

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

FEUPooling: Carpooling Platform

Bruno Ricardo Dias Gonçalves



Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Ademar Manuel Teixeira de Aguiar

July 19th, 2017

FEUPooling: Carpooling Platform

Bruno Ricardo Dias Gonçalves

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

President: Prof. Hugo José Sereno Lopes Ferreira

External Examiner: Prof. Jorge Manuel de Azevedo Pereira Simões

Supervisor: Prof. Ademar Manuel Teixeira de Aguiar

July 19th, 2017

Abstract

Nowadays, automobiles are one of the most popular means of transportation and the majority are used by only one person. This leads to a lot of traffic in the urban areas, which causes, for example, an increase in atmospheric and sound pollution. Besides that, the price of fuel, tolls and parking has been gradually increasing. In many cases, public transportation is not an option because of the lack of comfort, freedom and flexibility of schedules. Therefore, the idea of carpooling emerged with the goal of minimizing these problems. Carpooling is the sharing of private vehicle journeys so that more than just the driver can travel in the same car.

However, not everyone is receptive to this idea, and the main obstacles stated are the lack of security and trust in sharing a ride with strangers. Nevertheless, in the specific context of an organization like FEUP, the community of users is known and trusted, so this problem does not apply. On this community, several behaviors, routines, travels and schedules can be anticipated. This can help simplify several aspects of carpooling, making the adoption easier. In addition, this allows for an integration of the platform with the existing organizational information system of FEUP which in turn increases the security and trustability of the system.

With this kind of ride sharing, money can be saved by splitting costs of fuel, parking and tolls by the passengers involved. Apart from that, it is a more ecological and sustainable way of commuting because it reduces the traffic density, consequently reducing the emission of greenhouse gases. Furthermore, the surrounding area of FEUP lacks free parking, so the introduction of such a service would be beneficial.

The project proposed by the *Comissariado para a Sustentabilidade da FEUP* consisted on developing mobile applications for the described community. This system, which is the core of the project, aimed to be as easy to use as possible in order to have the maximum possible adoption. The users – in this case, the students – are able to post ride sharing offers as drivers or search for them as passengers. Additionally, a points system was added. This system can be considered a simple gamification mechanism, in order to motivate and reward users who use the service. A back-end system then verifies if a trip was valid, by calculating the similarity of the passenger's routes before rewarding the trip participants with points. These points can then be exchanged later on by rewards to be defined by the faculty.

Besides that, a web application for administrative purposes was also developed so that the system can be controlled and managed. The administrator(s) can view information about trips and users in order to ensure that there are no irregular behaviors. The platform also allows an administrator to exchange a user's points for rewards.

The goal is for all interested students to be able to access this application in order to share their commute in an easy way without being concerned about security.

Resumo

Atualmente, os automóveis são um dos mais populares meios para transporte, sendo que a maioria apenas é utilizado por uma pessoa. Isto faz com que em zonas mais urbanas se crie muito trânsito, o que leva, por exemplo, ao aumento da poluição atmosférica e sonora. Para além disso, o preço de combustíveis, portagens e estacionamento tem vindo a aumentar gradualmente. Em muitos casos, os transportes públicos não são opção devido à falta de conforto, liberdade e flexibilidade de horários. Assim sendo, a ideia de *carpooling* surgiu com objetivo de minimizar estes problemas. *Carpooling* é a partilha de viagens em veículos privados, onde se pretende que viaje mais que uma pessoa por viatura, para além do condutor.

Contudo, nem todos são muito recetivos a esta ideia, sendo que é apontado como um dos principais entraves a falta de segurança e confiança em partilhar transporte com desconhecidos. Porém, no contexto específico de uma organização como a FEUP, já existe uma comunidade de utilizadores conhecida e de confiança, o público alvo deste projeto, sendo que este problema não se aplica. Nesta comunidade podem antecipar-se vários comportamentos e rotinas, viagens e horários. Isto pode ajudar a simplificar vários aspetos do *Carpooling*, tornando assim a adoção mais fácil. Adicionalmente, isto permite uma integração da plataforma com os existentes serviços de informação organizacionais da FEUP, o que aumenta o factor de segurança e confiança no sistema.

Com este tipo de partilha de transporte, há a possibilidade de poupar dinheiro ao dividir custos de combustível, estacionamento e portagens pelos passageiros envolvidos. Para além disso, é uma forma mais ecológica e sustentável de viajar pois reduz a densidade do trânsito, diminuindo assim a emissão dos gases com efeito de estufa. Mais ainda, a zona envolvente à FEUP carece de estacionamento gratuito, daí que a introdução de um serviço deste tipo seja benéfica.

O projeto proposto pelo *Comissariado para a Sustentabilidade da FEUP* consistiu principalmente em desenvolver aplicações móveis para a comunidade descrita. O sistema, que é o cerne deste projeto, tinha como objetivo ser o mais simples de usar possível de forma a ter o máximo de adesão alcançável. Os utilizadores – neste caso, os estudantes – podem publicar anúncios de ofertas de partilha de transporte como condutores ou procurar por viagens como passageiros. Adicionalmente, um sistema de pontos foi adicionado. Este sistema pode ser considerado um mecanismo simples de *gamification*, de forma a motivar e recompensar utilizadores que usem o serviço. Um sistema *back-end* verifica se uma viagem foi considerada válida, calculando a similaridade das rotas dos passageiros antes de recompensar os participantes na viagem com pontos. Estes pontos podem posteriormente ser trocados por recompensas a serem definidas pela faculdade.

Para além disso, uma aplicação *web* para propósitos administrativos foi desenvolvida de modo a que o sistema possa ser controlado e gerido. O(s) administrador(es) pode(m) ver informação relacionada com viagens e utilizadores com o objetivo de garantir que não existem comportamentos irregulares. A plataforma também permite que um administrador troque os pontos dos utilizadores por recompensas.

O objetivo final é que todos os estudantes interessados acedam a esta aplicação de forma a poder partilhar transporte de forma facilitada e sem preocupações acerca da segurança.

Acknowledgements

Regarding of the development of this master's thesis, I would like to express my gratitude to all those who, directly or indirectly, helped in its development. Therefore, this a message of thanking and recognition for those people.

First of all, it is impossible not be thankful to my parents, because none of this would be possible without their the support and encouragement along the way, in the good and bad moments. They always believed in me even when I doubted.

Then I would like to thank my supervisor Professor Ademar Aguiar for all the guidance during the development and for giving me the opportunity to work on an interesting and meaningful project that I could relate as a college carpooler myself.

I would also like thank my "second supervisor", Eduardo, that I gladly call a friend. He was and is indispensable and I could not think of a better person to share this experience with. I am very grateful for all the help, motivation and patience. "You've got a friend in me" for life!

Last, but clearly not the least, I would like to thank all my friends and family that took some of their time to try to keep me motivated and supported me all the ways they could. I can also never forget about departed loved ones, that live in my thoughts every step of my path, always hoping that they would be proud of me.

Bruno Gonçalves

*“Many of life’s failures are people who did not realize
how close they were to success when they gave up.”*

Thomas Edison

Contents

1	Introduction	1
1.1	Context	1
1.2	Motivation and Goals	3
1.3	Report Structure	3
2	Ride Sharing	5
2.1	Carsharing	6
2.2	Carpooling	6
2.2.1	Benefits	6
2.2.2	Inconvenients	7
2.2.3	Types of Carpooling	10
3	Mobility Platforms	13
3.1	Carpooling Platforms	13
3.1.1	Boleia.net	13
3.1.2	BlaBlacar	15
3.1.3	Galpshare	16
3.1.4	ToGoCarpool	17
3.1.5	Pool My Ride	18
3.1.6	WazeCarpool	18
3.1.7	RideAmigos	19
3.2	Other Platforms	19
3.2.1	Uber	19
3.2.2	Cabify	21
3.2.3	MEO Táxi	21
3.2.4	mytaxi	22
3.2.5	car2go	22
3.2.6	citydrive	23
3.3	Mobility Platform Features	23
4	FEUPooling: Carpooling Platform	25
4.1	Problem	25
4.2	Friendly Mobility Study	26
4.3	Solution	27
4.4	FEUPooling Backend	29
4.4.1	Federated authentication	32
4.4.2	Gamification System	33
4.4.3	Reputation System	34

CONTENTS

4.5	FEUPooling Web	36
4.5.1	Requirements	36
4.5.2	Features	38
4.5.3	Implementation	40
4.6	FEUPooling Mobile	41
4.6.1	Requirements	41
4.6.2	Features	46
4.6.3	Implementation	49
5	Validation	53
5.1	Usability	53
5.2	Performance and Energy	54
5.3	Trip Validation	56
6	Conclusions	59
6.1	Contributions	59
6.2	Future Work	60
A	<i>FEUPooling Server API</i>	63
A.1	Authentication	63
A.2	User Routes	63
A.2.1	Notification Routes	64
A.2.2	Rating Routes	64
A.3	Trip Routes	64
A.4	Ride Request Routes	66
A.5	Admin Routes	67
A.6	Models	69
B	Screenshots	71
B.1	<i>FEUPooling</i> Web	71
B.2	<i>FEUPooling</i> Mobile	80
C	<i>FEUPooling Server Database Schema</i>	95
	References	101

