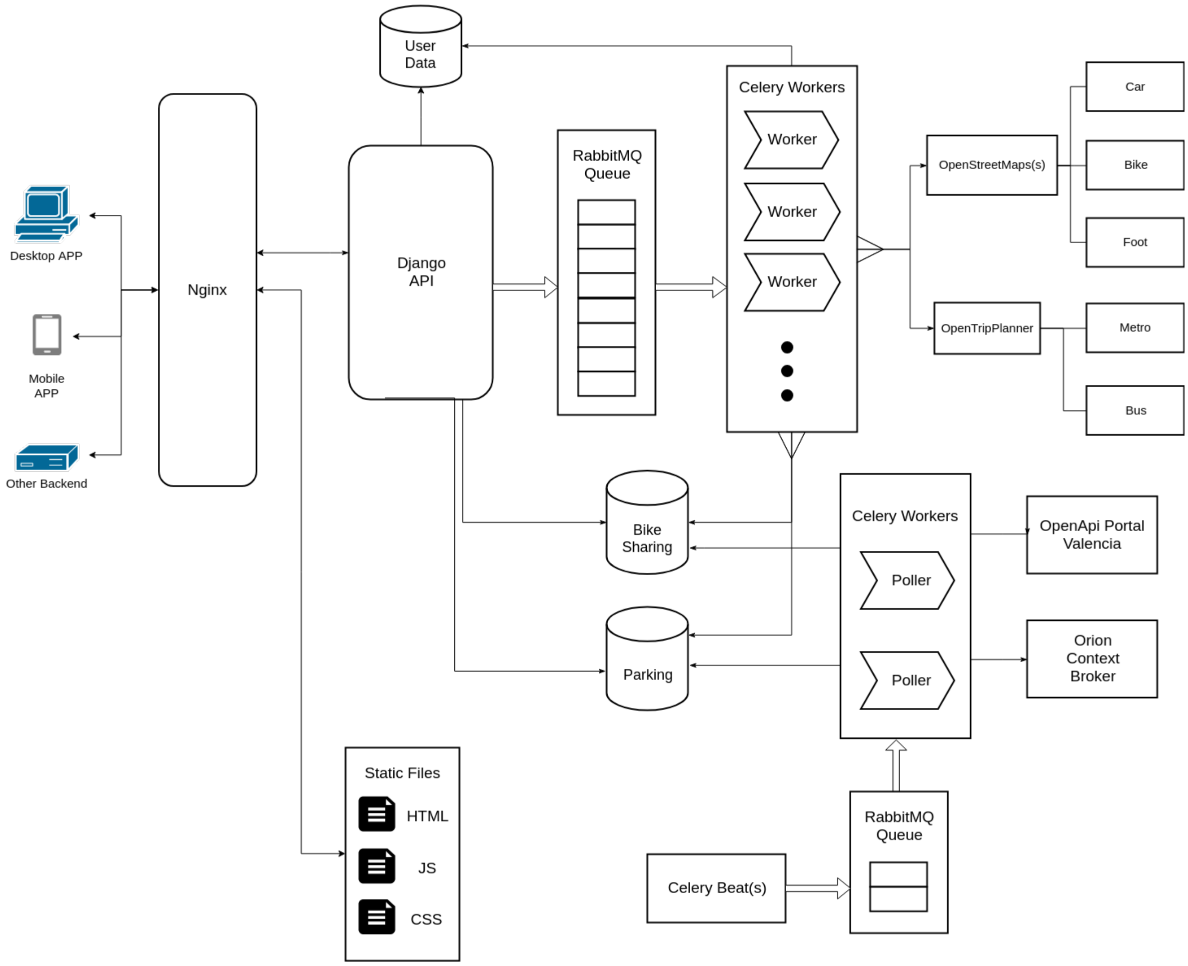


Figure 5.5: System Architecture

Presented in 5.5 is the architecture of the system, with all its components. From left to the right, the system becomes less accessible to the exterior (while the only component that is reachable from outside being nginx) and more cohese. There are represented 3 schemas, but it should be noted that there is only one Postgres instance (to diminish the number of components needed for the deployment).

5.6.2 - API and nginx

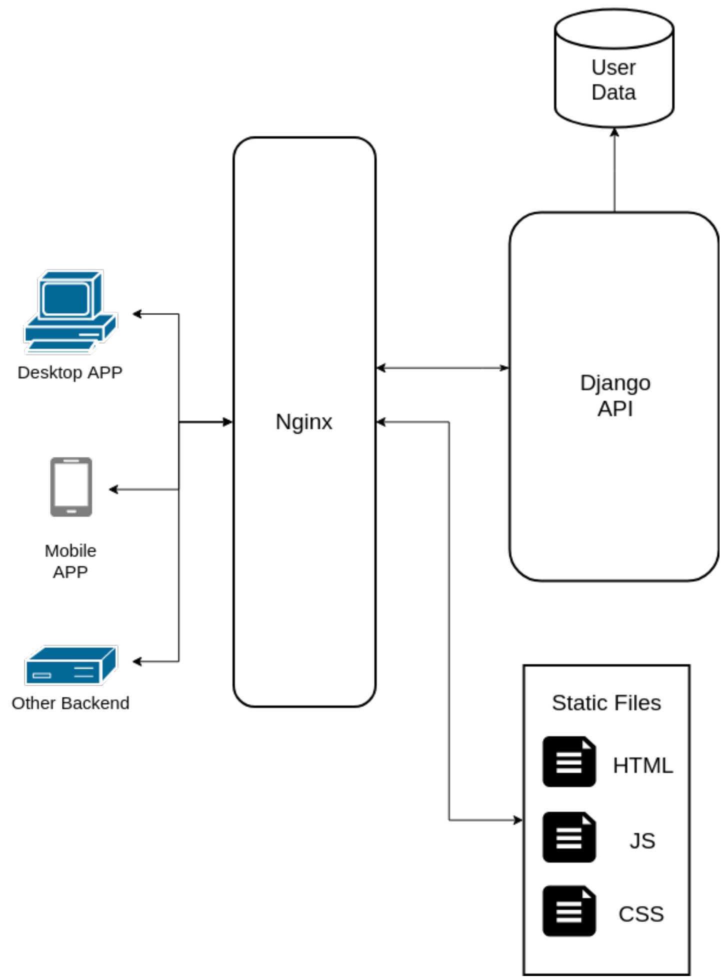


Figure 5.6: API and nginx Subsystem

The usage of nginx allows for usage of certificates for the connections (HTTPS) and, in the future, allows for the implementation of load balancing solutions. Therefore, the nginx instance serves as a gateway between the exterior and the deployed Django server (as a WSGI application).

This component also serves directly the static files, like HTML, JS and CSS; this has been done in the testing frontend.

5.6.3 - Queuing System

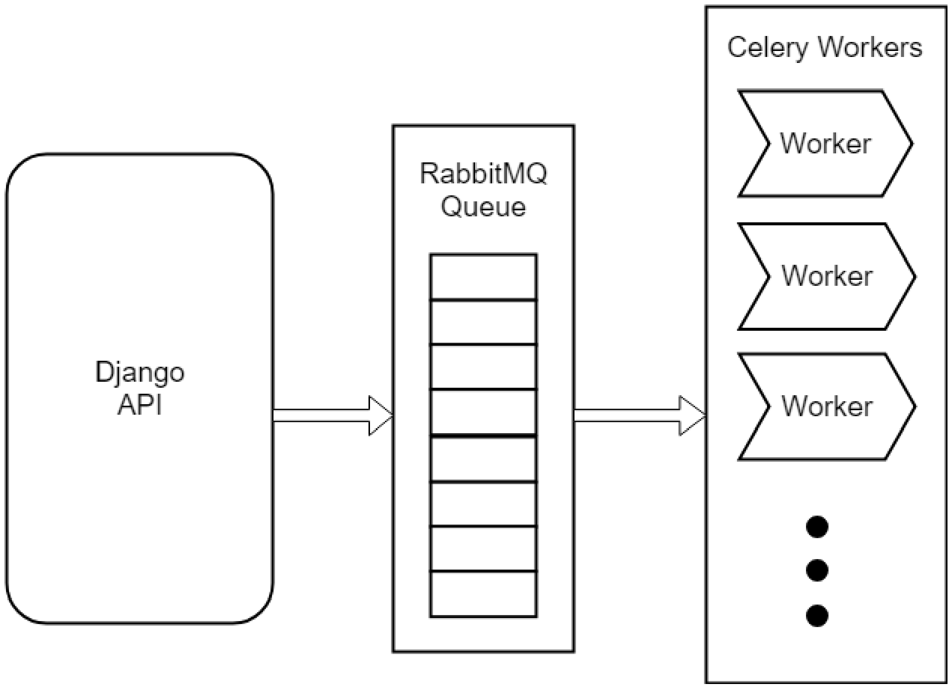


Figure 5.7: Queuing Subsystem

The route suggestion system, to allow for faster response times with both simple and more complex requests, is tailored in an asynchronous work disposition; instead of calculating all different options in a single workflow, the different work tasks are put in a ”pending work” queue. Celery workers, instances that run whichever task is in their appointed queue, execute the task and save the results in the database. When all tasks are completed, the API can then execute the necessary operations and respond to the request.

This process flow, while somewhat hard and slow to implement by the intern, who didn’t had any knowledge on the practice, revealed not only a instructive option but also an effective one: during development, response times fell to under 25% when this flow was implemented, in contrast to a single workflow one.

This flow is better explained, using workflow diagrams, in the implementation section.

5.6.4 - Route Request System

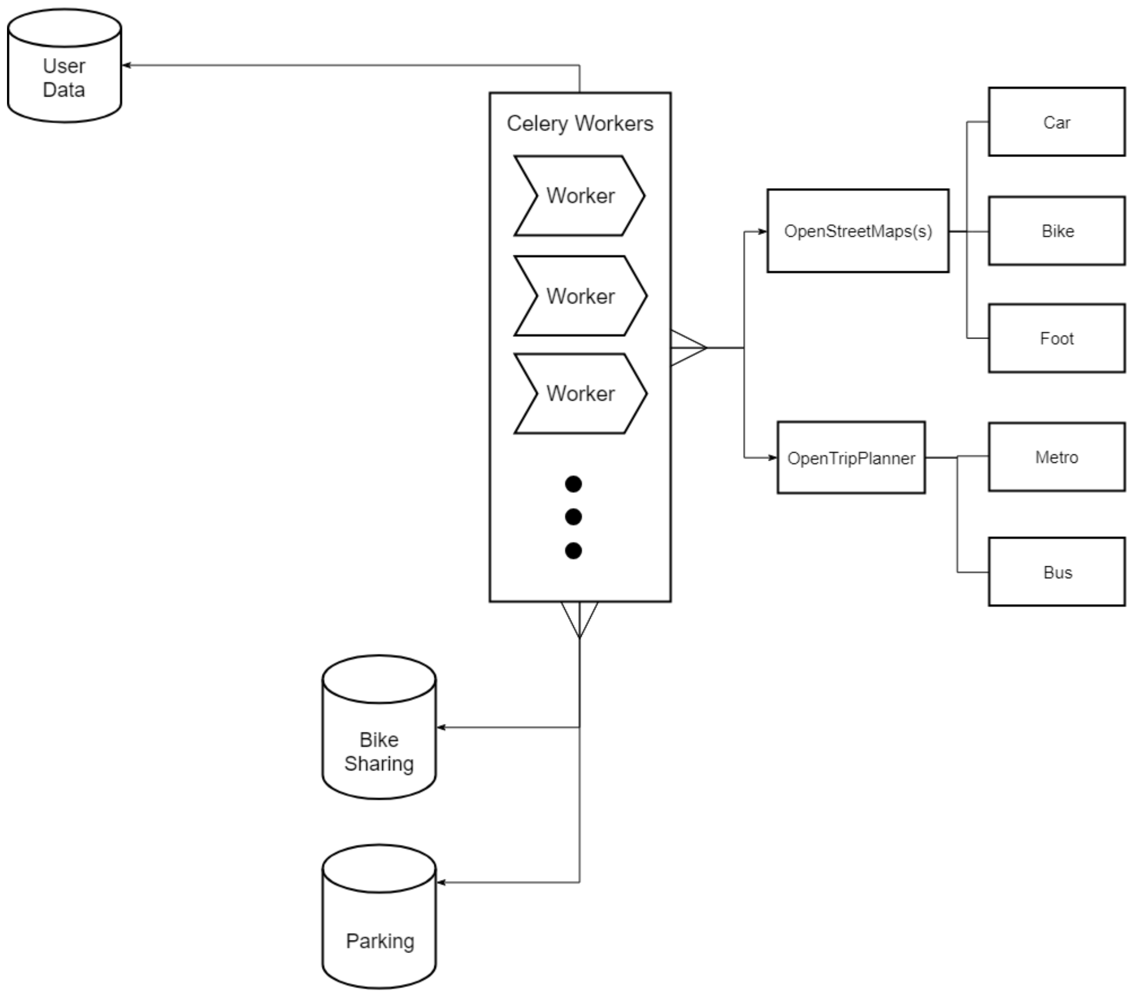


Figure 5.8: Route Request Subsystem

The components OSRM and OTP, visible in the architecture, are both open source and standalone. It should be kept in mind that OTP uses multiple graphs (like metro and bus) and, in this architecture, runs at the rate of one instance for each city. In contrast, as each instance of OSRM can only deal with one type of transport (foot, car or bike), the necessary number of instances per city is 3 (to deal with the necessary requests).

The usage of the multiple databases by Celery tasks allow for lower coupling: when a task needs bikesharing or parking information, it accesses the database directly (reading entries but never writing or updating them).

Each one of the Celery tasks is responsible for one or more requests to the routing components, which performed readily enough to never be considered the possibility of needing to scale both OTP and OSRM.

5.6.5 - Information Updaters

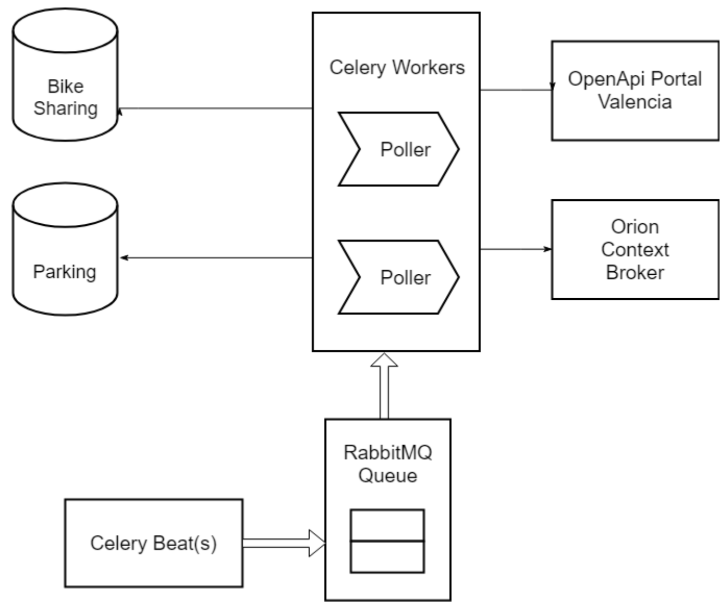


Figure 5.9: Information Updaters Subsystem

To keep information updated, from external entities to the project, there was a need for an asynchronous workflow, too. Right from the start of the project, there was obvious that the information from bikesharing and parking could not be accessed on demand, with the risk of compromising the performance of the system, and increasing dependency from the external entities.

With that objective in mind, Celery tasks where used, too, in conjunction with a work queue and Celery Beat. The last component, Celery Beat, allows for task scheduling, by the time elapsed and hour schedule. In the present architecture, the Celery Beat component inserts tasks in the queue, while the Celery workers(s) remove the task and execute the procedure, updating the information in the databases.