List of Figures

1.1	Growth projection of motor vehicles [SG08]	2
1.2	Comparative growth of the number of registered vehicles, miles travelled and population of the United States of America [SG08]	2
2.1	Anti-Nazi campaign poster used in the United States of America to encourage transport sharing during WWII [lea17]	8
3.1	<i>Boleia.net</i> driver and passenger advertisements [Lda16a]	14
3.2	<i>BlaBlacar</i> trip details page [Bla17]	15
3.3	<i>Galpshare</i> trip details page [Gal17]	16
3.4	<i>ToGoCarpool Android</i> application: group with trip ads (left) and trip details page (right) [Sto17i]	17
3.5	<i>WazeCarpool Android</i> application: ride found screen (left) and map showing an active ride (right) [Sto17k]	18
3.6	<i>RideAmigos</i> dashboard [Rid17]	19
3.7	<i>MEO Táxi Android</i> application: setting the vehicle preferences (left) and map showing the taxi on his way (right) [Sto17f]	21
3.8	<i>mytaxi Android</i> application: map showing available taxis (left) and rating screen at the end of each trip (right) [Sto17g]	22
3.9	<i>car2go Android</i> application: map showing available vehicles (left) and vehicle details (right) [Sto17c]	23
4.1	<i>Laravel MVC</i> [lar17]	29
4.2	Database Schema	31
4.3	Users <i>UML</i> for <i>FEUPooling Web</i>	36
4.4	<i>UML</i> Use Cases for <i>FEUPooling Web</i>	38
4.5	User page - <i>FEUPooling Web</i>	39
4.6	Trip page - <i>FEUPooling Web</i>	39
4.7	Users <i>UML</i> for <i>FEUPooling Mobile</i>	41
4.8	<i>UML</i> Use Cases for <i>FEUPooling Mobile</i>	46
4.9	Main screen map - <i>FEUPooling Mobile</i>	47
4.10	User profile - <i>FEUPooling Mobile</i>	48
4.11	On trip mode for the driver - <i>FEUPooling Mobile</i>	49
5.1	Test trip 1	55
5.2	Test trip 2	58
B.1	Login - <i>FEUPooling Web</i>	71
B.2	Users' list - <i>FEUPooling Web</i>	72

LIST OF FIGURES

B.3	Trips' list - <i>FEUPooling</i> Web	73
B.4	Trip page - <i>FEUPooling</i> Web	74
B.5	Trip page - <i>FEUPooling</i> Web	75
B.6	Settings page - <i>FEUPooling</i> Web	76
B.7	User page - <i>FEUPooling</i> Web	77
B.8	User page - <i>FEUPooling</i> Web	78
B.9	User page - <i>FEUPooling</i> Web	79
B.10	Splash screen - <i>FEUPooling</i> Mobile	80
B.11	Login page without internet connection - <i>FEUPooling</i> Mobile	80
B.12	Login page - <i>FEUPooling</i> Mobile	81
B.13	Main screen map - <i>FEUPooling</i> Mobile	81
B.14	Trip list - <i>FEUPooling</i> Mobile	82
B.15	Create trip part 1 - <i>FEUPooling</i> Mobile	82
B.16	Create trip part 2 - <i>FEUPooling</i> Mobile	83
B.17	Departure or arrival place choosing view - <i>FEUPooling</i> Mobile	83
B.18	User profile - <i>FEUPooling</i> Mobile	84
B.19	Edit user profile - <i>FEUPooling</i> Mobile	84
B.20	Trips list of a user as a driver - <i>FEUPooling</i> Mobile	85
B.21	Trips list of a user as a passenger - <i>FEUPooling</i> Mobile	85
B.22	Notifications - <i>FEUPooling</i> Mobile	86
B.23	Trip page - <i>FEUPooling</i> Mobile	86
B.24	Trip page with available seats - <i>FEUPooling</i> Mobile	87
B.25	Trip page with pending request made by the user - <i>FEUPooling</i> Mobile	87
B.26	Cancel request made by the user - <i>FEUPooling</i> Mobile	88
B.27	Trip page with pending request waiting for the driver's approval - <i>FEUPooling</i> Mobile	88
B.28	Approval options for the driver's approval - <i>FEUPooling</i> Mobile	89
B.29	Driver's removal option from a trip - <i>FEUPooling</i> Mobile	89
B.30	Passenger's removal option from a trip - <i>FEUPooling</i> Mobile	90
B.31	Edit trip page part 1 - <i>FEUPooling</i> Mobile	90
B.32	Edit trip page part 2 - <i>FEUPooling</i> Mobile	91
B.33	Trip page ready to start - <i>FEUPooling</i> Mobile	91
B.34	Trying to find user position before start trip - <i>FEUPooling</i> Mobile	92
B.35	On trip mode for the driver - <i>FEUPooling</i> Mobile	92
B.36	On trip mode for passengers - <i>FEUPooling</i> Mobile	93
B.37	Trying to stop a trip far from the arrival location - <i>FEUPooling</i> Mobile	93
B.38	Trip page that already ended - <i>FEUPooling</i> Mobile	94
B.39	Rating page - <i>FEUPooling</i> Mobile	94
C.1	Database Schema	100

List of Tables

4.1	<i>FEUPooling Web</i> Actors Description	36
4.2	<i>FEUPooling Web</i> User Stories for Unauthenticated User	36
4.3	<i>FEUPooling Web</i> User Stories for an Authenticated User	37
4.4	<i>FEUPooling Mobile</i> Actors Description	42
4.5	<i>FEUPooling Mobile</i> User Stories for Unauthenticated User	42
4.6	<i>FEUPooling Mobile</i> User Stories for Authenticated User	43
4.7	<i>FEUPooling Mobile</i> User Stories for Driver Authenticated User	44
4.8	<i>FEUPooling Mobile</i> User Stories for Passenger Authenticated User	45
5.1	Actions and number of taps required	54
5.2	Battery consumption comparison	55
5.3	Data consumption comparison	56
5.4	Validation Scenario 1	56
5.5	Validation Scenario 2	57

LIST OF TABLES

Abbreviations

API	Application programming interface
FEUP	Faculdade de Engenharia da Universidade do Porto
HOV	High-occupancy vehicle lane
IdP	Identity Provider
JSON	JavaScript Object Notation
MVC	Model-View-Controller
OPEC	Organization of the Petroleum Exporting Countries
REST	Representational State Transfer
SAML	Security Assertion Markup Language
SP	Service Provider
SSL	Secure Sockets Layer
SSO	Single Sign-On
TLS	Transport Layer Security
UML	Unified Modeling Language
WWII	World War II
XML	eXtensible Markup Language

Chapter 1

Introduction

1.1 Context

The number of vehicles on the roads has been gradually increasing. The number of vehicles per household has also increased, which, in turn, has increased the number of journeys made, and vehicles circulating on the roads [Cam02]. Mobility needs have grown as a result of economic, technological and social development [Rei11]. This means that, especially in urban centers, traffic congestion is increasing, which also makes people's journeys more time-consuming [BSB⁺10]. The longer the vehicles are stuck on the road due to traffic congestion, the more emissions of polluting gases occur. In addition to that, there is a decrease in accessibilities and a great loss of time.

The existing vehicles already emit a huge amount of greenhouse gases into the atmosphere and are one of the biggest contributors to the exhaust of non-renewable resources like fossil fuels. In the short term, this could lead to major economic and environmental issues [BSB⁺10].

It is estimated that the number of motor vehicles at a global level is growing at a rate of 3% per year, with the growth rate in Europe being less than 1% and in the United States of America between 1% and 2%. However, in China and India it is estimated that this growth may be 7% or 8% [SG08].

It is also expected that the number of vehicles worldwide reaches 2 billion by 2030. This increase will intensify the innumerable problems mentioned above, especially within the bigger cities [BSB⁺10].

These situations negatively affect human health and significantly degrade the quality of life of populations. It is believed that the main cause of aggression during driving comes from road congestion because it increases the people's frustration [LPS99]. On a business or education level, these situations can negatively affect productivity, both at individual and team

Introduction

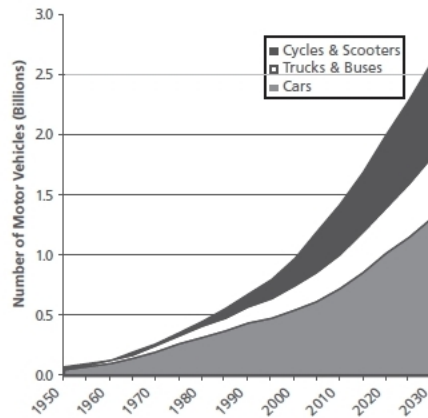


Figure 1.1: Growth projection of motor vehicles [SG08]

level [BSB⁺10]. Besides that, but not less important, it also harms the environment through air and noise pollution [BSB⁺10].

Regarding energy consumption, in 2008, about 85 million barrels of oil per day were consumed. It is estimated that this demand can reach 120 million barrels per day by 2030 [BSB⁺10].

“One-fourth of all the oil consumed by humans throughout history will be consumed during the next 10 years” (2008) [SG08]

Besides the problems caused at an individual level, this excessive demand entails difficulties of access to non-renewable energy resources and consequently leads to climate changes [BSB⁺10].

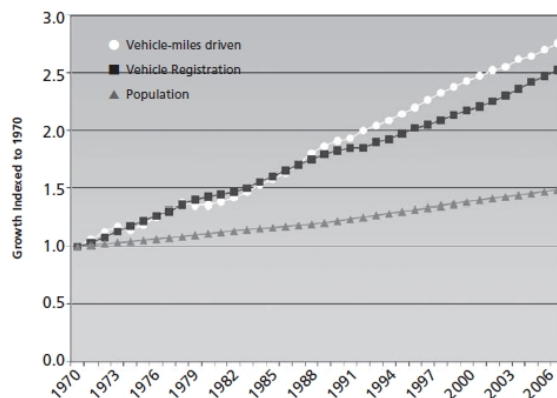


Figure 1.2: Comparative growth of the number of registered vehicles, miles travelled and population of the United States of America [SG08]

Regarding climate changes, the scientific community believes that greenhouse gas emissions need to be reduced by 50% to 80% by 2050 in order to avoid a catastrophic economic and

Introduction

environmental crisis. Automotive production companies have been trying to adapt to this new reality by investing in prevention measures, particularly in electric cars that are gradually becoming more common nowadays [BSB⁺10].

There are innumerable reasons why people should try to reduce big traffic accumulations. In addition to making people lose their time, money and increasing the risk of accidents, the main consequence is the high value of greenhouse gas emissions into the atmosphere [BB09].

That being said, these issues have drawn the attention of governments, companies and even individuals, which are trying to find alternative ways to make travel management more efficient [BSB⁺10]. The goal is to reduce the number of travels made by motor vehicles by encouraging the use of alternative transportation in individual daily commutes.

1.2 Motivation and Goals

The increase in transport flow and the consequent continuous traffic congestion in more urban areas has been, over the years, trying to be opposed with the construction of new infrastructures and by increasing the capacity of the road network. However, the solution of building new infrastructures or improve existing ones is no longer viable, so better ways of mobility should be chosen [Cam02].

Every day, thousands of students, employees and teachers travel to *Faculdade de Engenharia da Universidade do Porto*. This reality means that mobility in this urban area is somewhat lost due to the enormous congestion, which limits some freedom associated with travelling in private vehicles. Regarding this transportation problem, due to the urban expansion and the increase of the number of people with cars, it is mandatory to adopt measures to optimize the use of private vehicles. Additionally, the area in question also lacks free parking, which is very inconvenient and expensive.

It was intended, then, to implement a platform to be used by the *FEUP* community to ease the sharing of transportation. In this platform, all interested members of the community can use this form of mobility, named carpooling, in order to share trips with people that have similar travel routes. The objective is to increase the occupancy rate of vehicles and to decrease the number of vehicles in circulation, resulting in a minimization of all the problems that this entails.

1.3 Report Structure

Besides the introduction, this report contains five more chapters.

In the chapter 2, ways of sharing transport are described, mainly focusing on carpooling, which aims to minimize some of the problems identified on this chapter. Regarding carpooling, a brief historical context is given, along with some benefits, inconveniences and factors that can act as a barrier to the adoption of this concept.

Introduction

On the other hand, in chapter 3, the state of the art is described and works related to the subject in question are presented. Some mobility platforms are described and analyzed and at the end some relevant functionalities are summarized.

Regarding the chapter 4, a geographic context of the faculty is given, a study on friendly mobility that allows to get to know better the community in question is mentioned, and the solution implemented to the problems identified is presented and described.

The following chapter, 5, focuses on some tests done with the application.

Finally, in chapter 6 some conclusions of what has been done are presented and the work that can be developed in the future is put in perspective.

Chapter 2

Ride Sharing

With the increase in traffic, there is a great chance for an energy crisis and disastrous climate changes to become reality in the near future. It is then necessary to focus on alternative methods of transportation [HTZ11].

The increasing costs associated with travelling, such as fuel, road taxes, parking, tolls and car expenses are other factors that are driving users to look for alternatives [BSB⁺10].

People want to travel quickly and cheaply, but at the same time with the greatest comfort possible. There is then the dilemma of choosing between private interests or the general population interest [SG08].

The solution will hardly be to eliminate all personal vehicles. They offer freedom, flexibility and comfort to their owners and households, as motorized vehicles have revolutionized the modern life [SG08]. There are many eco-friendlier solutions, like traveling by bicycle. However, factors such as the lack of comfort and protection for adverse weather conditions make the option not very viable for continuous use.

Public transportation is not always the best option. The costs of monthly passes are sometimes too high, the trips take too much time because there is sometimes the need to change lines and/or the frequency is not what is wanted/needed. In addition, there are frequent system malfunctions, which is problematic for people with commitments.

Automobiles dominate the transport system and some innovative alternatives for transport sharing have emerged, on which individual vehicles are supposed to be used in a collective way, such as [SG08]:

- Carsharing – Concept that arose in Switzerland;
- Carpooling – Concept that arose in the United States of America.

2.1 Carsharing

Carsharing is an alternative transportation method where it is possible to rent a vehicle for a certain period of time, usually short, for a particular fee, normally with fuel and parking included. It is a service designed for more casual users. There are many ways this system can work. For instance, someone who owns a vehicle that will not be used for a period of time, can rent it to another person for that same period. For example, someone who is at work from 9 a.m. to 6 p.m. could rent their car during this time period. This concept emerged in Europe between 1940 and 1980, but only became more popular around 1990, and continues to expand nowadays [SC07].

2.2 Carpooling

Carpooling (also associated with ride-sharing or lift-sharing), which is the main focus of this study, is a different concept from the one summarized previously, and can be generalized as the sharing of a single vehicle by several users [HTZ11]. However, the concept of transport sharing is common to both. This concept appeared as an attempt to solve some of the problems mentioned before, even though it is not a new way of transportation.

“An arrangement between people to make a regular journey in a single vehicle, typically with each person taking turns to drive the others.” [Dic]

Carpooling is not only defined by owning a private vehicle but also having the desire to share it. The flexibility of schedules, costs, security and waiting times are key factors for people willing to join this concept. There is a connection between beliefs, feelings and behaviours, but above all, liking or being willing to try out carpooling is the key factor [HTZ11].

Certain trips sometimes involve costs that are too high, because of charges related to fuel, parking, tolls, among others. Thus, the first major advantage is saving money when travelling, since the costs mentioned can be divided by all passengers. In addition, it is environmentally friendly because it helps protect the environment, since all passengers on that trip would travel separately if they were not carpooling. It is also worth noting that, on more regular journeys, new and interesting people can be met. Sharing the vehicles can make traveling less monotonous and much more social.

2.2.1 Benefits

Some of the main advantages of this more sustainable mobility method can be summarized in the following list:

- Cost – There is a great reduction of expenses with fuel, parking, tolls and vehicle maintenance;
- Comfort – The comfort of traveling in a private vehicle is much higher than that of public transportation;

Ride Sharing

- Traffic — There is significantly less congestion of traffic on the roads and consequently less accidents and less stress. In addition, it also facilitates the movement of priority vehicles, such as ambulances;
- Environment – If there is less traffic on the roads, less atmospheric and noise pollution is likely to exist;
- Meeting new people – There can be sharing of knowledge, creation of friendship ties and consequently easier integration on some environments.

Regarding organizations, there are more specific benefits, such as [Cam02]:

- Parking — Reduce the need of parking spots;
- Traffic decongestion — There is less traffic in the organizations' surrounding areas;
- Pollution — Fewer vehicles implies less pollution, which can reduce the collaborators' health problems such as asthma, bronchitis, among others.

2.2.2 Inconvenients

Despite the numerous advantages, there are some aspects that may not be as advantageous, for example:

- Dependence — There may be less freedom/flexibility compared to traveling alone;
- Personality traits – Participants cannot have conflicting personalities, or else the carpooling may not work as good as expected;
- Commitment – In order for this method of transport sharing to work well, participants must comply with the agreed situations, such as schedules and cost division;
- Routes – The routes must be similar so people can share transport;
- Privacy – People may lose some privacy when sharing transport, especially if it is with strangers.

This way of travel aroused when the first motor vehicles appeared, in order to combat the temporary lack of fuel and materials. Historically, the interest in carpooling increases exponentially during some economic or energetic crisis, when there's a greater need to use the available resources rationally. In the case of the United States of America, this concept was particularly relevant during World War II, when a number of transport sharing awareness campaigns were conducted in order to combat fuel shortages. When the war ended, this concept was somehow forgotten and only regained special relevance during a new oil crisis in the 1970s [HTZ11].



Figure 2.1: Anti-Nazi campaign poster used in the United States of America to encourage transport sharing during WWII [lea17]

In 1973, Arab members of the OPEC (Organization of the Petroleum Exporting Countries) embargoed the export of oil to the countries that were allies with Israel in the war with Syria and Egypt. The need for a more sustainable mobility method became, once again, central, and several measures have been taken to encourage it. In the following decade, there was a period of economic recovery and expansion, which made this way of transportation decline once again [Rei11].

Nowadays, carpooling is a practice used not only with the fuel price as the reason but also because it is a more efficient way of transportation. Besides that, it is a viable solution because it is not necessary for people to abandon all of their private cars [HTZ11]. When sharing a vehicle, the occupancy rate is increased and thus the number of vehicles on circulation is decreased, while maintaining the advantages that private vehicles offer [Cam02].

In Portugal, this still is a relatively recent concept and the ride sharing that occurs is usually more punctual and not through transport sharing networks.

However, other countries even have high-occupancy vehicle lanes (HOV) for vehicles with a minimum of two occupants. Cities like Beijing, Hong Kong and California, have these types of lanes dedicated to all vehicles, public or private, as long as they have a high occupancy rate. Consequently, users have a faster, safer and less polluting way of traveling. In addition, another measure to reduce congestion, and at the same time encourage the sharing of transport, is a discount on the value of tolls for vehicles that are being shared [Cam02]. It is also important to notice that

Ride Sharing

there are cities that are also limiting car access or forcing a payment tax for entry, the so-called ecological tolls. The objective is to encourage the use of alternative transportation methods in order to decongest the roads and decontaminate the air in the city centers.

The idea of transport sharing represents a form of more sustainable mobility. It arose with the goal of making travels more economic and sometimes even more social, either because they involve more costs, are longer or because they are made more frequently, for example when going to college or work.

Carpooling can be done for personal reasons, like saving money, but also for more altruistic reasons like helping the environment. There are several factors that may influence the decision of choosing to carpool. These factors can be personal and psychological characteristics, economic situation, geographic location, trust issues in other users involved, among others. The mentioned characteristics can act as a barrier to embrace this concept [BSB⁺10]. Some of the most influential factors may be [Cam02]:

- Distance – The greater the distance to go, the more inclined a person will be to carpool;
- Number of vehicles in the household – In households with fewer vehicles, people are more inclined to share transport;
- Income – People with low incomes usually are more receptive to the concept;
- Household members – In case there are minors within the household, people may not be as much willing to do it;
- Transportation characteristics – Comfort, costs, safety, among others.