The bikesharing and parking information in the database also follow models derived from Fiware DataModels, which means that this part of the architecture can be used, with little effort, in another project. This structure allows for the necessary modularity, too: if the information source or models change, only the task procedure needs to be updated.

5.7 - Docker Virtualization

In order to run the different components both in a local development system, as well as in deployment, Docker was used. This platform allows virtualization of the all the architecture in a controlled environment, with control over networks, volume mapping and state.

The direct mapping of the different components to a corresponding volume, and the ability to make the developed components, compiled, tested and ready to use inside the company proved useful both to the intern and the employees using modules of the internship project in other applications.

The usage of Docker represented a big change in the development practices for the intern; the usage of virtualization was not usual in his background and the ramp-up time was considerable.

Nevertheless, it's safe to say that the usage of Docker, as well as Docker-Compose (basically, a managing tool for Docker), allowed for better coding practices: the development was completely independent of the operating system used in development, mitigating many of the expected compatibility problems when using databases, frameworks and other variables.

This usage is also transported to the deployment phase of the project, and that's the main reason why it is used in most projects developed by Ubiwhere. It is also held as one of the greatest cases of the ability of the intern to adapt to both the technologies and mindset of the hosting company.

An example of the usage of Docker, is seen in Figure 5.10. This usage spares the usage of components installed in the native OS. Since the architecture uses many different components, this usage also minimises clutter and allows better organisation of running components.

```
            Name                              Command                State
-------------------------------------------------------------------------------
backend_api_1                  python manage.py runserver ...   Up
backend_osrmcolognebike_1      osrm-routed -p 5002 koeln- ...   Up
backend_osrmcolognecar_1       osrm-routed -p 5001 koeln- ...   Up
backend_osrmcolognewalk_1      osrm-routed -p 5003 koeln- ...   Up
backend_osrmportugalcar_1      osrm-routed -p 5007 portug ...   Up
backend_osrmvalenciabike_1     osrm-routed -p 5005 valenc ...   Up
backend_osrmvalenciacar_1      osrm-routed -p 5004 valenc ...   Up
backend_osrmvalenciawalk_1     osrm-routed -p 5006 valenc ...   Up
backend_otp_1                  java -Duser.timezone=Europ ...   Up
backend_pollerbeatcologne_1    celery --app=poller beat - ...   Up
backend_pollerbeatvalencia_1   celery --app=poller beat - ...   Up
backend_pollercologne_1        celery --app=poller worker ...   Up
backend_pollervalencia_1       celery --app=poller worker ...   Up
backend_postgis_1              docker-entrypoint.sh postgres    Up
backend_rabbit_1               docker-entrypoint.sh rabbi ...   Up
backend_worker_1               celery --app=maas worker - ...   Up
```

Figure 5.10: Docker Command Line Interface Usage

Chapter 6

Implementation

Regarding the implementation of the project, there were multiple questions at the beginning of the project and raised over time. This chapter sections regard the multiple relevant characteristics of the system, as well as the processes developed, that make the project fulfil both functional and non-functional requirements.

6.1 - Component Communication and Isolation

The communication with the project is always to the component nginx, using HTTPS, as seen in the Figure 5.5. This component makes the forwarding necessary to the necessary components, in a typical architecture, and in this case, forwards the requests to the Django application. All the media files and static files, not delivered by the Django component (like JS and CSS files), are directly served by nginx. This allows for a simple but effective layer of isolation, restraining the access to the remaining components of the project.

The network of the project is also configured in the docker-compose file; in this case, only one network is necessary, providing a layer of communication between the components in the project. In practical terms, all the sub-systems are aware of the existence of the others and can communicate freely and in an isolated matter. The usage of this virtualization proved effective and simple while allowing the expansion of the system in the future.

6.2 - Requests for External/Internal Components

For the necessary communication with the external systems, as well with the systems running locally that use HTTP, it was used the *Requests* package. Firstly, though, the used package was *urllib*; the transition was suggested by supervisors at Ubiwhere, studied, and then taken on. The *Requests* package usage, while simple, allowed for better customisation of the request, as well as using authentication to use with the components that require it. The error reporting system implemented also allowed for error awareness and redundancy behaviour to be implemented, which not only proved effective but also remains the go-to tool for the job is performing, both among Ubiwhere employees and online community.

6.3 - Account Management and Authentication

The project, being built around the user customisation of his/her request, allows the creation of accounts, login actions and customization of preferences. In the following figures, one can notice the different types of requests to the system.

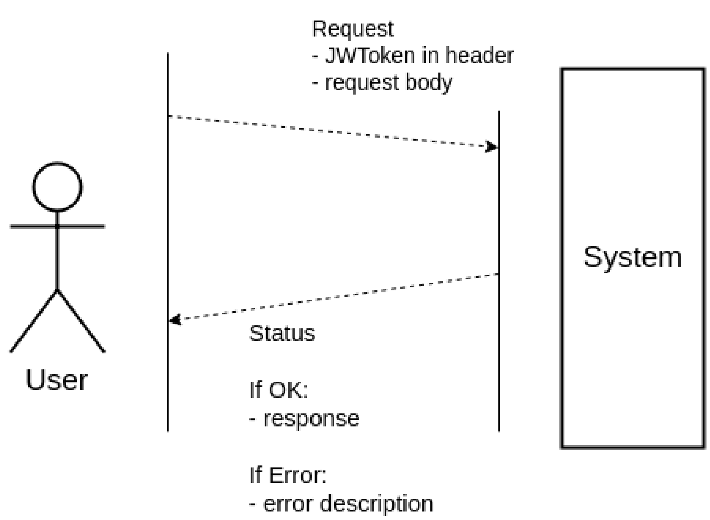


Figure 6.1: Account Registration

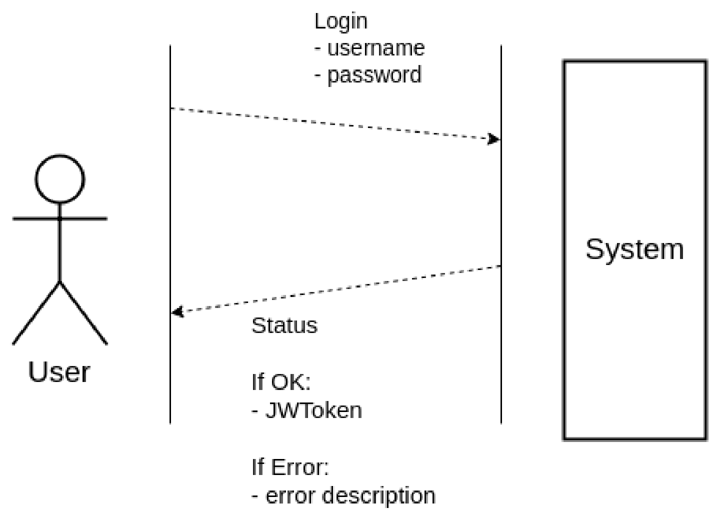


Figure 6.2: System Architecture

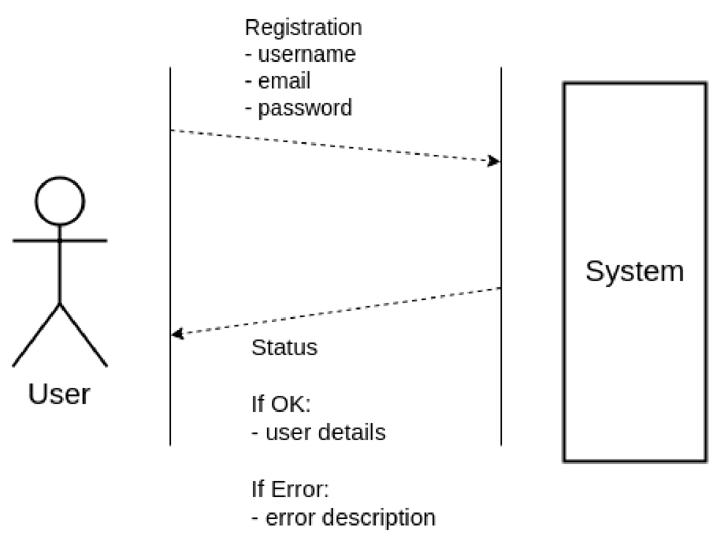


Figure 6.3: System Architecture

In Figure 6.1, one can notice the required fields to create an account in the system. It should be noted that, since the user mapping is made using both his/her email and username, those are unique for each user.

In Figure 6.9, is visible the authentication process; the user submits his username and password and, if correctly matching to a user in the system, his/her JSON Web Token (JWT) is included in the response (if that is not the case, the response is informative on the error that occurred). This JWT, with an expiration time of 7 days (being this the chosen time interval) should then be used in the header of all the remaining requests made to the system, as seen in Figure 6.3; if it does not match an active token, the system rejects the request. This system, while being effective in terms of security, is also integration friendly. In the future, a possible integration with social networks can also be implemented, and the usage of JWT allows for standardisation of all the login possibilities.

It should be kept in mind that the guidelines of the General Data Protection Regulation were not enforced, as the project is not in a deployable product state, some thought and effort was put into the preparation to that, nonetheless; as an example, all information of the system on the user (both preferences and any kind of routes requested) can be deleted, not compromising the system status and information integrity. It should be noted that, in the future product, there should be also issued a privacy statement; that is not necessary for the internship, due to the restricted and controlled access to the platform.

6.4 - Request Workflow

To process the multiple necessary works adequate to any request, a linear processing soon proved insufficient to match the system quality attributes appointed. With that in mind, instead of the linear workflow, an asynchronous Celery worker system was implemented:

- Each one of the workers has all the possible tasks implemented.
- When the Django component wants a task to be done, it should be put in the work queue.
- When a worker realises a task is pending in the queue, it removes it and processes the necessary work.
- When the work is performed, the Django component receives the callback and can now access the updated information in the database.

This process is illustrated in the Figure 6.4.

It should be noted that the information is consulted in the database, after the multiple asynchronous processes, to aid in the stability of the system (an asynchronous complex response could lead to interpretation and queue overload problems).

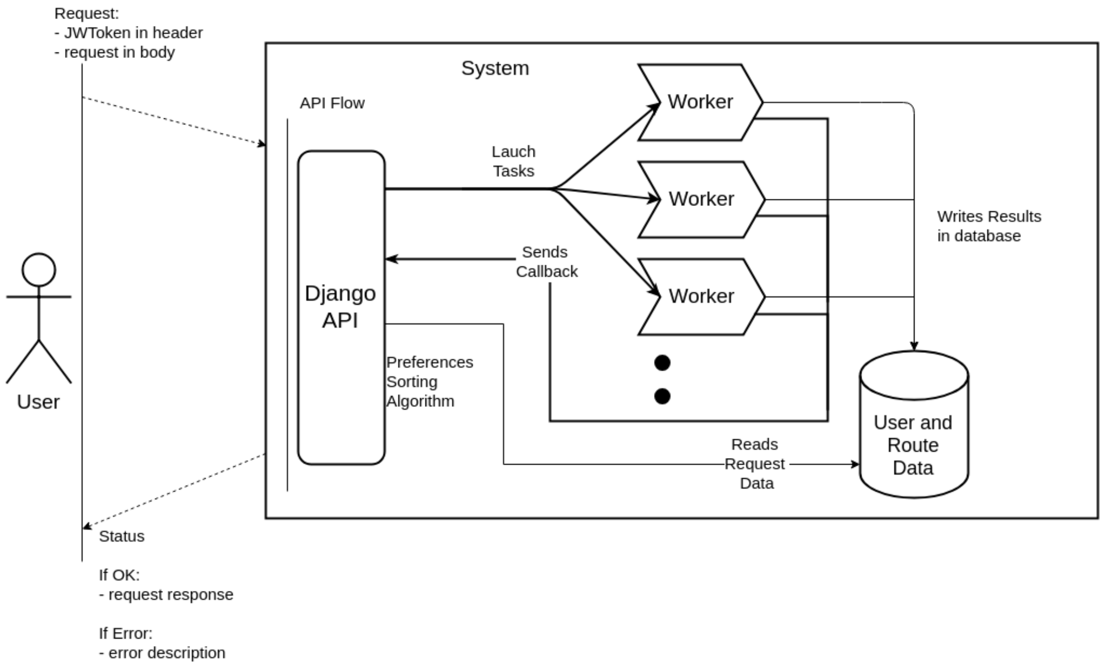


Figure 6.4: Request Workflow

With all this process implemented, we have fulfilled the necessary requirements to implement an asynchronous workflow, while the REST system input and output are synchronous and blocking. This implementation was a great part of the project, due to lack of knowledge of the intern on the problem solutions, technical limitations and ramp-up time of the used technologies, as well as the need for great code organisation.

6.5 - Information Structure on Database

To better acknowledge the information structure, the Figure 6.5 represents the Entity Relation Diagram (ER) of the system.

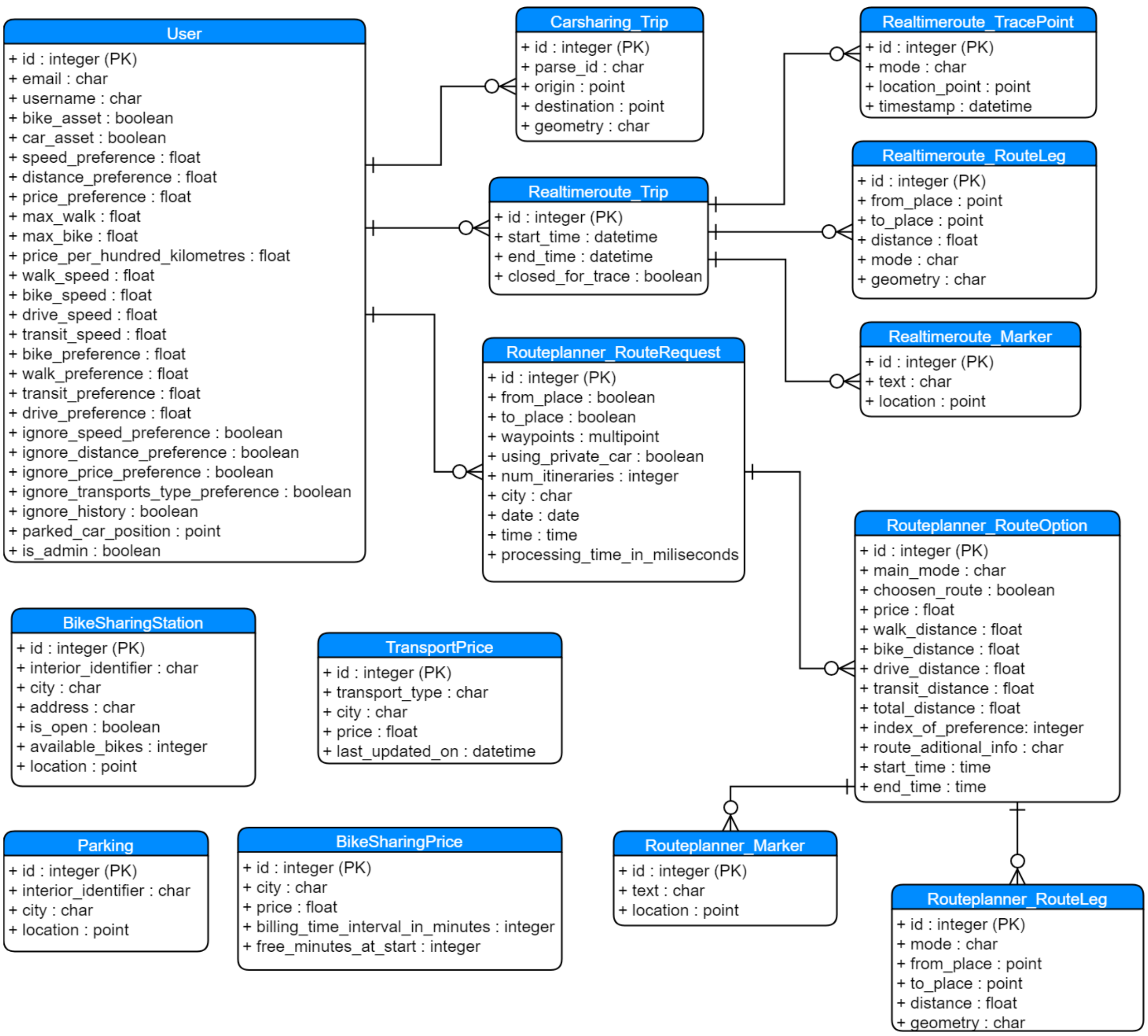


Figure 6.5: Database Models and Relationships

In the ER, are illustrated the different tables. The auxiliary ones, which are never updated by the backend serving the requests, are:

- **BikeSharingStation** - this table holds the information on the bikesharing stations in the different cities, and is updated on a regular basis by the poller components in the architecture.
- **Parking** - this table holds the information on the car parks in the different cities, and is updated on a regular basis by the poller components in the architecture.
- **TransportPrice** - this table holds the different prices of public transportation in the cities.
- **BikeSharingPrice** - this table holds the different prices of bikesharing tariffs in the cities, with the necessary information to calculate the price associated.

Considering the route requests and remaining tables, the division is as follows:

- **User** - this table holds all preferences and information on the users.
  - **CarSharing_Trip** - this table holds the information on the carsharing trips components.
  - **RealTimeRoute_Trip** - this table holds the information on the trip tracing component.
    - **RouteLeg** - this table holds the information on each of the step of the trip, with a geometry that can be displayed on a map.
    - **Marker** - this table holds the information on the markers of the trip, showing the multiple points of interest for the trip (like bus stops or transport mode changes).
  - **RoutePlanner_RouteRequest** - this table holds the information of the route planning requests submitted but the user.
    - **RouteOption** - each entry in this table represents one route option generated by the system, considering the route request.
      - **RouteLeg** - this table holds the information on each of the step of the trip, with an geometry that can be displayed on a map.
      - **Marker** - this table holds the information on the markers of the trip, showing the multiple points of interest of the trip (like bus stops or transport mode changes).

This information organisation was not only used internally, to keep maintain good practices but also is observable in the system response: the response of the request follows the structure exposed above, easing the usage of the project as a modular backend.

6.6 - System Modularity

One of the most required and looked at quality attributes was the modularity of the system; since the project was to be run with a great span of different components, file organisation and error reporting were subject of great thought and effort.

Regarding the backend implementation, the division of the components in the Django project is crucial; not only as an organisation practice but also as code reuse. This project decisions regarding that were as follows:

- If a determined set of features can be isolated from the rest of the system, it should be implemented in a different Django app.
- If different apps shared databases and/or resources, it all should be well defined in the settings, in order for an exterior person could understand the dependencies.

These rules proved effective, as two of the apps (Carsharingtrips and Realtimeroute) were already used standalone, on other projects by Ubiwhere, or integrated with other mobility solutions. The Django project apps are illustrated in Figure 6.6.

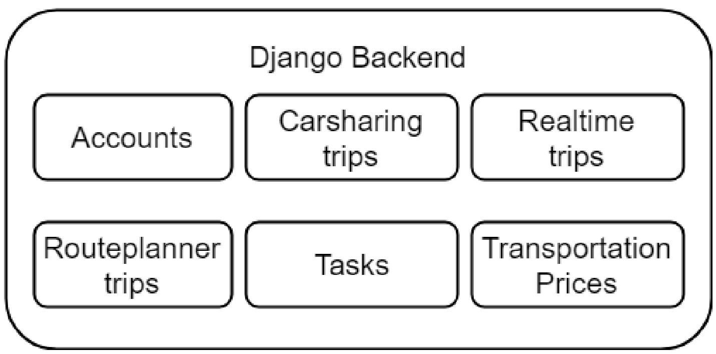


Figure 6.6: Django Apps implemented

Another customisation option, greatly appreciated by the hosting company and implemented by the intern, is the ability to define both the cities and the associated features to each using a JSON configuration file. Not only the system only uses the necessary tasks in conformity with the available features in the city of the request (after assessing them), but can also use the defined URLs to of internal and external components to make the OTP and OSRM requests.

```
{
    "cities": {
        "valencia": {
            "center": [
                -0.372223,
                39.472396
            ],
            "circle_range": 0.15,
            "routers": {
                "OTP": "http://otp:8080/otp/routers/valencia/",
                "CAR": "http://osrmvalenciacar:5004/route/v1/car/",
                "BIKE": "http://osrmvalenciabike:5005/route/v1/bike/",
                "WALK": "http://osrmvalenciawalk:5006/route/v1/foot/"
            },
            "available_features": [
                "CAR",
                "TRANSIT",
                "WALK",
                "BIKE",
                "PARKING",
                "BIKESHARING"
            ]
        }
    }
}
```

Listing 6.1: City Settings Configuration

This configuration comprehends, for as many cities as the system should embrace:

- **center** - the coordinates to center of the city.
- **circle_range** - the range on which requests should be mapped to that city.
- **routers** - the URL used in the requests of the routing components, on each mode (Public Transport, Car, Bike and Walk).
- **available_features** - the list of features the route planner should suggest to the user.

As an example, there could be declared in the configuration, to the same city, an internal OSRM server city, and an external OTP. As long as the communication between the systems follows OSRM's and OTP's API standards, it takes place seamlessly.

6.7 - Price and Distance Calculation

In order to complete the analysis of each route options, computed in response to a route request, there was necessary to assess the distance and price of a set of route steps.

Calculating the distance of a route option was considerably simple; both used routing engines provide distances for each step. There was also the need to save the distances by each transport mode, which was done and saved in the model of the route option. This distances, differentiated by mode, were used in the route analysis algorithm.

The calculation of the price of each route option was somewhat more difficult. To better explain this process, it should be divided into multiple explanations:

- Regarding the Walk and Bike modes, the price was considered 0 per hundred kilometres. This is due to the fact that Bike was considered on a user-owned bike, without additional tariffs associated.
- When analysing the Driving mode, the price was calculated using a user-defined *price per hundred kilometres* variable in its preferences, and the total distance of driving in the route option.
- Regarding Bikesharing, and using different tariff methods in the different cities, the model BikesharingPrice was idealised: it comprehended tariffs where some minutes at the start were not charged, and the different prices applied after the free period (or when this is not applicable, right from the start), as well as the tariff time rate. In practice, it also uses the user-defined variable *bike_speed*, which can be relevant since different bikes reach different speeds, which the user may configure.
- When analysing Public Transport, the price is applied on the basis of 1 tariff per bus or metro trip. When a route option has multiple public transport trips to fulfil the trajectory, the respective tariffs accumulate.
- When a combination of the said modes is presented to the user, the above rules apply to each division of the route option.

It should be duly noted that the prices of Public Transport and Bikesharing are based on information resulting from the research on city hosted websites, tourist reports and open data available. This pricing information should be confirmed before any deployment as a product but, being considered a main part of the route analysis, was integrated and used, even if it may not correspond to the prices billed in the cities, at this moment.

6.8 - Route Option Analysis and Matching

The computation of all the options for a determined route request happens with little input from the user preferences; some variables like *max_walk* and *max_bike* are used to force a transport route to have a smaller foot distance, but should not be blocking. These should be kept in mind, but overridden when necessary, extending that value, in order to not compromise the normal behaviour of the OSRM and OTP. It is useful, in another hand, to never compute routes based on the usage of a car, if the user does not have one (and the same applies to a private bike). Those are the only considerations on the user preferences while computing the route options.

As expressed in the remaining report, one of the most important objectives of this project was to truly include the user desires and likes in the usage of the platform. With that in mind, an options analysis process happens after all the route options are computed. This focus in a series of main purposes:

- Then routes that don't match the *max_walk* and *max_bike* values of the user, should never be considered more relevant than the ones that do.
- The user history of the last 10 chosen trips should be considered, and weight significantly in the actual decision to be made (if the user wants to).
- The user preferences towards modes of transport and speed, distance, and price should be considered (if the user wants to).

With those main purposes in mind, the analysis should provide a list of the more user adequate options, in order of relevance. This process was implemented, building indicators of the suitability of the option, based on the following factors which, to represent the most suitable option, should be minimised:

- How close is a given route option, considering a given transport mode, to the user history, considering the same transport mode.
- How close is a given route option, considering a given transport mode percentage, to the user preference.
- The difference between a given route's necessary time, considering the smallest necessary time among all route options and the user's speed preference (ranging from 0 to 1).
- The difference between a given route's distance, considering the smallest distance among all route options and the user's distance preference (ranging from 0 to 1).
- The difference between a given route's price, considering the cheapest among all route options and the user's price preference, times the users' preferences (ranging from 0 to 1).

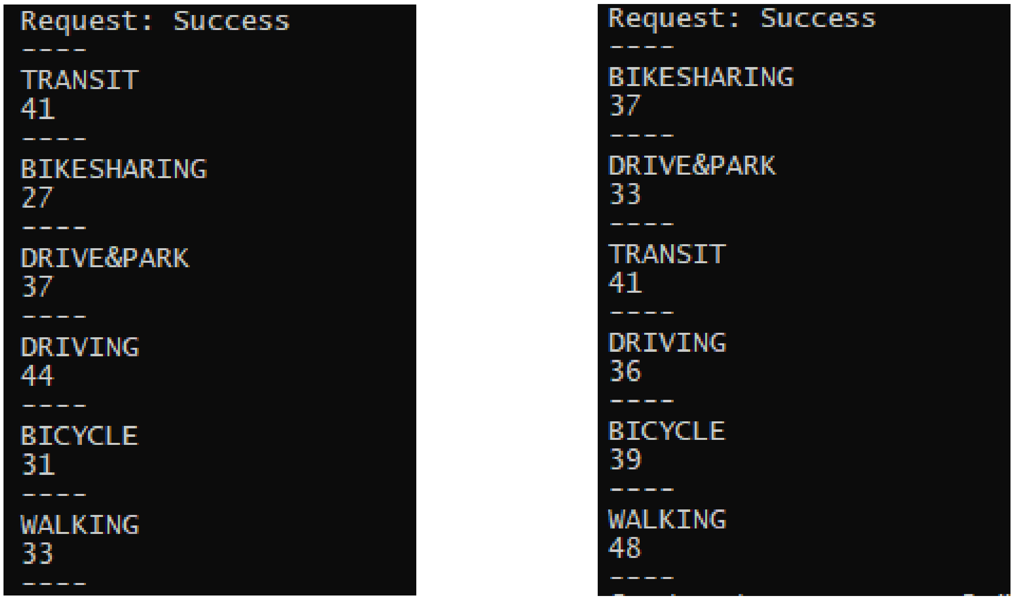


Figure 6.7: Examples of Output of Route Analysis

Depicted in Figure 6.7 are an example of the indexes output of the route option analysis. As said, the lower the number, the more it fits the user's preferences. Therefore, the system response contains a set of route options, ordered by increasing index.

6.9 - Showcase Frontend

To better demonstrate the applicability of the developed project, as well as allowing a better testing during development, a simple web page was developed, using HTML, JavaScript, AJAX requests and the necessary CSS. Using the open source map framework Leaflet, both widely used among the online community, and frequently used in Ubiwhere's products. This small development task, used only to demonstrate the capabilities and the possible interactions with the system, quickly grew thanks to the options necessary to test the system.

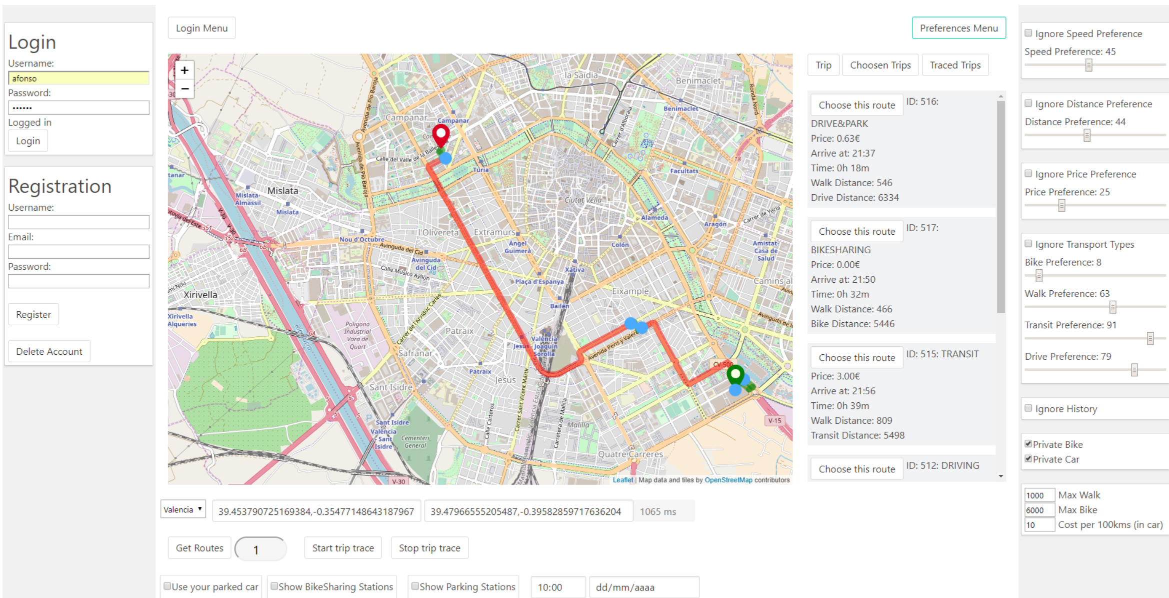


Figure 6.8: Overview Screen

In Figure 6.8, is the overview of the frontend developed. One can see the login, registration, preferences sliders and inputs and, in the main area, the map and options to customise the request.

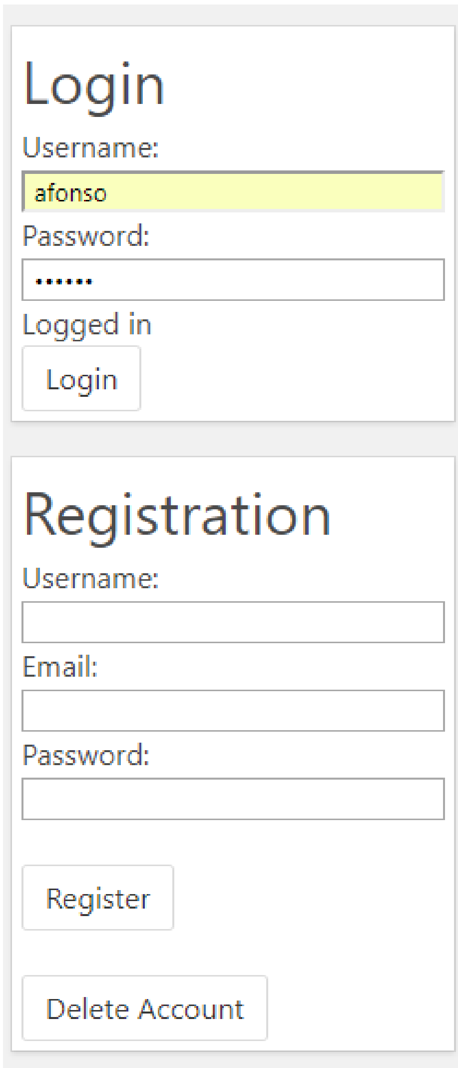


Figure 6.9: Login Screen

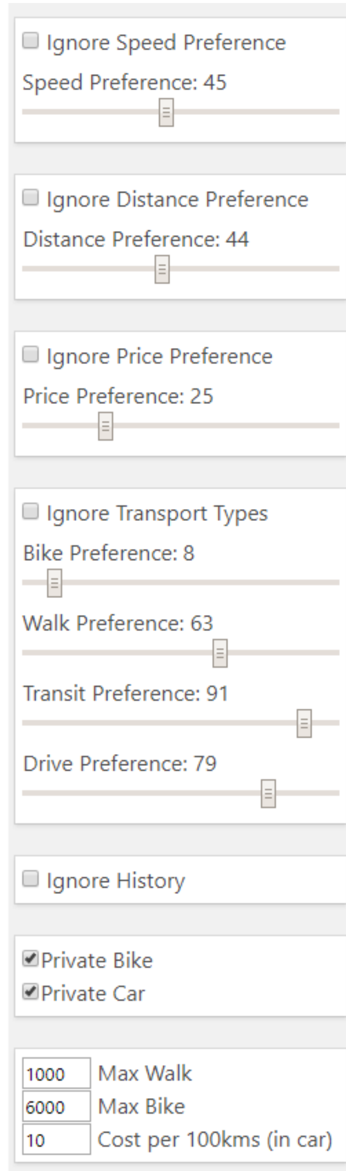


Figure 6.10: Preferences Screen

Both in Figures 6.9 and 6.10, are presented the sidebars. The first allows the user to create an account, log in and delete its account. In the second, are visible the multiple sliders, inputs and checkboxes, allowing the user to customise the account preferences. The user can also choose to ignore some of the fields, that will have an impact in the route option analysis and when are changed any of the values, both the preferences in the server are updated and the requested display is made again.