Sharing transport with family, friends and/or acquaintances is already a very common practice for people who travel common or coincident routes. Giving a ride and hitchhiking is quite common among people with a good relationship. However, there are people who end up travelling on a daily basis on similar routes and if they were aware of it, they could share transport in order to reduce travel costs.

While carpooling, the role of each person can be:

- Only as driver;
- Exclusively as a passenger;
- Both functions (driver and passenger).

Four major types of carpoolers (people with whom carpool is done), can also be identified [Tea87]:

- People from the same family – This first case is the one that has greater representativeness when sharing transport due to relations of affinity and sometimes geographic proximity;

Ride Sharing

- Friends and acquaintances – Very common when sharing transport for leisure purposes, such as going to the cinema or to the beach;
- Co-workers – Mostly travelling to work-related places;
- Strangers — Possible through transport sharing platforms.

2.2.3 Types of Carpooling

The types of person with whom people can do carpooling is directly related to the way it can be done. Regarding the ways of Carpooling, there are at least two forms: Personal Carpooling and Organizational Carpooling.

Personal Carpooling

Personal carpooling is a form of transportation sharing that can be done with people from the same family, friends, acquaintances or even strangers. These are usually more sporadic journeys on which people want to share costs associated with the trips.

Organizational Carpooling

This way of carpooling is done with co-workers and/or members of the same organization.

One of the determining factors for the organizations' adoption of carpooling is its size. The larger the organization, the greater the possibility of having similar trips made by the members of the community that compose it. There is then a need to promote, stimulate and encourage people to adopt this concept [BSB+10].

In addition, there are areas with more limited accessibility for public transportation, such as technological and/or industrial parks, where a large number of people with compatible schedules go to on a daily basis.

In the case of educational institutions, carpooling can allow the sharing of knowledge between the participants and a facilitated integration in college activities. Currently there are already universities implementing a carpooling program.

The implementation of an organizational carpooling system not only brings advantages to the organization but also to its employees and to the society in general.

Peninsula College - An example illustrating this idea is the *Peninsula College*. Located in Washington, United States of America, it has already implemented a carpooling program to support sustainable mobility and reduce the number of vehicles on campus. In order to join in, students

Ride Sharing

and college staff must meet certain prerequisites, such as having a vehicle available, guarantee the transport of at least three people (driver included) and completing a form on a quarterly basis. One way to encourage the student and teacher staff to join is by offering free parking for the participants during the period on which the carpooling license is valid. However, these parking spaces are subject to availability as they are limited [Col17].

Texas A&M University - Another example of an educational institution that encourages this method of transportation sharing is *Texas A&M University*. Students have created a non-profit organization where they intend to offer rides to potential prospects within a certain time period. The service has been active for some years now and has seen a great adoption [Stu17].

Ride Sharing

Chapter 3

Mobility Platforms

In this chapter, the study of the state of the art is presented and described, and products by some means related to this theme are presented. The concept of carpooling is not new, even though it has been gaining more relevance recently. It is intended to describe some applications that already exist in this domain and the way they work.

The proliferation of the use of smartphones and their use of the internet, contributes to the emergence of new mobility applications with the purpose of planning certain trips.

The analysis of the existing systems allows for the evaluation of functionalities, the discovery of development opportunities and the improvement of concrete points. This analysis of mobility platforms was divided in two sets: Platforms that focus directly on carpooling and platforms where it is possible to apply the same concept, also having similar functionalities, even though it is not their main focus.

3.1 Carpooling Platforms

In these platforms, the concept of carpooling is fully applied, being their main objective. Some that are considered relevant for the study of the state of the art are mentioned below.

3.1.1 Boleia.net

The web application was launched in Porto in 2013 and is open to everyone who wants to search for or offer a ride. To be able to start using the service it is necessary to create a free account or login through *Facebook*. It currently has more than 11,000 likes on *Facebook*, which proves some relevance.

The platform works on the basis of advertisements, where people fill in fields like the desired starting point and destination. The search then shows a list of both offers and transport requests,

Mobility Platforms

date and time of departure and arrival, as well as the cost of the trip per person. In addition, users can check the driver's profile along with their rating given by other users and the number of seats available to share. The driver profile evaluation was designated to indicate reliability of each driver registered in the service. The purpose of the platform is not for advertisers to have profit on sharing transport, but to share a part of the costs of traveling. Interested parties can contact advertisers. Payment and reservation is then agreed between both.

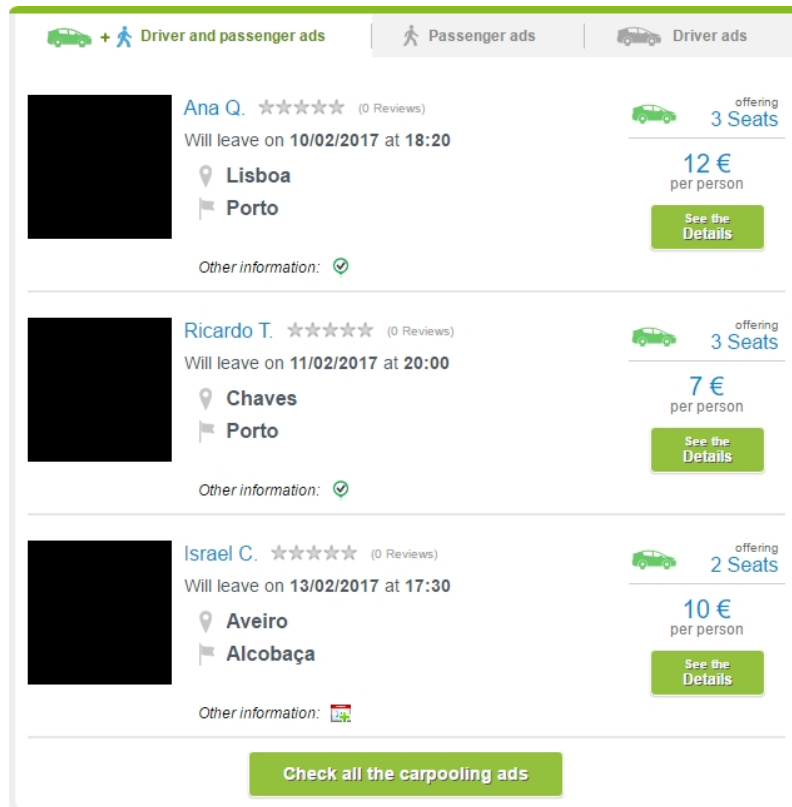


Figure 3.1: *Boleia.net* driver and passenger advertisements [Lda16a]

In addition, they also have two specific sub-services with similar functionalities: *Empresas à Boleia* and *Dragões à Boleia* [Lda16a].

Empresas à Boleia

This first sub-service is focused on companies and organizations where employees with similar routes can share transport if they wish to. The goal is for workers from the same institution, or from the same city councils, hospitals, among others, who travel similar routes in their daily lives, to share transportation with each other. The service is free for users, but in order to use and benefit from it, their companies must join a subscription by choosing one of the available plans. The service is free for companies with a maximum of 50 employees, but there are other plans for medium and large companies. The first plan is for companies between 50 and 250 employees and has a cost of

Mobility Platforms

1000€/ year. The second plan, for more than 250 employees, has a cost of 4000€/ year. In both cases, there are other advantages that entities can benefit from. Some of the entities that have joined this service are *Deloitte*, *Taguspark* and *KuantoKusta* [Lda16c].

Dragões à Boleia

This second sub-service is intended to join supporters of *Futebol Clube do Porto* who intend to share transport when going to games and sport events on which their club participates. The service, launched in early 2015, is supported by the team in question and was also advertised on their official website [Por15].

In addition, it also allows the purchase of tickets for the games and sporting events of the team. Finally, it is pointed out as an advantage that users can meet new people with common interests, in this case, the support for the team [Lda16b].

3.1.2 BlaBlacar

This company was founded in France and has been providing carpooling services since 2006. Currently the network covers 22 countries, mainly European. The creator of this service claims to have conceived this network when he realized that most of the vehicles on the road circulated almost empty. This social network of transport sharing arrived in Portugal in 2012 and continues to expand to more countries.

The screenshot displays the details for a BlaBlacar trip from Porto to Lisbon. The route is titled "Porto → Lisbon" with a "See route" link. The offer was published on 09/02/2017 and has been seen 15 times.

Key Trip Details:

- Pick-up point:** Porto, Portugal
- Drop-off point:** Lisboa, Portugal
- Date:** Today - 06:30
- Details:** Pick-up in a 15 minute window, Large (suitcase), Max. 15 min. detour.

Driver Information: Miguel T, 37 y/o, 4.9/5 - 9 ratings, Good driving skills — 2.8 / 3. Car: Fiat Ducato, Colour: white.

Price and Booking: Price per seat is £14.50. There is 1 seat left. A "Request to book" button is available, along with a "Guaranteed arrival" icon.

Schedule and ride group: A table shows the route from Porto (06:30) to Lisbon (~09:50). The driver Miguel T (37 y/o) is shown with a "Booked seat" icon.

Figure 3.2: *BlaBlacar* trip details page [Bla17]

Mobility Platforms

The platform works on the basis of transport-sharing advertisements, where users can define departure location, arrival and cities where they intent to go through. On a trip page, the data previously mentioned can be seen, as well as other, more specific details. For example, the route to be taken, if it is made by highway, the maximum detour time the driver is willing to take to catch someone, the vehicle that will make the trip, the number of seats available and the price for each one. In addition, the profile of the driver and passengers associated with the trip can be seen. Each driver has an overall rating, from 0 to 5, given by other users with whom the driver has already travelled. Passengers can also rate the driving on a scale of up to 3, which is also visible by everyone. The driver can set four basic preferences, which are: enjoys talking, listening to music, allows smoking in the car or carry pets, while traveling. In a profile a user can also have a photo or associate it to a *Facebook* account. If someone is interested in sharing transport, can contact the advertiser through the platform and book a seat. The payment is made through the service, being the company an intermediary of the transaction [Bla17].