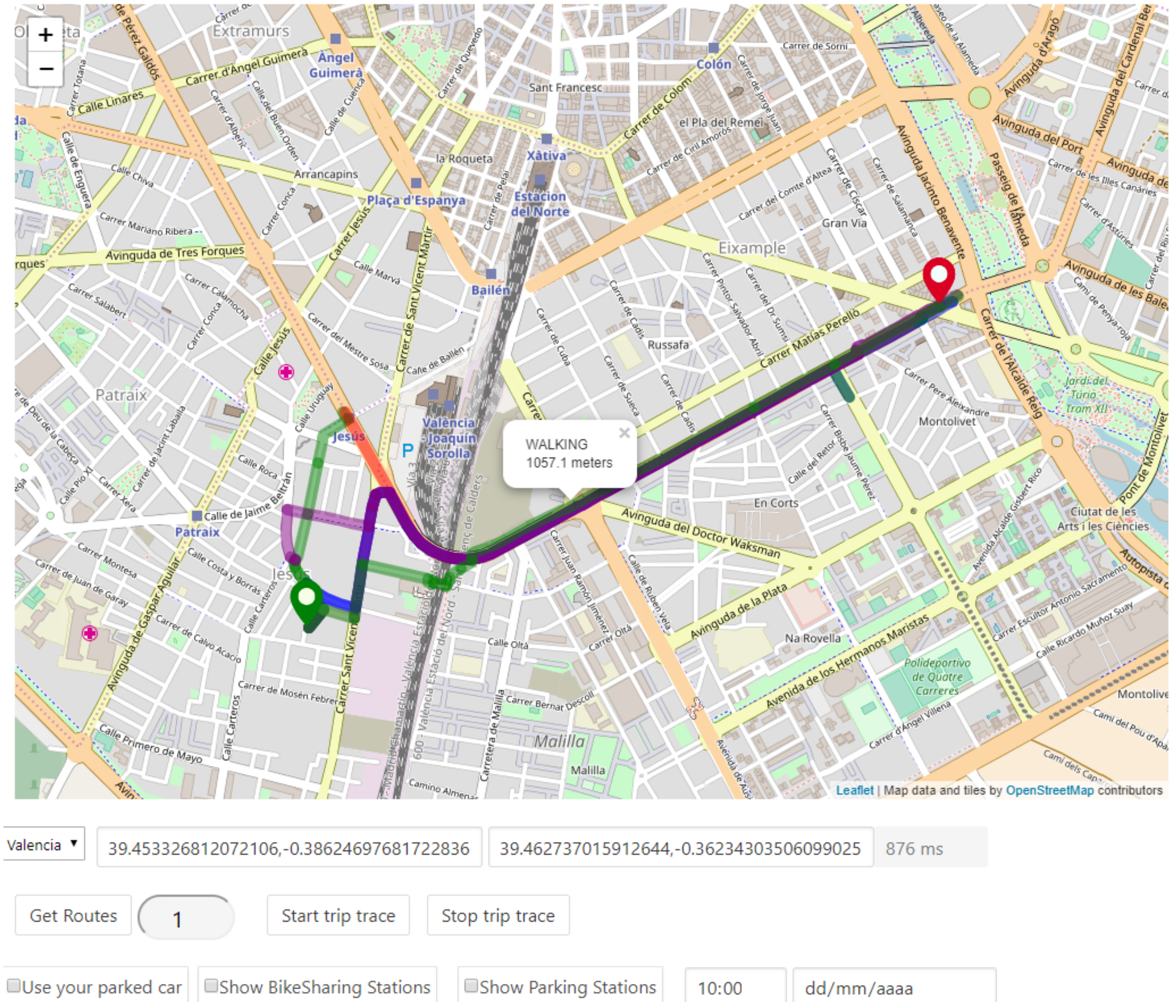


Figure 6.11: Dashboard Screen

In this dashboard, the user can click the map, setting both the origin and the destination. Right-clicking also brings up a context menu, allowing it to set waypoints, update his private car position and send real-time route traces (for testing purposes).

Below the map, are visible other inputs and pieces of information: a selector for the city to display, a text field that is updated with the number of milliseconds taken to process the request, as well as the number of different itineraries from each mode desired. Are also present the buttons to start a real-time route trace, as well as stopping it.

The checkboxes, in the bottom, allow for customisation of the presentation of the points of bikesharing and car parking in the map, as well as an option to use his private car in the route required. On the right part, are also present the hour and date inputs, which allow the user to specify the departure specific time, which force the routing planning components to use that specific hour and date, as long as the data available allows it.

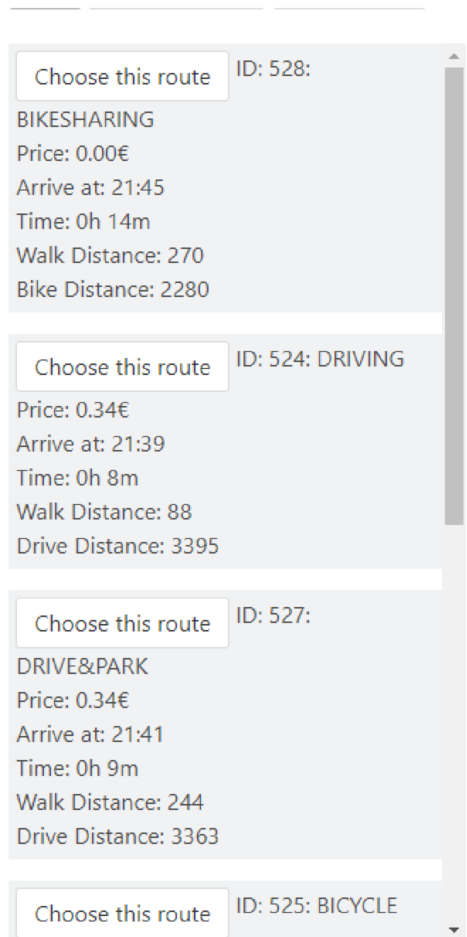


Figure 6.12: Route Options Screen

It is visible in Figure 8.2 the way the route options are presented to the user. The most important information is presented,such as the mode, price associated, the hour of arrival, the time necessary and the distances in the different modes. Clicking one of the cards, makes the route be displayed on the map, as well as the associated markers and points of interest. The user can also click the *Choose this route* button, which chooses this route among all the requests options. Afterwards, the chosen route is also visible in the ”Chosen Trips” tab, at the top of the page.

Chapter 7

Testing

In this chapter are described the tests performed on the system. They can be divided essentially into two groups: the tests performed to the API, testing the creation of the instances, along with the protection of inputs, and the tests performed to the route planner component.

7.1 - Unit and Flow Testing

Inside the simpler group of tests performed on the API, the aim was to assess the following:

- The system only accepts the requests that come with the Authentication Header;
- The system only creates the entities, which data comes from a request, when the data is fully validated, among both the rules imposed on the model, but also the system definitions. As an example, the system should reject the requests that don't belong to the configured cities.
- The endpoint usage flow should be respected. As an example, in the case of a real-time route, the tracepoints can only be added, with an open tracing route.

The tests can be presented in a table, seen in Tables 7.1, 7.2, 7.3, 7.4 and 7.5:

Num.	Component	Description	Passed
1	Account	The user can create an account.	Yes
2		The user cannot create an account with a username already in the system.	Yes
3		The user cannot create an account with a email already in the system.	Yes
4		The user can delete his/her account.	Yes
5		The user can change his/her preferences of the range (0 - 1) successfully.	Yes
6		The user can change his/her preferences of the range (0 - 1) to a value outside of the due range.	Yes
7		The user can change his/her preferences binary (true - false) preferences.	Yes
8		The user can change his/her max_walk and max_bike, to an integer number equal to 0 or higher.	Yes
9		The user cannot change his/her max_walk and max_bike, to an invalid number.	Yes
10		All the requests expect login and register must be accompanied by an authorization token.	Yes

Table 7.1: Test Suite 1

Num.	Component	Description	Passed
11	Car Sharing	The user can submit a carsharing ride with valid origin, and destination.	Yes
12		The user cannot submit a carsharing ride if the origin and destination fields are incorrect, or if the points don't belong to a supported area.	Yes
13		The user can search for carsharing rides with valid origin, destination and range.	Yes
14		The user cannot search for carsharing rides if he/she provides a invalid origin, destination and range.	Yes

Table 7.2: Test Suite 2

Num.	Component	Description	Passed
15	Realtime Routes	The user can start a new realtime route.	Yes
16		The user can end a available realtime route.	Yes
17		The user can add traces to a realtime route.	Yes
18		The user cannot end a available route, without submitting the required valid fields.	Yes
19		The user can see a list of his/her traced routes.	Yes
20		The user cannot insert a trace, without submitting the required valid fields.	Yes

Table 7.3: Test Suite 3

Num.	Component	Description	Passed
21	Route Planner	The user can submit a new route planning request, inserting the required valid fields.	Yes
22		The user cannot submit a new route request whose locations are outside of the cities available.	Yes
23		The user cannot submit a new route request with invalid inputs.	Yes
24		The user can access his/her chosen route options.	Yes
25		The user can access his/her chosen route options.	Yes

Table 7.4: Test Suite 4

Num.	Component	Description	Passed
26	Other	The user can access all bikesharing stations in the system.	Yes
27		The user can access all car parks in the system.	Yes

Table 7.5: Test Suite 5

With the above test suites implemented, it is safe to say that the endpoint validation, as well as the flow of the creation and behaviour of the REST API, is tested and according to Ubiwhere's standards.

7.2 - System/Request Testing

While the REST API can be tested and validated, there are also additional characteristics important to test in this particular project. Since it was developed to respond to real-world mutable needs, is of great importance to test the quality of the response of the system, when a given input is entered, but it should also be considered remarkably difficult to do so.

Bearing this difficulty, the showcase frontend was developed, in part, because there is no better analysis that the human one; only a true user would be able to distinguish between a route option suitable for the request, and a slightly worst one (at least, without some lengthy investment in artificial intelligence and machine learning, taking large amounts of investment and time). This frontend proved useful into its job: it allowed both testing and also fine-tuning of the multiple variables used in the computation of the route options, which allowed the final internship product to be a solution even closer to the possible users' expectations.

Another way to test the system response, was to create testing scripts that map the city to a set of variables, corresponding to the coordinates, and simulate multiple requests that aim to test both system response times, and request-response: it assesses the percentage of the requests that contain a given mode.

An example of input is visible in Figure 7.1, for a division of 4; each intersection of the lines, represents a point. To each of the points, there is made a request to each one of the remaining ones. In total, for a division of 4, 256 requests are performed.

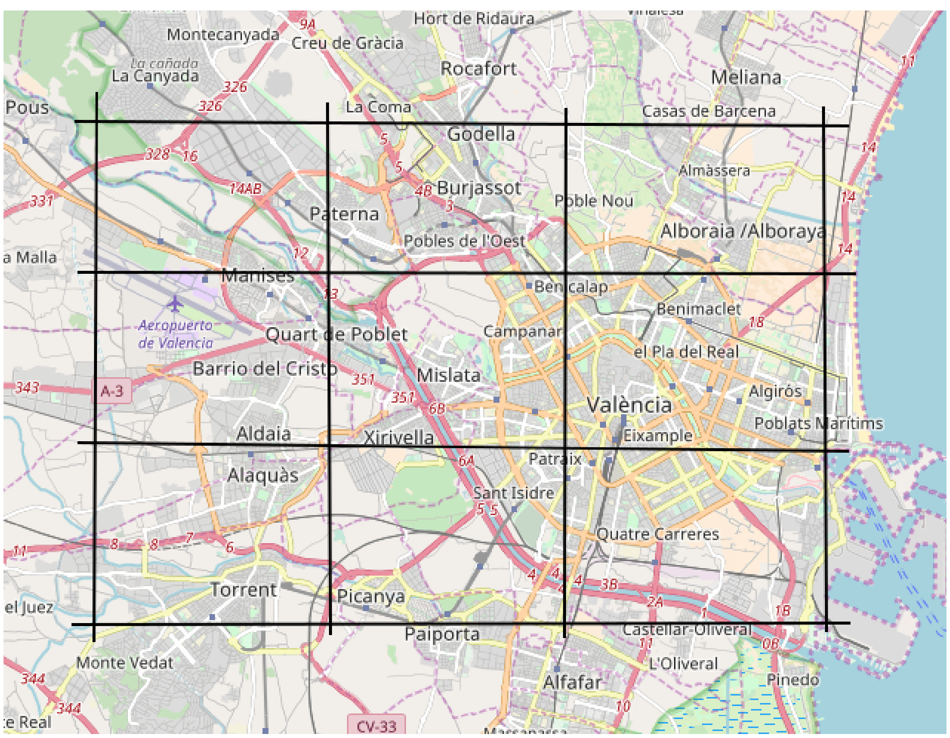


Figure 7.1: Map Testing Grid

As the scenarios indicate, the requests are divided into two groups:

- **Simple** - represents minimal calculations, and serve a set of multiple transport modes, independent.
- **Complex** - represents more difficult computation requests, which exchange of transport modes mid-journey and calculation of best paths multiple times, for each trip.

```
-----------
Processing time in average (ms): 3044
-----------
Test: 240 out of 256
Summary:
WALKING: 240                          Percentage: 100   %
BICYCLE: 240                          Percentage: 100   %
DRIVING: 240                          Percentage: 100   %
TRANSIT: 132                          Percentage: 55    %
PARK&GO and BIKESHARING: 32           Percentage: 13    %
PARK&GO and TRANSIT: 32               Percentage: 13    %
PARK&GO and BIKE: 32                  Percentage: 13    %
DRIVE&PARK: 90                        Percentage: 37    %
PARK&GO and WALK: 32                  Percentage: 13    %
BIKESHARING: 72                       Percentage: 30    %
```

Figure 7.2: Testing Scripts - Simple Requests

```
Processing time in average (ms): 6186
-----------
Test: 240 out of 256
Summary:
BIKESHARING AND DRIVE: 180          Percentage: 75    %
TRANSIT AND DRIVE: 180              Percentage: 75    %
BIKE AND DRIVE: 180                 Percentage: 75    %
WALK AND DRIVE: 180                 Percentage: 75    %
WALKING: 240                        Percentage: 100   %
BICYCLE: 240                        Percentage: 100   %
DRIVING: 240                        Percentage: 100   %
TRANSIT: 132                        Percentage: 55    %
PARK&GO and BIKESHARING: 32         Percentage: 13    %
PARK&GO and TRANSIT: 32             Percentage: 13    %
PARK&GO and BIKE: 32                Percentage: 13    %
PARK&GO and WALK: 32                Percentage: 13    %
DRIVE&PARK: 90                      Percentage: 37    %
BIKESHARING: 72                     Percentage: 30    %
```

Figure 7.3: Testing Scripts - Complex Requests

In both Figures 7.2 and 7.3, is visible the output of the tests performed on the system. The list of modes, followed by the percentages, indicate the quantity of requests a given mode was present in.

Since the requests points, in many of the tests, are outside of the acclaimed usage area (like suburbs and areas several kilometres away from the city), its considered that the tests certify the performance of the system, in both simple (with an average processing time of 3044 milliseconds) and complex (with an average processing time of milliseconds).

Chapter 8

Use Cases

While the internship focused on developing a door-to-door mobility system, one of the objectives was to use the multiple system sub-parts to support other Ubiwhere's projects. This proves that not only the project stands as a viable future product, but that the development guidelines of modularity and organisation helped raise the project value.

8.1 - Implemented Use Cases

In order to develop a project with real use cases, Ubiwhere provided information on both cities of Cologne and Valencia. Those cities are very different in both morphology organisation, street and city planning and in terms of the transport infrastructure available. This allowed the conditions for good testing cases on the system, as seen in Section 6.9

The data used in the system, for the city of Cologne, allows the usage of public transport, walk, driving and bike modes. This is typical in the mobility platforms, providing the usual features among this type of products available.

The project shines when used in the city of Valencia; the combination of the same features available in Cologne, but also the bikesharing and car parking locations, allows for an environment where the user gets access to features that distinguish the development product among its competitors.

8.2 - Real Use Case (Thumbeo)

One of the integration cases was the usage of the carsharing component to be integrated in a corporate project developed by Ubiwhere, named Thumbeo. This application allows the users to submit the trips they are going to do and, after that, when other users search for a carsharing possibility, the system responds with the matches by proximity on both the ends of the trip and in intermediate points. Calculated to each trip are also its details (distance and time necessary) as well as a geometry to represent the trip on a map framework.

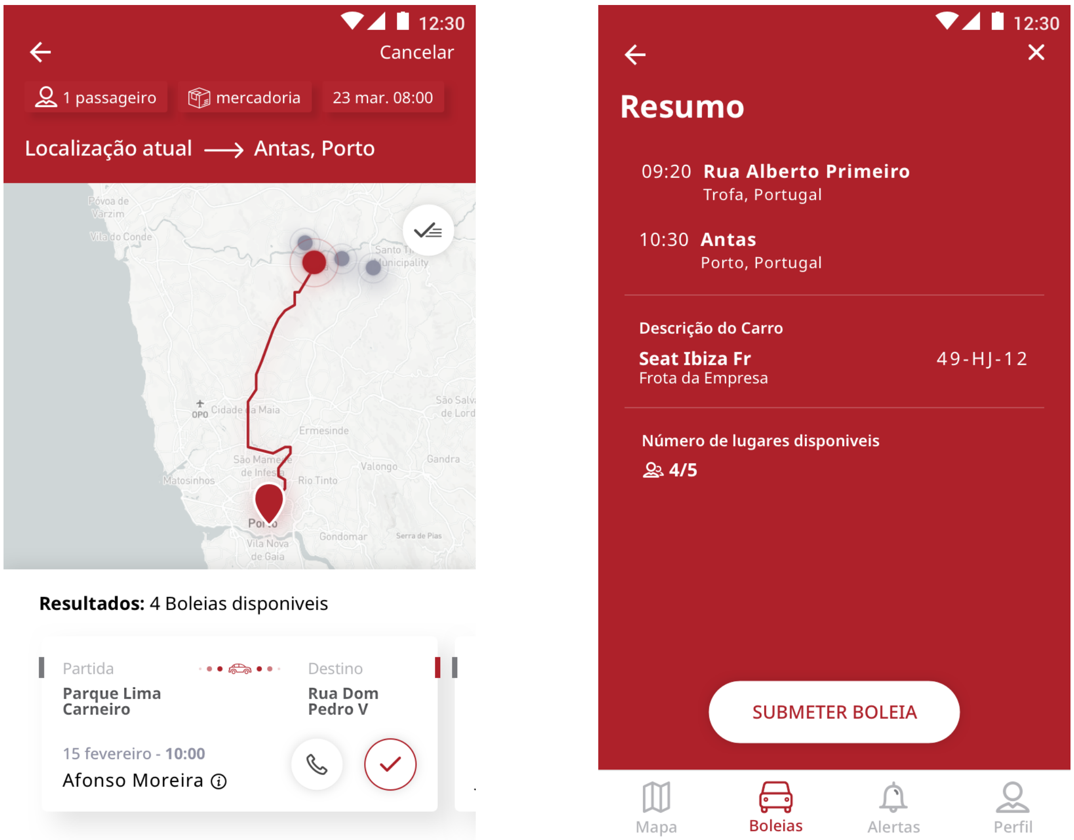


Figure 8.1: Thumbeo Screens

8.3 - Real Use Case (Trace Routes)

Another real case for usage is seen on a new platform, Mobility as a Service, developed by Ubiwhere. The real-time route component, developed and tested in the scope of the internship, is used to execute part of the backend needs of route generation based on real-time traces.

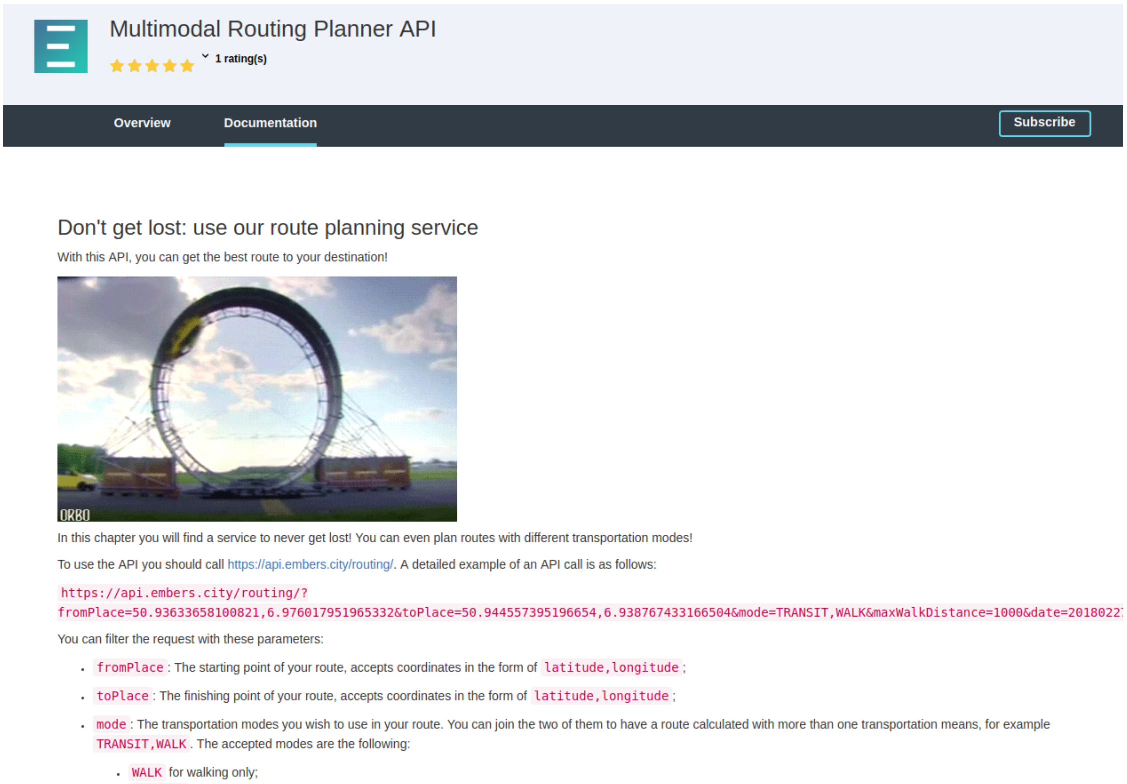


Figure 8.2: Embers City Mobility as a Service page

The application of the real-time component in mobile applications can be in the near future, as it allows devices to send periodically their position to the system and, when desiring to end the real-time route tracing, receive a route geometry that can be displayed on a map. This allows the user to use the system to store a history of his/her traced routes, which can be retrieved at any time.

Chapter 9

Final Product of the Internship

In this chapter, an analysis on the final product of the internship is made, with an overview over the fulfilment of the requirements, correspondence with the competitors, success of the system as a proof of concept and the general opinion of the hosting company on the subject development.

9.1 - Correspondence to Functional Requirements

As seen in the Chapter 4, the requirements were expressed both with user stories, a theoretical and systematic approach, and personas, a more practical and close approach to the end user. To evaluate the system, the best approach is to check if all the personas desires and usages are matched. With that in mind, it should be duly noted that all of the personas action should be based on the backend part of the system, since not all features are usable in the showcase frontend developed. With that in mind, the information can be therefore organised, as follows:

Persona 1

Name:	**Maria**
Goals:	Wants to assess if his new school is within a distance she considers viable to make using a bicycle, as well as comparing it to public transportation, both in terms of travelling time, speed and price.
Resolution:	Since Rita can see how many minutes it takes her to reach her destination using her bicycle and public transport, she can know each one is the faster option. She can also consult the prices and see at which hours she should leave home to take the bus.

Persona 2

Name:	**Pedro**
Goals:	Wants to assess if there is a way to make his daily commute using bikesharing, otherwise, he will take public transportation to do so, and would like to know which routes to take and the hours to do so.
Resolution:	Using the platform, Pedro can not only see if his commute is possible, within his limits to travelling on foot to a bikesharing station, but also see the multiple stations where to pick his bicycle and the ones where to drop it off, calculated using multiple factors, if Pedro requests more that one option for each mode.

Persona 3

Name:	**José**
Goals:	Would like to know not only the best way to reach the old part of the city using public transportation. If possible, he would also like to know if there a car park as close as possible, within the city range, so it can have a less stressful visit to the city.
Resolution:	Since José request origin is outside of the range of car parks, and the destination if inside of it, the system suggests options where the user should drive to the closest car park, and then take public transportation to his destination. If José states that his car is parked in the city, the system is also able to suggest him the opposite: use public transportation to his car and then drive to the point outside the city.

Persona 4

Name:	**Ricardo**
Goals:	As he is travelling to a city soon, would like to know the best way to reach the hotel and different cultural sites around the city, as well as how to reach the airport.
Resolution:	The system uses all the options available to suggest the multiple options to Ricardo, as well as the prices and time necessary to use each one.

Persona 5

Name:	**Helena**
Goals:	As she is travelling across the country, and many cities have no indication about car parks, she would like to know both the way and the car park closer to her destination. Because she drives the car multiple times per day, she would like to keep track of her car in a map and use a system that considers the updated position whenever she wants to get directions for the next destination.
Resolution:	The system suggests to Helena both the driving route, but also the closest car park to her destination. When leaving a city, the system also uses her car position, indicating directions on foot (or other modes, if it makes sense, considering the distance) to her car, and the directions to be taken next.

Persona 6

Name:	**Rogério**
Goals:	As Rogério is about to make a trip between cities, he would like to know if any other coworker is in need of a carpool to a city in its path or the other way around.
Resolution:	The system, considering the existent car sharing trips in the system, returns to Rogério the list of the ones that match his preference.

Persona 7

Name:	**Rita**
Goals:	Wants to check the time and details of its commute trip, using public transportation, at a different hour and day of the week that she is used to, allowing her to better prepare her day in advance.
Resolution:	Rita, when making her request, can select what hour and date she wants to start her trip; the system uses those inputs to calculate the trips using public transport, which means that all trips are adjusted and suited to the real-time schedules practised; in other hand, is visible that some routes have no nocturne schedule, as an example of the use of this. The usage of the system also allows Rita to understand how different train lines can be used in the same option, being able to see the hours at which she should take the trains.

With all the personas idealised being satisfied, the system corresponds to the needs that have been assessed. Considering the user stories initial and final iteration, and considering that all ”Must Have”, ”Nice to Have” and all ”Could Have” user stories have been implemented, the system fulfils all the assessed user needs.

9.2 - Correspondence to Quality Attributes

To better analyse the system behaviour, the scenarios were a suitable way to better regulate the development and, in a final phase of the internship, assess if the system corresponds to the expectations in terms of quality, modularity and performance. Following is presented the list of the quality attributes taken into account, as well as a small description of the way it has shaped the architecture, system, and coding practices.

- **Maintainability** - The system was designed with as much effort into good coding practices and smart organisation as possible. In this process, the code review and help in the decision making, with the help of the hosting company employees and internship colleagues was crucial.
- **Reliability** - The usage of docker virtualization allowed and effective debugging process, as well as understanding on flaws that can happen to this system in a real-world scenario. While error preventing reliability enhancing practices have been implemented, the error reporting ones are to be implemented at a later stage in the final product development roadmap.
- **Availability** - The availability is a quality attribute only measurable in the deployed environment (after all, the downtime can be forced by external conditions to the project). With that in mind, the measure appointed can easily be met using Ubiwhere's deploying model, using Docker images and a registry to allocated pre-compiled images.
- **Interoperability** - This quality attribute is met using error awareness, using external systems, and a periodic save of the information, allowing the secure execution of the system processing, using information in store in databases.
- **Performance** - The system response times are as expected; below the maximum appointed in the performance scenarios. This is also one of the system strengths: as the multiple execution time reducing techniques were used, the system developed proved capable of handling even the most complex requests (previewed for the internship) in a stable manner and taking less than the imposed time.
- **Scalability** - The system, while not fully scalable at the closure of the internship, is characterised by a scalable architecture that, when used in conjunction with load balancing and replicating architectures, is capable of delivering the features as stated in the scenario. The implementation/deployment of such product is the kind of work that requires multiple factors taken into account, as well as a time quantity simply not available in the terms of the internship. With that in mind, the hosting company admits that the system quality, in terms of scalability, is a good step in the delivery of a final, fully scalable, product.
- **Security** - The input parsing used in Django, using serializers and hard-coded validation, verifying all the requests sent to the backend, also provides a good layer of security against bad and malicious use of the system. With that in mind, also the user authentication is made using JWToken and the authorship of the following requests is maintained.
- **Testability** - While this was considered a hard quality attribute to meet, giving the hard testability of a mobility planner. As reported in Chapter 7, there been implemented unit tests and a testing script that measures both processing time and the response of the system. However, this does not substitute completely the pressure tests that a final developed product should be put through.

9.3 - Hosting Company Expectations

Considering the usefulness of both the internship product and investigation, the hosting company expectations are considered to be met. The investigation around the concept allowed a better view of the competitors market, relation with the cities and public approach. The several months of work allowed also a different point of view of the mobility problem from the typical research and business-oriented innovation: the act of developing a product for the sake of investigation and possibilities exploration allowed a bird's eye of view over the problem, without business constraints. At the same time, a person's opinion, outside of the concept development, allows the sedimentation of the product's real user expectations, as an example.

Regarding the project development, Ubiwhere's expectations were also met; the development not only followed the development guidelines imposed to the intern, but also the process managing took place as smoothly as possible. After a brief analysis of the features developed, it's easily noticed that the system largely fulfilled of the project success criteria.

With that in mind, the intern's research and development generated features, plans, and suggestions that do match hosting company's futures plans for the product. This is the most important motivation, at the eyes of Ubiwhere, to hosting an internship: the ability to have a fresh vision on the concept, with time dedicated to a formal research, problem study and assessment, is largely appreciated and extremely valuable.

Chapter 10

Conclusions

In this chapter are presented the conclusions of the internship, and is mostly an analysis of the process, project roadmap and development and integration in the company. There is present also a reflection on the intern development and the obstacles encountered.