This service is available through a web application and mobility applications for *Android* and *iOS*. The way these work is similar to the web application, having only a more simplified interface, which is typical of mobile applications. In the case of the *Android* application, it accounts for 503,581 ratings, has an average rating of 4.4 out of 5 and has been downloaded between 10,000,000 and 50,000,000 times [Sto17a]. On the other hand, in the case of *iOS*, the rating is similar, having 4.5 in 5 [iS17a].

3.1.3 Galpshare

Lisboa Porto 3 lugares disponíveis. 2 x Mês
Atualizado em 26/09/2013

Dados Gerais

Origem: lisboa Destino: porto

Tipo: Recorrente Transporte: Carro
Procuror: Passageiro Custo Diário: 74,28 €
Custo Semanal: 148,56 €

Horários e preferências do percurso

	SEG	TER	QUA	QUI	SEX	SÁB	DOM
Partida	--:--	--:--	--:--	--:--	19:00	--:--	--:--
Regresso	--:--	--:--	--:--	--:--	--:--	--:--	19:00

Preferências

	Sexo	Fumador
Preferências	Indiferente	Indiferente

Partilhar Percurso

Mapa Satélite

Portugal

Lisboa Porto

Google Dados do mapa Termos de Utilização

Figure 3.3: Galpshare trip details page [Gal17]

Mobility Platforms

Galpshare is a free to use platform developed by *Galp Energia*. This aims to promote mobility in a more sustainable way for people's daily lives and thus reduce the huge amount of vehicles that travel with only a single occupant. Users can announce itineraries they want to share, including the date(s), time(s) and some more basic preferences like whether they accept smokers on the car or not. The trips can be something frequent like going to work or punctual like going to concerts and sporting events. The advertisements can be for sharing car trips, but also for public transport, cycling or even walking [Gal17].

3.1.4 ToGoCarpool

ToGoCarpool is a mobile platform focused on carpooling, with a user base mostly from India, available only for the *Android* operating system. The rating given by users in the *Google Play* store is 4.7 out of 5 [Sto17i]. The purpose of the application is to create communities of users who are interested in sharing transportation on their journeys. After registering, a user can enter the details of a route and also the place of departure, arrival and preferences of users with whom they want to share transport, such as gender. The user is then added to a group chat of people with similar routes. There, it is possible to announce the offer of transport or see the existing ones and their details, such as user name, photo, place of departure and arrival. Besides that, the system also shows a calculation of a tariff that would be fair to pay when dividing transportation. However, the payment is agreed between the driver and the interested parties and the value shown is only a suggestion. If there is interest in sharing, there is the possibility of sending a message or calling the advertiser to set details [Ltd17].

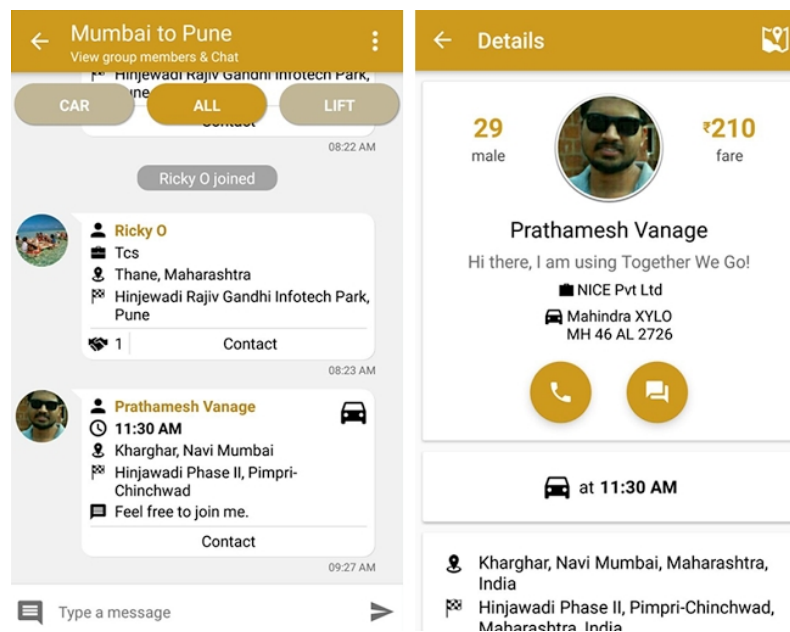


Figure 3.4: *ToGoCarpool Android* application: group with trip ads (left) and trip details page (right) [Sto17i]

3.1.5 Pool My Ride

This application way of working is similar to the platform described before. After creating an account, a user can view and create advertisements for transport sharing. There is also the possibility of connecting to other people through friendship requests. In the advertisements list, many details can be seen, like the route, how long the advert was placed and how much time is left for the trip in question. On each page it is even possible to see more details, such as information of the profile of the advertiser, ways to contact the users and the estimated time of departure and arrival. A passenger can reach an agreement with a driver through private messages. The application is available for the *Android* and *iOS* operating systems [Log14] [Sto17h] [iS17g].

3.1.6 WazeCarpool

The *Waze* company is providing a yet experimental, and only available in in San Francisco (United States of America) and Israel, carpooling service. *WazeRider* is the name of the application and is available for or *Android* and *iOS* mobile devices [Sto17k] [iS17i]. Each user can define their weekly plan and offer/ask for transportation. By the searches and offers, the system makes the passengers assignments. When it gets a match, the passenger can see the profile of the driver, the cost to pay for the trip, among others, being also able to accept or refuse the assignment. When the trip is happening, the user can see on a map where the driver is and how close is to the user [Waz17].

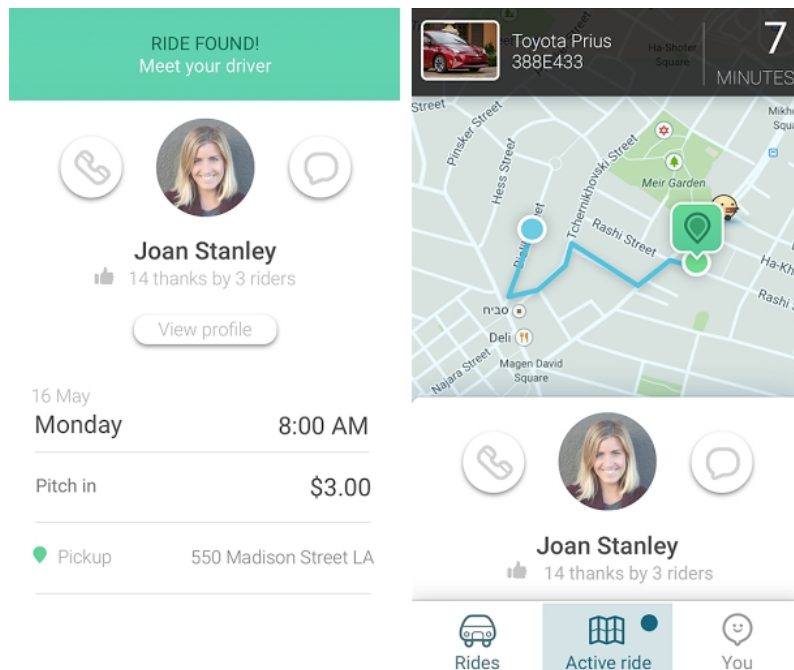


Figure 3.5: *WazeCarpool Android* application: ride found screen (left) and map showing an active ride (right) [Sto17k]

3.1.7 RideAmigos

RideAmigos provides a web platform focused on companies, institutions, organizations, universities, among others, with the goal of analyzing, managing and planning employee trips. With this service, users can plan their trips from the point of departure to the destination, being able to travel in their daily commutes with other colleagues, whether by carpooling, walking, cycling or public transportation. There are then various statistics that can be accessed in order to make these shares more efficient, even from a business point of view, like travelling to work events like conferences [Rid17].

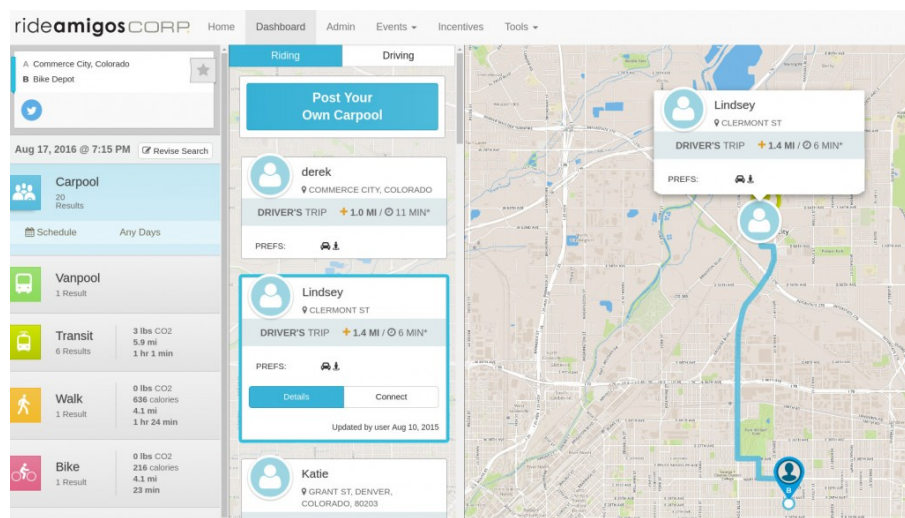


Figure 3.6: *RideAmigos* dashboard [Rid17]

3.2 Other Platforms

In this section, some of the many mobility platforms will be mentioned, on which the main goal is not carpooling but where the concept can also be applied. Their study is relevant because they have interesting functionalities for this type of systems.