10.1 - Realization

Considering the internship's process, and the amount of effort put into all phases, the internship's results were a success: the concept development proved relevant, the practical use cases implemented are useful and representative and the product value added to the hosting company is a nice stepping stone to the development of future solutions.

With that in mind, the internship contributions can be listed, for a better understanding of the project:

- Research on solutions and competitors in the market, as well as data providers and business cases.
- Development of a modular route planning backend, that matches user preferences to better correspond to a city transports user.
- Development of a real-time trip tracing backend, to be integrated with other projects.
- Development of a mobility backend to support the a car sharing platform, to be integrated with other projects.
- Contributions of the possible uses and requirements of the future mobility solution.

It should also be noted that all the development is according to the hosting company standards, which improves the quality of the contributions.

10.2 - Obstacles

At this point, the evaluation of the obstacles is an important part of evaluating a project. Following, are described both process and development obstacles, as well as the workarounds and solutions proposed and put into practice.

- The technologies used, on several occasions, proved a lack of documentation that compromised the development process. In some cases, the solution came from an experienced Ubiwhere employee, which also proves the support that the hosting company was always ready to provide.
- Since the project was considerably large, in terms of development, the time schedule had to be adapted and decisions were taken to minimise the possibility of internship process compromise. However, the development results don't deviate from the expected, at the end of the internship.
- The data unavailability, as feared and analysed in the risk management section, proved true only in the first stage of the internship. Since the data was available from the moment it was necessary to support the development, this risk associated problem existed but reflected in no damage to the internship process.
- The massive architecture focus necessary to develop a successful project was threatening, due to the lack of background knowledge, by the intern. This resulted in a more extended study phase and more effort on the subject that, as should be noted, was always supported and under supervision by the Ubiwhere advisor.

The obstacles of the internship are a relative subject, however. Those have been taken into account at both internship plan inception and requirement specification phases and, after an analysis in the work done, didn't compromise the success of the internship.

10.3 - Lessons Learned

Since the internship was one of the main contacts of the intern with a company workplace and team inclusion, there was obviously a development of the soft skills necessary to work in this area. Skills like time management, scrum development, as well as communication and presentation skills were greatly developed during the nine months that the internship took.

The architecture of the project, developed with supervision from the Ubiwhere advisor, also allowed a great skill development: problem evaluation and solution design. That is one of the greatest achievements of the internship, to the intern: a real case of architectural analysis and development of a system, especially this size, usually does not occur in the student degrees taken by the intern and represented a great addition to his skill set.

When making an analysis on the hard skills developed in during the internship, the extended usage of Django, as well as the indexed testing and development techniques, allowed the gathering of great proficiency on this framework. In another hand, the usage of Celery was extent enough for the intern to feel comfortable in this usage, at a professional level. Since all the work on this hard skills was under supervision and review by a Ubiwhere employee, those skills are also validated.

10.4 - Future Work

Looking at the development of the project, an already impressive example on how can a mobility system like this radically diversify the way a commuter or a tourist looks at the city transport, there are also some guidelines set for the future, based on this project development, that can even distinguish this solution, reflecting themselves in features but also architectural or implementation ones, as the most valuable one among his competitors. They are, quite simply, as follows:

- The system, while based on an updated data from several sources, lacks the real-time update on the state of the city. As one can notice on a daily basis, it is relevant to understand the city as a mutable transport network, and there is of huge importance the use of data on events to adapt the route planner algorithm. As an example, if the city maintains information on the streets that are closed or constricted due to maintenance, a user should deviate from that possible route option.
- While the developed frontend is enough to demonstrate the project capabilities, a website or mobile application should be developed from scratch.
- A true recommendation system can and will be developed in the future. This can be implemented as an external application or an internal one. The process should be effective to change the results from the users' requests, but also spread its effect to other users: in a practical example, the reluctance of the user on using a given road, can be applied to the other users. This component should receive daily batches of information, digest it, build indicators and change the route planner behaviour based on those indicators. The recommendation system is a difficult task to implement, not only due to the necessary implementation of a good and responsive data digestion component but also the application of its results in the route planner system, which is a challenge Ubiwhere is already taking head-on.

Bibliography

[1] *How Perth managed to increase bicycle commuting by 28,6 per cent*. https://www.bikecitizens.net/perth-cycling-right-direction/.

[2] M. Kracheel, R. McCall, V. Koenig, and T. Engel, “A multimodal mobility behaviour logging methodology,” *Proceedings of the 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications - AutomotiveUI ’13*, no. October, pp. 254–257, 2013. http://dl.acm.org/citation.cfm?id=2516540.2516573.

[3] Science for a Healthy Planet and a Safer World, *Car Emissions and Global Warming — Union of Concerned Scientists*, 2016. http://www.ucsusa.org/clean-vehicles/car-emissions-and-global-warming.

[4] M. Merforth Istanbul, “Codatu XVI 2015 SUMP-NAMA Roundtable Urban Mobility Plans as strategic tool for local and national level authorities,” 2015. http://www.codatu.org/wp-content/uploads/PRESENTATION-Urban-Mobility-Plans-as-strategic-tool-for-local-and-national-level-authorities-EN.pdf.

[5] S. Gautam, R. Singh, and S. S. Yadav, “Coupling And Cohesion,” *International Journal of Computer Technology & Applications*, vol. 4, no. 1, pp. 155–161, 2013, ISSN: 2348-795X. https://courses.cs.washington.edu/courses/cse403/96sp/coupling-cohesion.html.

[6] R. M. Lane Vanessa, “Door to Door: Combined Mobility and the Changing Transit Landscape,” *Transit Leadership Summit*, no. May, pp. 127–141, 2015.

[7] AGE Platform Europe, “Door-to- door transport systems : older people’s point of view,” Tech. Rep., 2002. http://www.age-platform.eu/images/stories/EN/transport_services_and_older_people.pdfhttp://www.age-platform.eu/policy-work/news/age-door-door-transport-systems-older-peoples-point-view.

[8] Y. Chao-Fu, “FIGURE 1: Level of motorization (automobile ownership) on national and regional scale,” *CODATU XIII*, pp. 2–13, 2007. http://www.codatu.org/wp-content/uploads/A-study-on-

feasibility-of-passenger-intermodal-transport-in-city-of-the-developing-world-Chao-fu-YEH.pdf.

[9] R. Rieland, *Helsinki Has a Plan to Get People to Stop Owning Cars*, 2014. http://www.smithsonianmag.com/innovation/helsinki-has-plan-to-get-people-stop-owning-cars-180952694/.

[10] MaaS Global, *Helsinki takes another pioneering step in mobility services: HRT public transport added to the Whim mobility app – MaaS Global*, 2016. http://maas.global/helsinki-takes-another-pioneering-step-in-mobility-services-hrt-public-transport-added-to-the.

[11] J. J. Trip and Y. Bontekoning, "Integration of small freight flows in the intermodal transport system," *Journal of Transport Geography*, vol. 10, no. 3, pp. 221–229, Sep. 2002, ISSN: 09666923. http://www.sciencedirect.com/science/article/pii/S096669230200008X.

[12] J. B. Gordon, "Intermodal Passenger Flows on London ' s Public Transport Network," 2012. https://dspace.mit.edu/bitstream/handle/1721.1/78242/830539087-MIT.pdf?sequence=2http://jaygordon.net/.

[13] "Intermodal Passenger Transport in Europe," *The European Comission's Directorate-General for Mobility and Transport*, pp. 1–48, 2015. http://www.mobiel21.be/sites/default/files/publications/Brochurelinkkleiner.pdf.

[14] T. A. Litman and T. Litman, "Introduction to Multi-Modal Transportation Planning Multi-Modal Transportation Planning," 2017. http://www.vtpi.org/multimodal_planning.pdf.

[15] T. Willgren, "Legal Aspects to MaaS," https://www.iru.org/system/files/18.WillgrenTuuliLegalaspectstoMaaS.pdf.

[16] Finnish Transportation Agency, "MaaS Services and Business Opportunities," 2015. https://julkaisut.liikennevirasto.fi/pdf8/lts_2015-56_maas_services_web.pdf.

[17] MuleSoft, *Microservices vs Monolithic Architecture — MuleSoft*, 2016. https://www.mulesoft.com/resources/api/microservices-vs-monolithic.

[18] L. Bershidsky, *Millennials Want Apps, Not Cars - Bloomberg*, 2014. https://www.bloomberg.com/view/articles/2014-07-15/millennials-want-apps-not-cars.

[19] Microsoft, *Quality Attributes*. https://msdn.microsoft.com/en-us/library/ee658094.aspx.

[20] B. Foulser, "Reimagine places: Mobility as a service," *Kpmg*, no. August, pp. 1–32, 2017. https://maas.global/www.kpmg.com/uk/reimagine-maas.

[21] M. Kamargianni, W. Li, M. Matyas, and A. Schäfer, "ScienceDirect A critical review of new mobility services for urban transport," *Transportation Research Procedia*, vol. 14, no. 14, pp. 3294–3303, 2016, ISSN: 23521465. http://ac.els-cdn.com/S2352146516302836/1-s2.0-S2352146516302836-main.pdf?_tid=ecb96984-9e25-11e7-b871-00000aacb361&acdnat=1505927318_cc4fe5c8b8c138d191e1590e0e32fdd8.

[22] N. Tschanz and H.-D. Zimmermann, "The Electronic Mall Bodensee as Platform for the Development of Travel Services," in *Information and Communication Technologies in Tourism - Proceedings of the International Conference in Innsbruck, Austria 1996*, Vienna: Springer Vienna, 1996, pp. 200–210. http://www.springerlink.com/index/10.1007/978-3-7091-7598-9_23.

[23] B. W. Goodall, T. Dovey, J. Bornstein, and B. Bonthron, "The rise of mobility as a service," *Deloitte Review*, no. 20, pp. 112–129, 2017, ISSN: 00130613. https://www2.deloitte.com/content/dam/Deloitte/nl/Documents/consumer-business/deloitte-nl-cb-ths-rise-of-mobility-as-a-service.pdf.

[24] ——, "The rise of mobility as a service," *Deloitte Review*, no. 20, pp. 112–129, 2017, ISSN: 00130613. https://www2.deloitte.com/content/dam/Deloitte/nl/Documents/consumer-business/deloitte-nl-cb-ths-rise-of-mobility-as-a-service.pdf.

[25] D. Banister, "The sustainable mobility paradigm," *Transport Policy*, vol. 15, pp. 73–80, 2013. https://ac.els-cdn.com/S0967070X07000820/1-s2.0-S0967070X07000820-main.pdf?_tid=112cffb6-c53f-11e7-967f-00000aab0f02&acdnat=1510226212_a0100918172fedd897bd397fb541b86f.

[26] ——, "The sustainable mobility paradigm," *Transport Policy*, vol. 15, pp. 73–80, 2008. http://www.civil.ist.utl.pt/~martinez/PDF/MobiCredit/Paper10.pdf.

[27] R. Tempier and P. Rapp, "Towards a European Multi-modal Jorney Planner," Tech. Rep. September, 2011, pp. 1–85. https://ec.europa.eu/transport/sites/transport/files/themes/its/studies/doc/2011_09-multimodal-journey-planner.pdf.

[28] C Chang and T Tan, "TRANSPORT SYSTEMS," vol. 8, no. March, pp. 171–179, 2009. https://ts.catapult.org.uk/.

[29] P. G. Giannopoulos, "Urban Mobility the door - to - door strategy URBAN MOBILITY : The aim," 2010.

[30] WHO, *Urban population growth*, 2015. http://www.who.int/gho/urban_health/situation_trends/urban_population_growth_text/en/.

[31] E. Buhl, "Urban transport 2.0 From A to B with smart connections," https://www.mobility.siemens.com/mobility/global/SiteCollectionDocuments/en/integrated-mobility/integrated-mobility-platform/urban-transport-2-0-en.pdf.

[32] B. Cohen, "Urbanization in developing countries: Current trends, future projections, and key challenges for sustainability," https://ac.els-cdn.com/S0160791X05000588/1-s2.0-S0160791X05000588-main.pdf?_tid=9421fc06-c53e-11e7-82a5-00000aab0f6b&acdnat=1510226002_91442c61c5357e04176b4c9c9a34bbea.

[33] E. Grey, *Waynaut: a new platform for multimodal transportation - Railway Technology*. http://www.railway-technology.com/features/featurewaynaut-a-new-platform-for-multimodal-transportation-4847967/.

[34] J. Cory, *What is Data as a Service (DaaS)? - Definition from Techopedia*, 2013. http://www.techopedia.com/definition/28560/data-as-a-service-daas.

[35] MaaS Alliance AISBL, "White Paper: Guidelines & Recommendations to create the foundations for a thriving MaaS Ecosystem," Tech. Rep. 2, 2017, p. 21. www.maas-alliance.eu.

[36] United Nations, *World's population increasingly urban with more than half living in urban areas*, 2014. http://www.un.org/en/development/desa/news/population/world-urbanization-prospects-2014.html.

[37] *23+ Advantages of Python and Django in Web Development*. https://keyua.org/blog/python-django-web-development/.

[38] *A million requests per second with Python – freeCodeCamp*. https://medium.freecodecamp.org/million-requests-per-second-with-python-95c137af319.

[39] "A truly integrated transport system for sustainable and efficient logistics," http://www.ertrac.org/uploads/documentsearch/id46/2017IntegratedLogistics-SETRIS.pdf.

[40] *ABI Research Forecasts Global Mobility as a Service Revenues to Exceed $1 Trillion by 2030*. https://www.abiresearch.com/press/abi-research-forecasts-global-mobility-service-rev/.

[41] "Country fact sheet Portugal Mode of transportation," 2015. http://www.esranet.eu/sites/default/files/ESRA2015CountryFactSheetPortugal_0.pdf.

[42] *door2door Platform*. https://www.door2door.io/platform.html.

[43] *EMBERS*. http://embers.city/.

[44] *EMBERS Documentation · Apiary*. https://emberscity.docs.apiary.io/#.

[45] *ERTICO ERTICO - ITS Europe*. http://ertico.com/.

[46] *Ertrac*. http://www.ertrac.org/.

[47] "Evolution of the transport connection across the border : the role of the TGV and the future of the lorries," 2013.

[48] *Finland to develop one-ticket mobility service network — Eltis*, 2016. http://www.eltis.org/discover/news/finland-develop-one-ticket-mobility-service-network.

[49] *First Steps with Celery — Celery 4.2.0 documentation*. http://docs.celeryproject.org/en/latest/getting-started/first-steps-with-celery.html.

[50] *FIWARE*. https://www.fiware.org/.

[51] *Fiware Data Models*, 2017. https://github.com/Fiware/dataModels.

[52] *Fleet sharing platform*. http://content.comtradedigital.com/fleet-sharing-platform.

[53] *Getting started with Django — Django*. https://www.djangoproject.com/start/.

[54] *Green Route*. http://greenroute.smartsdk.ubiwhere.com/.

[55] *How Perth managed to increase bicycle commuting by 28,6 per cent*. https://www.bikecitizens.net/perth-cycling-right-direction/.

[56] "Interconnected projects enabling Door-to-Door Mobility," https://eu-smartcities.eu/printpdf/1429.

[57] *Leaflet - a JavaScript library for interactive maps*. https://leafletjs.com/.

[58] *Mobility as a Service Alliance — MAAS-Alliance*. https://maas-alliance.eu/.

[59] *Mobility as a Service Alliance — MAAS-Alliance*. https://maas-alliance.eu/https://maas-alliance.eu/%0Ahttp://maas-alliance.eu/.

[60] *Mobility integration: a step closer to multi-modal, door-to-door travel*, 2013. https://www.automotiveworld.com/megatrends-articles/mobility-integration-step-closer-multi-modal-door-door-travel/.

[61] *OpenTripPlanner*. http://www.opentripplanner.org/.