3.2.1 Uber

Uber was founded in San Francisco in the United States of America in 2009 but only launched its service in 2010. Since then, it has gradually expanded and is currently in 67 countries around the world [Ube17b]. In Portugal it has been operating since July 4, 2014 [Ube17a] and has generated discord and contestation from other similar service providers [dN16].

The purpose of the platform is for a user to be able to request a vehicle, in an easy way, to make a trip. It is available in the main application stores for *iOS*, *Android* and *Windows*. In these stores,

Mobility Platforms

the app has an average rating of 4.3 out of 5 stars from the users, which is quite good considering its context [Sto17j] [iS17h] [Mic17].

The method of operation of the application is quite simple. At the time of its execution, the main screen shown contains a map showing the user's current location and the approximate location of the vehicles performing the service at the moment. To request a vehicle, simply enter the desired destination. Then, a possible route, an estimate cost of the trip and the places available on the vehicle are shown to the user. At the end of the trip, its payment is made through a credit card or *PayPal* account registered in the application. There is also the possibility to schedule a trip from a location to a destination defined by the user. Also, at the end of each trip it is possible to evaluate the driver, taking into account the satisfaction of the service provided. *Uber* has been offering new services, even if temporarily. Some of these services are briefly described below.

uberGREEN

In a period of three months, between March and June 2016, *Uber* launched the *uberGreen* pilot project. The goal was to provide users with fully electric car trips, which results in no emissions of greenhouse effect gases such as carbon dioxide on those trips.

In addition, by using measuring devices implemented in the cars, users were able to verify the ecological footprint, which is the volume of emissions that would have been expelled if the vehicles were not fully electric.

This more ecological way of travelling has helped the environment from the harmful effects of fossil fuels by reducing the emissions of gases. At the end of this initiative, it was estimated that about 57 tons of carbon dioxide were saved from being sent to the atmosphere. This service has become available on an ongoing basis [Pip17] [dN17].

uberPOOL

Uber entered the carpooling field by providing the *uberPOOL* service. The goal is for people who wish to travel on similar routes to share the same vehicle. The trip is more economic for the passengers since they divide the cost. When requesting a vehicle for this type of transportation, users may or may not share the trip with other persons. If there are no more interested users at the time, the trip is performed like a normal *Uber* trip. This service is available in more than 30 cities around the world [Ube17c]. In Portugal, it was only available in Lisbon during an experimental period, from November 4 to 13, 2016, during a technological conference, the *Web Summit* [Sum17] [Fer16].

Uber for Business

Uber for Business is a variant of the normal *Uber* service, directed for small to large companies. It is a business solution for institutions that allows their employees to do work trips with *Uber*. For administrators, there is a platform for managing employees and travels, where they can also check the history, statistics and expenses. In this case, the concept of carpooling can also be applied

because employees with similar routes can share transportation, which results in lower costs for the company [Ube16].

3.2.2 Cabify

This international transport company offers services generally similar to *Uber* [Cab17]. It operates in several countries, mainly Latino ones, and in Portugal the service is available in Lisbon, Porto and Madeira. Through the mobile applications available for *Android* and *iOS*, it is possible to register and use the service. The overall average rating given by users in these application stores is slightly lower than *Uber*'s, with an average of 4 out of 5 [Sto17b] [iS17b].

3.2.3 MEO Táxi

MEO Táxi is a taxi request service, from the company *MEO*, that can be used through mobile devices, simply by creating an account. The service is already available in several Portuguese cities. The way it works is similar to *Uber*'s application. When requesting a taxi, it is possible to define desired characteristics according to the need of the user, such as the language that the driver speaks, permission to transport animals, if the vehicle has air conditioning, among others. After making the request, it is possible to follow, in real-time, the route performed by the taxi until it reaches the user at the starting point [Tá15]. At the end of the trip it is possible to evaluate the trip on a scale of 1 to 5. The application is available in the *Android* and *iOS* stores and has an average rating of 3.7 [Sto17f] [iS17e].



Figure 3.7: *MEO Táxi Android* application: setting the vehicle preferences (left) and map showing the taxi on his way (right) [Sto17f]

3.2.4 mytaxi

mytaxi is a service that allows the user to book a taxi through a browser or a smartphone application, available for *Android* and *iOS* [Sto17g] [iS17f]. It possible to follow the vehicles on a map in the application and select the one the user likes the most, using the descriptions provided. Payment is made exclusively through the application. At the end of a trip, the user can also rate the experience up to 5 stars. Besides that, the user can share the trip with others if wishes to. In Portugal this service is only available in Lisbon [Gmb16].

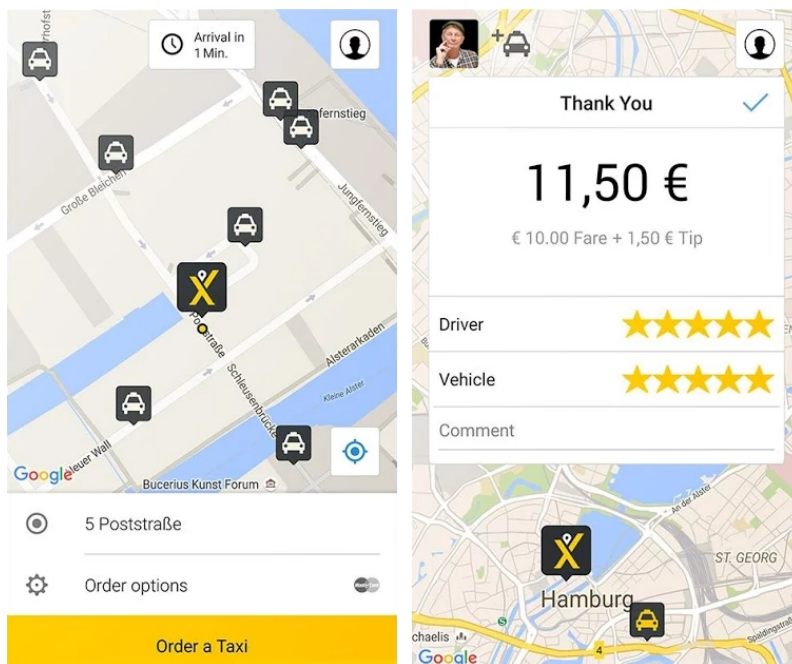


Figure 3.8: *mytaxi* Android application: map showing available taxis (left) and rating screen at the end of each trip (right) [Sto17g]

3.2.5 car2go

car2go is a car rental service for travelling. In order to do that, the user only needs to download and install the application, available for all the main mobile platforms [Sto17c] [iS17c]. The company has several vehicles parked around some cities. By using the application, it is possible to verify where they are parked and request one to use. The application is used to unlock the vehicle and, from that moment on, the payment starts. The user is the driver in this service. While driving, tolls and parking lots do not need to be paid as they are included on the rent price. At the end of using the vehicle, it does not need to be taken to any specific location, so the driver can leave it in any place. These trips can also be shared with other people. However, this service was not available in Portugal [cN17].

Mobility Platforms

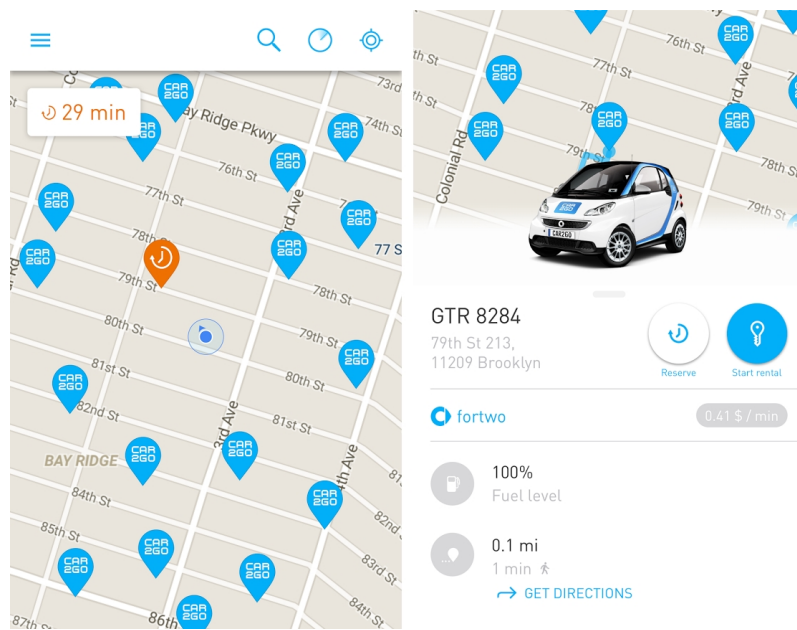


Figure 3.9: *car2go* Android application: map showing available vehicles (left) and vehicle details (right) [Sto17c]

3.2.6 citydrive

citydrive is a web and mobile platform available for *Android* and *iOS* [Sto17d] [iS17d], which provides a service similar to *car2go* [cN17], but is only available in the city of Lisbon. The service works the following way: after a registration, users can locate a car, book it and drive it. The vehicles have a usage rate charged to the minute. They also offer a service of carsharing for company employees' work travels. The operation is the same as above, and only differs in the payment that in this case is charged to the company in question [Cit15].

3.3 Mobility Platform Features

After an analysis of the state of the art of the domain where this work fits, it can be concluded that there are several aspects common to most platforms. Therefore, some of these can be summarized in the following list:

- Announcements – Possibility to offer and search for transport sharing advertisements;
- Profile – Personal profile where users can define some preferences with whom you want to share a car, for example, non-smokers, transport of animals and way to contact them;
- Rating – Rating of the driver/passengers in order to create a reputation system to make the service more credible and secure;
- Map – Responsible for representing the position of people involved in a carpooling trip.