[62] *Polis Network - Traffic Efficiency and Mobility*. https://www.polisnetwork.eu/working-groups/wg-mobility-and-traffic-efficiency.

[63] *PostgreSQL*, 2014. https://www.postgresql.org/about/advantages/http://www.postgresql.org/about/advantages/.

[64] *PostgreSQL: Documentation: 8.0: Tutorial*. https://www.postgresql.org/docs/8.0/static/tutorial.html.

[65] *Project OSRM*. http://project-osrm.org/.

[66] "Quality Attribute Generic Scenarios," http://sa.inceptum.eu/sites/sa.inceptum.eu/files/Content/QualityAttributeGenericScenarios-2.pdf.

[67] *Requests: HTTP for Humans — Requests 2.19.1 documentation*. http://docs.python-requests.org/en/master/.

[68] *TransitFeeds - Public transit feeds from around the world*. https://transitfeeds.com/http://transitfeeds.com/.

[69] *Whim app*. http://whimapp.com/fi-en/.

Chapter 11

Annexes

Available documents in annex:

- **Annex A** - Competitors analysis
- **Annex B** - Competitor products features
- **Annex C** - User Stories
- **Annex D** - Scenarios

Annex A - Competitors Analysis

Available MaaS Products

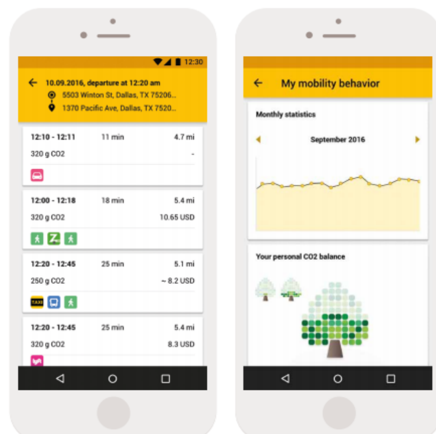
In this section are analyzed the biggest products in the mobility-as-a-service worldwide market. Their biggest features, offers and respective zones of operation are detailed, being them built as a product with profit in mind or implemented as a non-profit initiative.

A syntheses of the features of the available products/platforms is presented in the Table 11.1, in page 104.

FluidHub

Important features:

- Already available in 11 European and 2 American Cities
- Adapted Routing, Ticketing and realtime information system
- Multi-modal transport in city (Public Transports, Car/Bike Sharing/Pooling, Ride Sharing, Car/Bike Rentals, Taxis, other services)
- Available as a backend solution



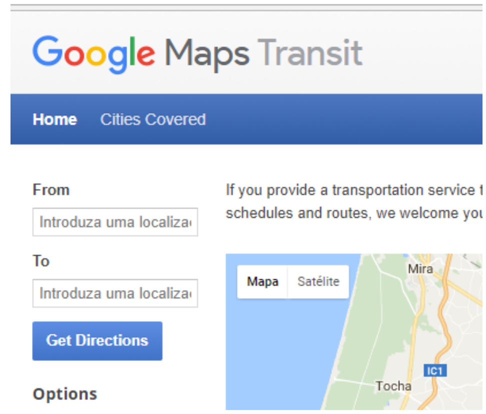
Competitor - Fluidhub

This platform, designed by FluidTime, aims to provide a better way to integrate a city's transport services, allowing for a MaaS solution to be implemented, suiting the region needs. This solution consists of a pure BaaS solution, put into action in different phases, starting as a pilot and culminating in a fully optimized solution for both its users and transport providers. Fluidtime's approach to a city mobility optimization issue, tackling the city organization and not the end user, proved intriguing but also effective, since already 13 developed cities have the project in use.

Google Transit

Important features:

- Free
- Routing System and Transit Alerts
- Multi-modal transport (Foot, Metro, Train, Bus)
- Available as a Website



Competitor - Google Transit

This platform developed by Google aims to provide a simple way to simulate a trip. While being more effective on long trips (as the bigger transport companies tend to cooperate better with an initiative as this one) than small ones and commutes, this platform is based on data submitted by the willing transportation operators. This way, the project lacks the information necessary to provide itineraries in smaller areas. The data that companies can provide is in GTFS format, suitable for this application.

Whim

Important features:

- Monthly Subscription pack (that includes the services)
- Multi-modal transport (Metro, Train, Bus)
- Car renting
- Available on Android and iOS

Competitor - Whim

This Helsinki area specific product, released in 2016 and developed by MaaS Global, aims to provide a better support for traveling in this Finish area, allowing a monthly subscription payment, with diverse package offers, to provide services like unlimited public transport access and car rental usage packs. The company is currently testing a second product, aimed for the United Kingdom, with the same features, so it a slowing growing availability is expected in the most developed countries in the mobility concept.

Waynaut

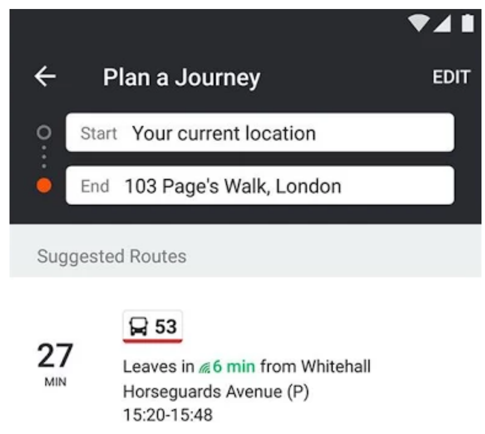
Competitor - Waynaut

Important features:

- Payment System for Public Transport
- Routing System
- Multi-modal transport (Foot, Metro, Train, Bus)
- To be available on Android and iOS

This platform, still in Early Access, has the goal designing innovative technologies to combine all means of transport into optimal and purchasable itineraries for travelers. Developed in Milan, and born in 2012, the product first was called *Youmove.me*, but was posteriorly adopted the current name. Winning multiple awards, it is expected a grand entrance in the mobility environment, when it is finally released.

Moovit



Competitor - Waynaut

Important features:

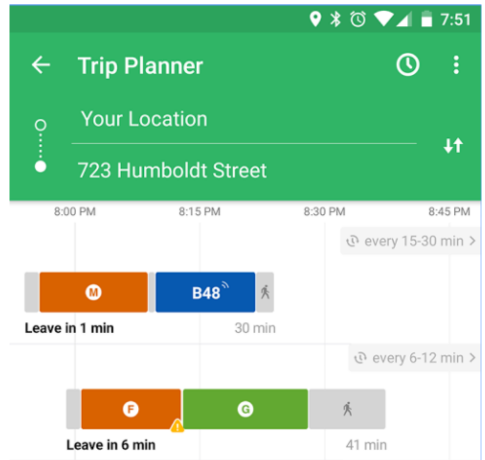
- Payment System for Public Transport
- Routing System and Transit/Delay System
- Multi-modal transport (Foot, Metro, Train, Bus)
- Available on Android, iOS and as a WebApp

Developed by Israeli start-up Tranzmate since 2011, this app became the number one worldwide reference platform for mobility and transit, winning multiple awards and being widely funded. With the most data collected and claiming an offer of services in more than 1500 cities, this product, although missing some features, is widely used all around the globe.

Transit

Important features:

- Specialized in commutes
- Routing System and Transit/Delay System
- Multi-modal transport (Foot, Metro, Train, Bus, Bike)
- Available on Android and iOS



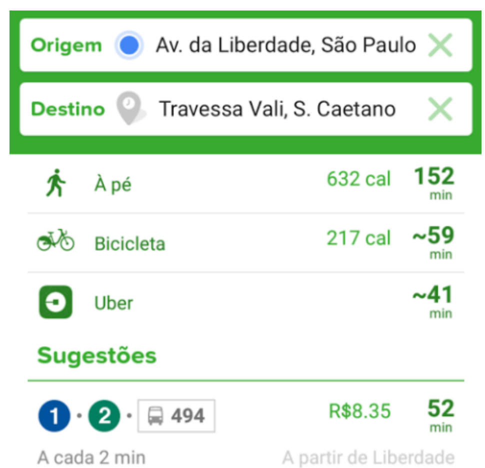
Competitor - Transit App

With a total of more than 125 cities supported, from North America, Europe and Oceania, Transit aims to provide a more suitable approach for the daily and small commute. Their goal is to make car commutes obsolete in high-density cities.

CityMapper

Important features:

- Specialized in commutes
- Routing System and Transit/Delay System
- Multi-modal transport (Foot, Metro, Train, Bus, Bike), and Uber
- Available on Android and iOS



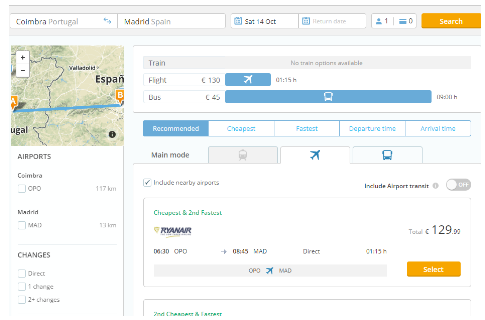
Competitor - CityMapper

With similar functionalities to Transit, the goal of this project is to revolutionize the commuters trip planning. Highly acclaimed, especially in some zones, like Sâo Paulo, in Brasil, this platform is capable of integration with external dependencies, like Uber, as well as support for live location sharing.

Goeuro

Important features:

- Major transport companies in Europe
- Routing System
- Multi-modal transport (Metro, Train, Bus, Plane)
- Available as website



Competitor - GoEuro

This platform was created with the goal to unify all transport systems in Europe, and with already with 12 countries available, is expanding into a platform suitable both really door-to-door mobility, as Uber is used to provide the ”last mile” service.

Transport Direct

Important features:

- Available in England, Scotland and Wales
- Routing System
- Multi-modal transport (Metro, Train, Bus, Plane)
- Available as website

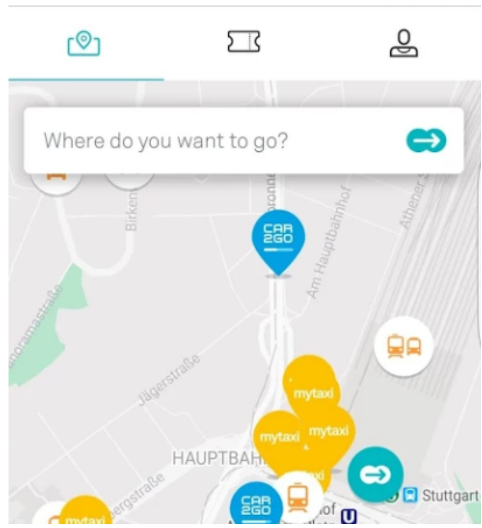
Competitor - TransportDirect

Transport Direct was launched in 2004, by the United Kingdom government, as the first journey planner to cover England, Scotland and Wales, proving useful the concept, only to be closed in 2014, establishing not only that a trip planner can be effective but also that the availability of transport information data can be beneficial for the transport companies.

Moovel

Important features:

- Available in Germany and USA
- Routing and Payment System
- Multi-modal transport (Metro, Train, Bus, Plane), Uber, Taxi and Car Rental
- Available as an application for Android and iOS



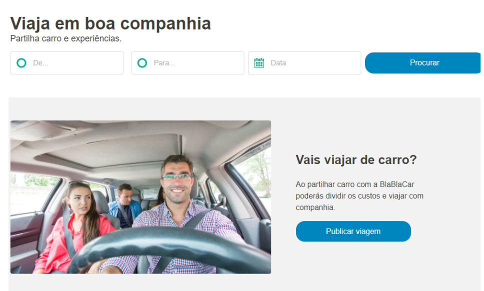
Competitor - Moovel

Moovel's concept is simple: one should never spend too much time on small trips nor think how to make them. With that in mind, this platform implements a routing system using public transport, but also external entities like car rental, taxis and Uber. Built for the smaller commutes, is expanding rapidly since 2012, as its already available in some USA cities.

Blablacar

Important features:

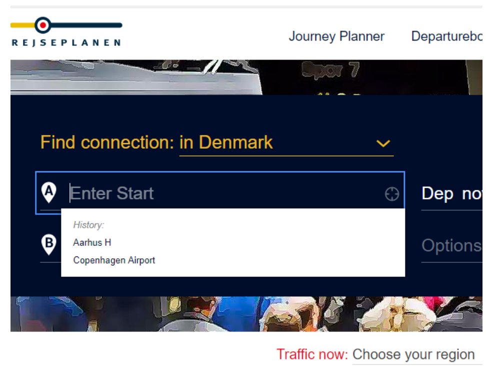
- Available Worldwide
- Car Sharing
- Available as a website



Competitor - Blablacar

This platform aims to ease the process of finding someone to split the expenses into the periodic or punctual trip, traveling in a personal car. Although its concept may be out of sync with the rest of the project, is important to keep in mind that this variant of transport, *carsharing*, is gathering interest, especially in the younger generation.

Rejseplanen



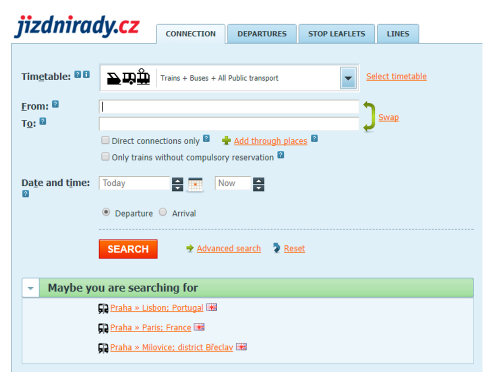
Competitor - Rejseplanen

Important features:

- Available in Denmark and international connections
- Routing and Ticketing System
- Multi-modal transport (Foot, Bike, Metro, Train, Bus, Plane)
- Available as a website

This platform focus on the Denmark transport system, allowing its users to find different routes, check its prices and buy the corresponding tickets, allowing routes in the country and connection with adjacent countries.

IDOS



Competitor - IDOS

Important features:

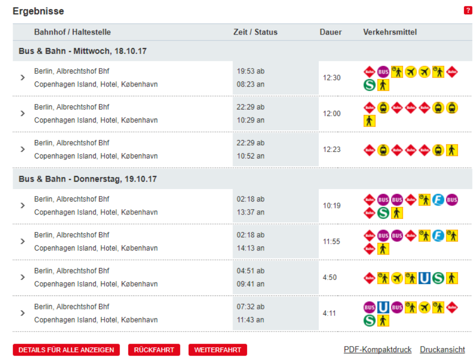
- Available in Czech Republic
- Routing and Ticketing System
- Multi-modal transport (Metro, Train, Bus)
- Available as a website

This platform acts as a multi-modal journey planner in the Czech Republic, allowing the user to find different routes using Metro, Train or Bus, allowing them to evaluate the prices and buy the tickets necessary.

Eu-spirit

Important features:

- Available in 6 european countries
- Routing and realtime information system
- Multi-modal transport (Metro, Train, Bus, Foot, Boat)
- Available as a website



Competitor - Eu-spirit

This information service, with the gathered and updated data from the countries Denmark, France, Germany, Luxembourg, Poland, and Sweden, allows the platform user to plan the route in a grand part for northern Europe. The available transport modes are on foot, plane, bus, train and boat, providing real-time transit information, and this non-profit web-based product can handle national as well as international trip planning.