Mobility Platforms

Chapter 4

FEUPooling: Carpooling Platform

In the specific context of an organization like *FEUP*, there is already a known community of users for whom various behaviors, routines, trips and schedules can be anticipated. Therefore, the community can benefit even more from the concept of carpooling than in a more open and generic context.

4.1 Problem

This community of students, technicians, researchers and teachers of *FEUP*, at the moment of the writing of this report, exceeds ten thousand [Sil16]. It is important to point out that of these, there are some groups that do not have access to private transport methods, which can make carpooling more difficult. Most students also do not have a driver's license on their first year of college. In addition, there are those who only travel by public transportation or go on foot, because they have no other viable choice. However, it does not mean that they are excluded from carpooling because they can ride share with another person who has private transportation and is interested in sharing the costs associated with the trip.

The geographical area in which *FEUP* is inserted is highly urbanized. There is, therefore, a high concentration of houses, educational institutions, infrastructures and even a hospital unit. The huge amount of people who travel daily through this area causes high levels of traffic, with a high concentration of vehicles, especially in the early morning and late afternoon. This concentration also varies according to the school calendar.

The surrounding area has good accessibilities, with public transportation such as metro and buses, and even proximity to traffic routes such as *Via de Cintura Interna*, *Estrada da Circunvalação* and *A3/A4* [Rei11].

At *Polo Universitário da Asprela*, the available parking is much lower than the demand. During the day there is a shortage of parking, as these places are occupied by the vehicles of students and workers who travel there, which are parked for long periods of time.

FEUP has parking lots for students, staff and teachers. However, the number of parking places is not enough for the high demand. Besides that, public parking in the surrounding area has been taxed since June 22, 2016 [Sil16]. The lack of parking sometimes makes students late for classes due to them having to look for a spot to park.

4.2 Friendly Mobility Study

From November 13th to 27th 2015, the *Comissariado para a Sustentabilidade da Faculdade de Engenharia da Universidade do Porto* carried out a survey entitled “Mobilidade Amiga do Ambiente: Car Sharing e outras Alternativas” to the *FEUP* community, which includes students, technicians, researchers and teachers.

This study had a considerable participation: about 1578 participants, of which 1277 were students and 301 permanent members. The analysis of these surveys allows for a better understanding of the community in question, its behaviors and the pre-disposition for the concepts of car sharing.

The questions asked can be grouped into analysis groups. In this specific case, some questions are directed to students who are not living in their usual residences during class time, traveling back home at the weekend and others to students who travel daily to *FEUP*.

Regarding the 38% of students who are displaced from their usual residence, 94% stated that they return home at weekends. As for the form of transport, the majority says that they use public transportation (60%), followed by car sharing (23%) and others travel alone by car (15%). Considering only those who travel alone, when asked why they do it, the answers were:

- 74% – Independence in the management of journeys/schedules;
- 44% – Personal comfort;
- 29% – Not knowing any car sharing platforms;
- 28% – Lack of an exclusive car sharing management platform for the *FEUP* community;
- 18% – Personal security;
- 8% – Other reasons;
- 3% – Road safety.

Still considering only those who travel alone at the weekend, when asked if they would consider sharing their car, the majority replied yes (44%), followed by maybe (42%) and no (14%).

FEUPooling: Carpooling Platform

Analyzing the case of those who travel daily to college, regarding employees, 75% travel by car and of those, 77% travel alone. Regarding students, the majority (67%) said that they use public transportation or go on foot and 30% travel by car, with only 32% of them sharing transport. Considering the collaborators or students who travel alone, when asked why they do not share their rides with others, the response percentages were:

- 80% – Independence in the management of journeys/schedules;
- 27% – Lack of an exclusive car sharing management platform for the *FEUP* community;
- 25% – Personal comfort;
- 23% – Not knowing any car sharing platforms;
- 9% – Personal security;
- 4% – Unawareness of potential candidates for sharing commutes;
- 2% – Take the children to school;
- 1% – Road safety.

Considering yet those who travelled by car alone, the majority stated that they perhaps would share the transport (54%), 27% would certainly share and 1% would not. Finally, when asked if they would share the car if there were incentives, the response rates change, although not very significantly. Regarding incentive suggestions, in order of interest, the participants mentioned:

- Guarantee of a parking spot in the college park;
- Sharing of travel expenses;
- Financial retribution or tip discounts;
- It would be enough to have a platform of transportation sharing for them to use.

4.3 Solution

All things considered, the conclusion was that there was a great need for *FEUP* to have one platform like this. The existing platforms were not ideal because they do not have desired features, such as reward systems. It was then intended to develop a product to *FEUP* because there was none at the moment of writing.

The solution, codenamed *FEUPooling*, consisted mainly on the development of a carpooling mobile application for the *Android* and *iOS* operating systems. The target audience is the *FEUP* community that wishes to share their daily commutes. A web platform was also developed for

FEUPooling: Carpooling Platform

administrative purposes. It was designed for administrator(s) to control the whole system and take actions when necessary.

Although the great majority of *FEUP*'s students are Portuguese, every year a considerable amount of foreign students come to study abroad. Therefore, the applications were developed in English so that every student could use them. The fact of not having a private vehicle should be no impediment to use the system as some students will only choose the role of passengers.

Mentioning the innovative aspects, it can be highlighted that it is a platform directed to a closed community of users from the same organization, which can help simplify various aspects of carpooling, making adoption easier. Integrating this platform with the existing organizational information systems is easier to achieve than in a broader and more general context.

Besides that, it was decided to implement simple gamification mechanisms, which is the use of gaming mechanisms applied in other contexts in order to make something more appealing. In this case, a system of points and rewards was implemented where it is possible to win them by carpooling and trade them for something to be defined by *FEUP*. It is then a mechanism for encouraging and rewarding the community's adherence and usage of the platform. This subject will be discussed later on.

This free service aims to allow users to offer and ask for transportation to travel to and from college, so people with geographical and time compatibility can do these trips together. It is then focused on the planning of a commute. This mobility solution is more sustainable than when compared to more traditional forms of transportation. All interested students should be able to access and use the platform, which allows them to share transport in an easy way, without them being concerned about their safety.

The implementation of this service was proposed and endorsed by *Comissariado para a Sustentabilidade da FEUP*. The starting point of the project was a prototype of a web platform developed in the academic year 2016/2017, in the Curricular Unit *Laboratório de Desenvolvimento de Software*. The new system was built from scratch but having in account the first prototype and the main concepts involved therein. The users' needs and requirements gathered were taken into account, which resulted in a new gathering of requirements not being performed.

A meeting with the stakeholders was conducted at the beginning of the project where what was supposed to be done was discussed. Tasks were ordered by priority and were implemented according to its priority and relevance. During the iterative development of the product feedback was gathered and improvements were made according to these suggestions.

Putting it simply, the main challenge was to design a good usable product. Some of the usability goals of the system were [DB16]:

- Efficient to use by allowing the user to perform the desired tasks quickly;
- Minimum amount of errors, or an easy recovery from these;

- Enable a good user experience by being pleasant to use;
- Easy to learn for first time users;
- Easy to remember how to use for returning users;

All the subsystems of the project will be described ahead. This chapter also contains the specification of the actors and their respective users stories, working as the agile documentation of the project requirements.

4.4 FEUPooling Backend

The backend system was the first application of the project to be written. Following a client-server model, it was mandatory to have a backend providing the service to the service requesters, which in this case are the clients. This central server handles some of the business logic, data storage and provides *APIs*.

The backend was written in *PHP* using the framework *Laravel*. This framework aids the development by simplifying some common tasks. It also provides tools needed to develop large and robust applications. *Laravel* follows the Model-View-Controller (*MVC*) architectural pattern. This software architectural pattern is applied by dividing the application into three components, which are the model (business logic), the view (presentation logic) and the controller (inputs). On *Laravel*, the business logic is usually represented by data models while the presentation layer is represented by the web pages [lar17].

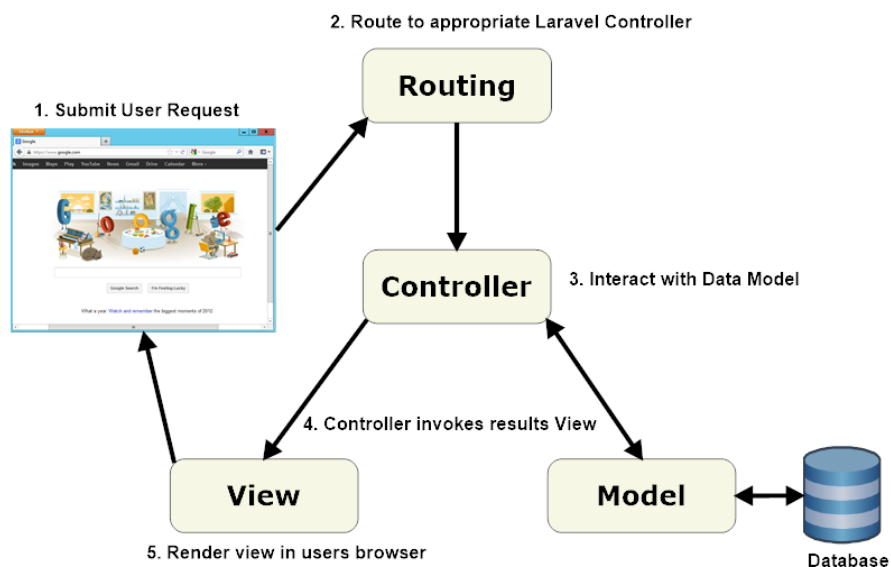


Figure 4.1: *Laravel MVC* [lar17]

FEUPooling: Carpooling Platform

A database was necessary to store all the data required for both the mobile and web parts of *FEUPooling*. The relational database management system used was *MySQL*. The database schema, which is its structure, is described in appendix C and is summed up below:

- *admin_login_tokens* – Stores login tokens to administrators of the *FEUPooling Web*.
- *login_tokens* – Stores login tokens for the users of *FEUPooling Mobile*.
- *notifications* – Stores notifications for users of *FEUPooling Mobile*. These notifications can be triggered by another user or by an administrator. Notifications triggered by other users have a trip associated, while administrative notifications are triggered by the administrators of the system, which an example is the acquisition or exchange of points.
- *on_trip_positions* – Stores user positions for ongoing trips.
- *places* – Stores places associated with trips.
- *points_histories* – Stores the history of points associated with a user, which can be of a negative or positive type.
- *ride_requests* – Stores the ride requests, triggered by the users when these request to join a trip.
- *settings* – Stores some system settings, such as the points to be acquired per trip as a driver and as a passenger.
- *trips* – Stores trips created by the users.
- *trip_passengers* – Stores the passengers of trips.
- *trip_validations* – Stores all the users' position checks for an ongoing trip.
- *users* – Stores all the users of the system.
- *user_admins* – Stores all the administrators of the system.
- *user_driver_ratings* – Stores all the ratings received by the users as drivers.
- *user_rider_ratings* – Stores all the ratings received by the users as passengers.