Annex B - Competitors Description

	Region	Transit Alerts	Routing System	Payment	Features: MultiModal Transport: Foot	Metro	Train	Bus	Plane	Bike	Private Car	Car Sharing	Taxi/Uber	Rental Car	Bike Rental	How they achieve the objective
FluidHub	Worldwide	X	X	X	x	x	x	X	x	x	x	X	x	x	x	Custom Software
Google Transit	Worldwide	X	X		X	X	X	X								Routing system using GTFS
Whim	Helsinki			X		X	X	X						X		Adaptative solution
Waynaut (EA)	Wordwide		X	X	X	X	X	X								ND
Moovit	Urban Centers	X	X		X	X	X	X								ND
Transit	Most Close Region		X		X	X	X	X								ND
CityMapper	Urban Centers		X		X	X	X	X								ND
Goeuro	Europe		X				X	X	X							Flight, Train and Bus Information
Transport Direct	Britain		X			X	X	X								Used the agencies data
Moovel	Multiple countries		X	X	X	X	X	X					X	X		ND
Blablacar	Worldwide											X				User input
Rejseplanen	Denmark		X	X		X	X	X								Data from agencies
IDOS	Europe		X			X	X	X								ND
eu-spirit	Northern Europe		X	X		X	X	X	X							Iniciative to promote aggregation of data

Table 11.1: Comparison of the available MaaS products/platforms.

Annex C - User Stories

User Stories Structure

US#

- **Title:** Used to rapidly give an idea on what should be implemented.
- **Description:** Clear specification of the initial conditions and the outcome expected.
- **Success Criteria:** States the multiple response scenarios, along with the output presented to the user.
- **Dependencies:** The dependencies of a user story are the identifiers (US#) of those that should be implemented before.
- **Priority:** Identified as ’Must Have’, ’Should Have’ or ’Nice to Have’, the user story prioritization allow for better decision making during the development.

Features described with user stories

1 - Account

US1.1

- **Title:** Register
- **Description:** As an unregistered User, I should be able to make a request to register in the system, with the obligatory fields ’username’, ’email’ and ’password’.
- **Success Criteria:**
  - In case of success, the user receives a message indicating success;
  - If some of the obligatory camps is missing, the user receives an error message a message with that indication;
  - If the email submitted matches an account’s email in the system, the user receives a message with that indication;
  - In case of another error occurring, the user receives a message with that indication.
- **Dependencies:** None
- **Priority:** Must Have

US1.2

- **Title:** Login
- **Description:** As an unauthenticated User, I should be able to make a request to login in the system, with the obligatory fields ’email’ and ’password’.
- **Success Criteria:**
  - In case of success, the user receives a message indicating success;

  - If some of the obligatory camps is missing, the user receives an error message with that indication;
  - If the email submitted doesn't match an account's email in the system, the user receives a message with that indication;
  - If the email submitted matches an account's email in the system, but the password submitted doesn't match the account in the system, the user receives a message with that indication;
  - In case of another error occurring, the user receives a message with that indication.
- **Dependencies:** US1
- **Priority:** Must Have

US1.3

- **Title:** Edit Preferences
- **Description:** As an authenticated User, I should be able to make a request to edit my account preferences, with the indexes presented in Section 4.1.1.
- **Success Criteria:**
  - In case of success, the user receives a message indicating success;
  - In case of error occurring, the user receives a message with that indication.
- **Dependencies:** US2
- **Priority:** Must Have

US1.4

- **Title:** Delete Account
- **Description:** As an authenticated User, I should be able to make a request to delete my account from the system.
- **Success Criteria:**
  - In case of success, the user receives a message indicating success and all the user data is wiped from the system;
  - In case of another error occurring, the user receives a message with that indication.
- **Dependencies:** US2
- **Priority:** Nice to Have

2 - Trip Usage

US2.1

- **Title:** Trip by private car
- **Description:** As an authenticated User, I should be able to make a request for trip options using my private car, if my preference indexes allow it, with the obligatory fields 'start position' and 'end position'. Additionally, the user can specify where its car is parked, so it can be considered during processing.
- **Success Criteria:**
  - In case of success, the user receives the multiple options of trips he can choose, processed and sorted by his indexes;
  - If any of the obligatory fields are missing or incorrect, the user receives a message with that indication.
  - In case of another error occurring, the user receives a message with that indication.
- **Dependencies:** US2
- **Priority:** Must Have

US2.2

- **Title:** Trip by private car with parking location
- **Description:** As an authenticated User, I should be able to make a request for trip options using my private car, if my preference indexes allow it, with the obligatory fields 'start position' and 'end position'. In the end of the option, if available, the route should include a parking location to allow the user to leave its car and continue on foot. Additionally, the user can specify where its car is parked, so it can be considered during processing.
- **Success Criteria:**
  - In case of success, the user receives the multiple options of trips he can choose, processed and sorted by his indexes;
  - If any of the obligatory fields are missing or incorrect, the user receives a message with that indication.
  - In case of another error occurring, the user receives a message with that indication.
- **Dependencies:** US2
- **Priority:** Must Have

US2.3

- **Title:** Trip by private bike
- **Description:** As an authenticated User, I should be able to make a request for trip options using my private bike, if my preference indexes allow it, with the obligatory fields 'start position' and 'end position'.
- **Success Criteria:**

  - In case of success, the user receives the multiple options of trips he can choose, processed and sorted by his indexes;
  - If any of the obligatory fields are missing or incorrect, the user receives a message with that indication.
  - In case of another error occurring, the user receives a message with that indication.
- **Dependencies:** US2
- **Priority:** Must Have

US2.4

- **Title:** Trip by public transport
- **Description:** As an authenticated User, I should be able to make a request for trip options using private transportation, if my preference indexes allow it, with the obligatory fields 'start position' and 'end position'.
- **Success Criteria:**
  - In case of success, the user receives the multiple options of trips he can choose, processed and sorted by his indexes;
  - If any of the obligatory fields are missing or incorrect, the user receives a message with that indication.
  - In case of another error occurring, the user receives a message with that indication.
- **Dependencies:** US2
- **Priority:** Must Have

US2.5

- **Title:** Trip by foot
- **Description:** As an authenticated User, I should be able to make a request for trip options by foot, if my preference indexes allow it, with the obligatory fields 'start position' and 'end position'.
- **Success Criteria:**
  - In case of success, the user receives the multiple options of trips he can choose, processed and sorted by his indexes;
  - If any of the obligatory fields are missing or incorrect, the user receives a message with that indication.
  - In case of another error occurring, the user receives a message with that indication.
- **Dependencies:** US2
- **Priority:** Must Have

US2.6

- **Title:** Trip by bikesharing

- **Description:** As an authenticated User, I should be able to make a request for trip options using bikesharing, if my preference indexes allow it, with the obligatory fields 'start position' and 'end position'. The options should include the first point of bikesharing, where the user should walk to pick the bike, and the second point of bikesharing, allowing the user to leave the bike and continue on foot.
- **Success Criteria:**
  - In case of success, the user receives the multiple options of trips he can choose, processed and sorted by his indexes;
  - If any of the obligatory fields are missing or incorrect, the user receives a message with that indication.
  - In case of another error occurring, the user receives a message with that indication.
- **Dependencies:** US2
- **Priority:** Must Have

US2.7

- **Title:** Trip using Park&Go
- **Description:** As an authenticated User, I should be able to make a request for trip options using my park&go, if my preference indexes allow it, with the obligatory fields 'start position' and 'end position'. The response route options should include the trip to a parking location in the vicinity of the city, and the remaining trip on public transport, foot, or bikesharing. Additionally, the user can specify where its car is parked, so it can be considered during processing.
- **Success Criteria:**
  - In case of success, the user receives the multiple options of trips he can choose, processed and sorted by his indexes;
  - If any of the obligatory fields are missing or incorrect, the user receives a message with that indication.
  - In case of another error occurring, the user receives a message with that indication.
- **Dependencies:** US2
- **Priority:** Nice to Have

US2.8

- **Title:** Trip using Go&Drive
- **Description:** As an authenticated User, I should be able to make a request for trip options using my go&drive, if my preference indexes allow it, with the obligatory fields 'start position' and 'end position'. The response route options should include the public transport, foot, or bikesharing trip to its car location, and then the driving trip to its destination.
- **Success Criteria:**
  - In case of success, the user receives the multiple options of trips he can choose, processed and sorted by his indexes;
  - If any of the obligatory fields are missing or incorrect, the user receives a message with that indication.

  - In case of another error occurring, the user receives a message with that indication.
- **Dependencies:** US2
- **Priority:** Nice to Have

US2.9

- **Title:** Choose Trip
- **Description:** As an authenticated User, after making any request for trip options, I should be allowed to mark a given route as chosen, so it can be considered in the history processing in the upcoming requests I make.
- **Success Criteria:**
  - In case of success, the user receives confirmation.
  - In case of an error occurring, the user receives a message with that indication.
- **Dependencies:** US2
- **Priority:** Nice to Have

US2.10

- **Title:** See chosen trips
- **Description:** As an authenticated User, I should be able to see a list of trip options I choose.
- **Success Criteria:**
  - In case of success, the user receives the desired information.
  - In case of an error occurring, the user receives a message with that indication.
- **Dependencies:** US2
- **Priority:** Nice to Have

Disclaimer: All the section Trip Usage user stories imply that coming with along the necessary information to represent the trip option(s), it should also be included polyline encoded geometries, allowing the route option to be displayed on most map applications or tools.

3 - Additional Features - Trip tracing

US3.1

- **Title:** Trip Tracing Registering
- **Description:** As an authenticated User, I should be able to make a to start a trip tracing.
- **Success Criteria:**
  - In case of success, the user receives an identification, so it can send position reports.
  - In case of another error occurring, the user receives a message with that indication.
- **Dependencies:** US2
- **Priority:** Could Have

US3.2

- **Title:** Trip Tracing Posting
- **Description:** As an authenticated User, I should be able to post a new trace report, with the obligatory fields ”location”, ”timestamp” and ”mode”, where the location is a GPS coordinates tuple, timestamp refers to a time reference and mode indicates one of the three options: ”walk”, ”car”, ”bike”.
- **Success Criteria:**
  - In case of success, the user receives a success confirmation.
  - In case of another error occurring, the user receives a message with that indication.
- **Dependencies:** US2
- **Priority:** Could Have

US3.3

- **Title:** Trip Tracing Closure
- **Description:** As an authenticated User, I should be able to end the oncoming trip tracing, and receiving the traced trip, as well as a polyline encoded geometry, allowing it to be displayed on most map applications of tools.
- **Success Criteria:**
  - In case of success, the user receives a success confirmation and the information requested.
  - In case of another error occurring, the user receives a message with that indication.
- **Dependencies:** US2
- **Priority:** Could Have

US3.4

- **Title:** See Trip Traces
- **Description:** As an authenticated User, I should be able to consult my trip traces,
- **Success Criteria:**

  - In case of success, the user receives a success confirmation and the information requested..
  - In case of another error occurring, the user receives a message with that indication.
- **Dependencies:** US2
- **Priority:** Could Have

4 - Additional Features - Car Sharing

US4.1

- **Title:** Car trip information
- **Description:** As an authenticated User, I should be able to request the system the distance and time necessary to complete a car trip, with the obligatory fields ”origin” and ”destination”.
- **Success Criteria:**
  - In case of success, the user receives the requested information.
  - In case of another error occurring, the user receives a message with that indication.
- **Dependencies:** US2
- **Priority:** Nice to Have

US4.2

- **Title:** Car sharing trip registering
- **Description:** As an authenticated User, I should be able to insert a new carsharing trip, with the obligatory fields ”origin” and ”destination”.
- **Success Criteria:**
  - In case of success, the user receives confirmation, as well as an encoded polyline geometry, that can be used to represent the trip on most map apps and tools.
  - In case of another error occurring, the user receives a message with that indication.
- **Dependencies:** US2
- **Priority:** Nice to Have

US4.3

- **Title:** Car sharing trip search
- **Description:** As an authenticated User, I should be able to search for an existing carsharing trip offer, with the obligatory fields ”origin”, ”destination”, and ”range”. The matched trips should include trips whose origin and destination match the requested trip origin and destination or are whose origin and destination are included in the set of passing points in the requested trip (considering range as the maximum detour).
- **Success Criteria:**
  - In case of success, the user receives confirmation, as well as an encoded polyline geometry, that can be used to represent the trip on most map apps and tools.

  - In case of another error occurring, the user receives a message with that indication.
- **Dependencies:** US2
- **Priority:** Nice to Have

Annex D - Scenarios

Scenery 1

Quality Attribute:	**Interoperability**
Source of stimulus:	User Application
Stimulus:	The User makes a request for a trip suggestion, that needs to use external information or components.
Environment Conditions:	Normal Operation
Artefacts:	System, External Systems
Response:	The system communicates successfully with external systems and computes the necessary routes.
Response Measurement:	The User receives the requested trip suggestions.

Table 11.2: Interoperability Scenery

Scenery 2

Quality Attribute:	**Performance**
Source of stimulus:	User Application
Stimulus:	The User makes any disclosed request as can be consulted in the User Stories Specification.
Environment Conditions:	Normal Operation
Artefacts:	System
Response:	The system responds as expected, in an expected time, in normal operational conditions.
Response Measurement:	The system responds in less than: • 0.5 seconds for account related requests. • 0.5 seconds for tracing and carsharing related requests. • 2 seconds for the single transport mode requests. • 4 seconds for the multiple transport mode requests. • 7 seconds for the multiple transport mode requests, with multiple options.

Table 11.3: Performance Scenery

Scenery 3

Quality Attribute:	**Scalability**
Source of stimulus:	Users
Stimulus:	The system receives an increasing number of requests.
Environment Conditions:	Normal Operation (but heavy load)
Artefacts:	System
Response:	The system must implement scalability good practices and support the increasing user base, usage and information storage. With that in mind, all components of the project must be implemented with this problem in mind, although it shall not be a may concern to have them deployed at the internship's project end.
Response Measurement:	The system scales correctly and the user's experience is not affected.

Table 11.4: Scalability Scenery

Scenery 4

Quality Attribute:	**Security (privacy)**
Source of stimulus:	Authenticated User
Stimulus:	The User authenticates itself in the system or performs a request.
Environment Conditions:	Normal Operation
Artefacts:	System
Response:	The User gets authenticated or receives the answer to his request, whilst the communication is secure. The user must perform the requests with an authentication token.
Response Measurement:	No other User gets access to the information, exploiting communication methods.

Table 11.5: Security Scenery (1)

Scenery 5

Quality Attribute:	Security (data protecting in system)
Source of stimulus:	Authenticated User
Stimulus:	The User has information relative to him stored in the system.
Environment Conditions:	Normal Operation
Artefacts:	System
Response:	The User gets authenticated or receives the answer to his request, and the information stored in the server is secure.
Response Measurement:	No other User gets access to the information, by accessing the system.

Table 11.6: Security Scenery (2)

Scenery 6

Quality Attribute:	Security (authentication)
Source of stimulus:	Unauthenticated User
Stimulus:	The User tried ties to authenticate itself in the system with wrong credentials.
Environment Conditions:	Normal Operation
Artefacts:	System
Response:	The User gets denied access to the system.
Response Measurement:	The User receives a negative response to the authentication request.

Table 11.7: Security Scenery (3)

Scenery 7

Quality Attribute:	**Testability**
Source of stimulus:	System Tester
Stimulus:	Test batch to assert quality and speed of new components added.
Environment Conditions:	Testing Branch
Artefacts:	Product under Testing
Response:	The system is conceived in a way that makes it easy for the tester to develop a test suite, run it and capture the results.
Response Measurement:	The system responds and continuously outputs the state of operation, it's internal state and statistics.

Table 11.8: Testability Scenery

The remaining quality attributes considered for the system are better presented with a description of the measure(s) to increase the quality of the system, as well as its measurements:

- **Maintainability** - The developed product should be implemented taking into attention its future use, as it should be suitable for revision, adaptation, expansion, and should be prepared in case of need to implement new requirements. With that in mind, the implementation should follow not only the rules imposed by general good practices but also the rules and recommendations imposed by the hosting company.
- **Reliability** - The system should be implemented with the most reliability increasing practices possible, as such: exception and error handling and recovery, redundancy tactics and situation logging and, in an eventual problematic situation, alert the system for the deployment team.
- **Availability** - Measuring the reliability of the system, the product uptime should be 99.9%, meaning its downtime per year should be no bigger than 8.76 hours.