FEUPooling: Carpooling Platform

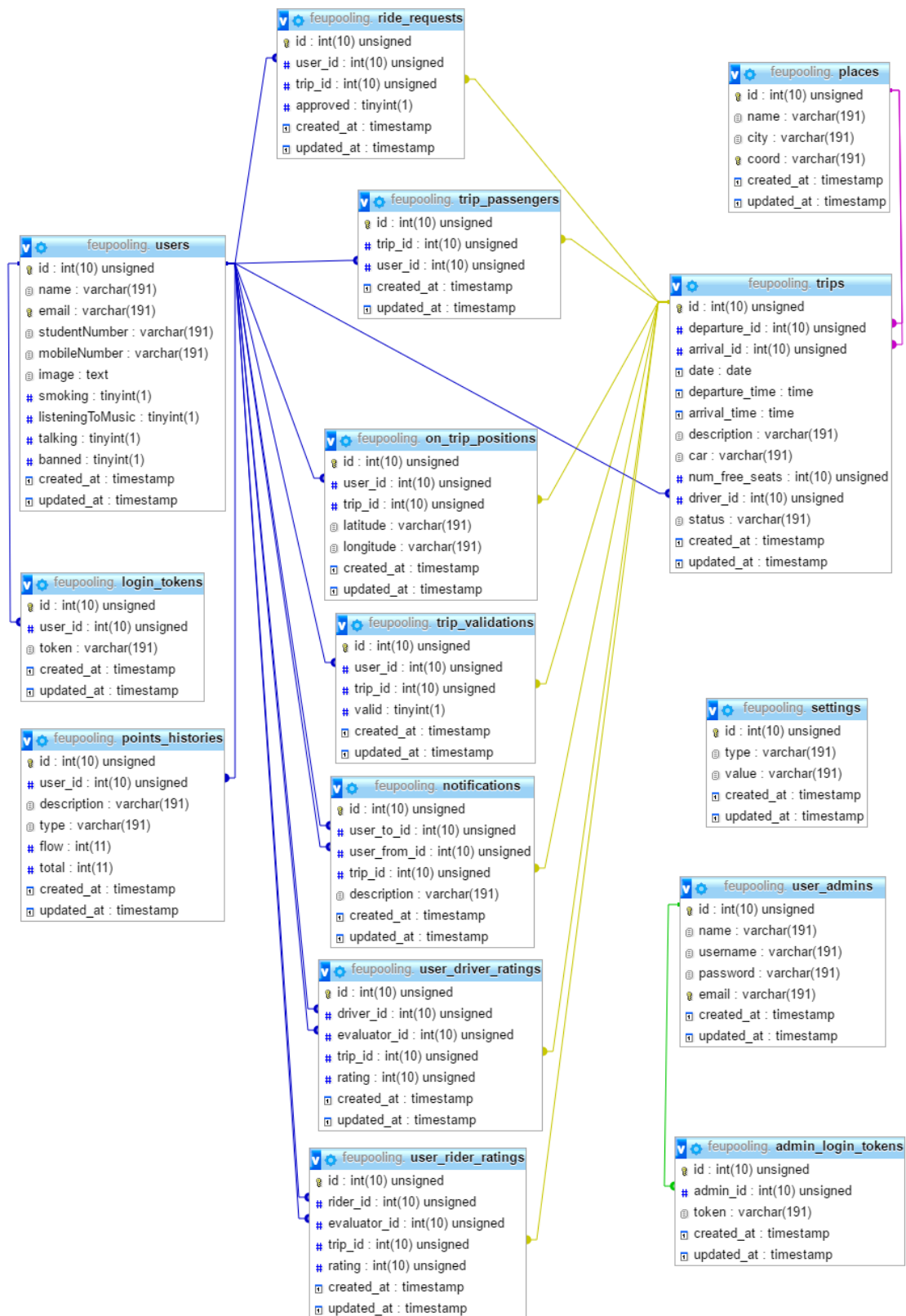


Figure 4.2: Database Schema

A *REST*-based *API* was implemented so that the web and mobile applications developed can communicate with this backend. The applications can provide or retrieve the information stored within. The *API* calls are described in appendix [A](#).

4.4.1 Federated authentication

One of the main reasons pointed as an obstacle to the carpooling platforms is that it is not easy to ensure that the user base is trusted and secure. As of that, an authentication system had to be implemented. That being said, in order to guarantee that the mobile application was only used by students from *FEUP*, the best way to do it was to use the *Universidade do Porto* federated authentication system.

With the aid of *FEUP*'s computer centre (*CICA*), federated authentication was implemented on the *FEUPooling* server. The *UPORTOaai* (authentication and authorization infrastructure) uses a three component infrastructure [[UPO10](#)]:

- Identity Provider (IdP) – Authenticates users with the Service Provider and returns attributes;
- Service Provider (SP) – Protects and authorizes access based on the required attributes sent by the Identity Provider;
- WAYF (Where Are You From) – Aims to redirect unauthenticated users to their home institutions.

The end user authenticates with the Identity Provider, where a session is created for the user. The session is verified through a communication between the Service Provider and the Identity Provider. Access control to the services is possible due to attributes transmitted through Security Assertion Markup Language (*SAML*) assertions. *SAML* is used in the communication between the identity provider and the service provider. This communication is encrypted by mechanisms such as *SSL/TLS* and validated by *XML* signatures [[Cen](#)].

The software used to implement the single sign-on (*SSO*) on *UPORTOaai*, which is a session and user authentication service using some credentials in order to access multiple systems [[Rou16](#)], is *shibboleth*. This open source software package implements federated identity standards (*SAML*) to provide a federated single sign-on and attribute exchange framework [[shi17](#)].

In order to enable federated authentication on the server, a *Laravel* package [[Cot17](#)] for *Saml2* integration as a Service Provider was used, and the Identity Provider and Service Provider were configured. The login redirects the user to the Identity Provider, which then redirects the user back to an endpoint at *FEUPooling*, which processes the response and then fires an event. The handling of the event was the next step required. If the login is valid, the attributes will then be analyzed, otherwise the user will be redirected to an error page.

The attributes received required for the system are the name, institutional email, student number and profile picture. The first step is to verify if the user is a student and from *FEUP*. If so, a token

will be generated and in case the user is already on the database, the profile picture will be updated if it is not up to date. This is necessary because it is the only field that can be changed on the institutional system. Nevertheless, the user might not be on the database, which means it is the first time using the system. In that case, a new user will be created on the database.

For each mobile device the user logs in, an alphanumeric token is generated, with a length of 128 characters. This token is guaranteed to be unique. A user can have several valid login tokens because the application can be used in more than one mobile device, but a token is only associated with one user. The user identifier and the login token will be then put in session and the user will be redirected to a new route.

4.4.2 Gamification System

Gamification is the application of game mechanics and dynamics in non-gaming contexts. It aims to make tasks or actions more appealing and therefore increase people's engagement [SRV13].

This technique has been increasingly more adopted in several areas mostly as a marketing method to increase the users' motivation to use a specific product. Gamification can be achieved by adding game mechanisms to applications or businesses. Some of these mechanisms are [Per14a]:

- Badges;
- Points;
- Leaderboards;
- Levels;
- Challenges or quests;
- Virtual currencies;
- Content unlocking or rewards;

In the case of this carpooling platform for institutional purposes, it is necessary to entice students to use the application and encourage them to share rides with other users. Therefore, a simple gamification points system was implemented that aims to create a solid user base. Apart from a points system, it is also a reward system, because users are given points whenever they perform a trip, that later on can be traded for something else. The rewards must be things that students want or need, otherwise this gamification system might not motivate them the way it is intended to. A practical example of what a reward could be is a guaranteed parking spot for a limited period of time by the exchange of points. Nevertheless, all rewards must be defined later on by the system administrator(s).

If a student needs or wants to travel from or to college, by using a private vehicle, and is aware of an official mobile application that not only allows expenses to be reduced, but also the gathering of points that can be exchanged for some interesting rewards, it can be said that user would almost certainly join the platform.

The bright side is that this system can encourage many students to start to use the application. The not so good side is that users could start taking advantage of the system by corrupting the ideal behind it in order to gain points without actually performing trips to or from college with colleagues. Therefore, the trips must be validated before points can be awarded. The system administrator(s) can, whenever desired, rectify the attribution of points but the decision to consider a trip valid or not is always of the system itself.

In order to validate trips, server-side cron jobs were implemented. These are scheduled tasks performed within a defined time interval. To aid the implementation, a *Laravel* library [lie17] was used. This library aims to simplify the creation and usage of cron jobs.

The cron job verifies, for the all trips of that day that are occurring, the positions of the drivers and passengers. The passengers positions are compared to the driver's and the result is stored on the database. If the trip arrival destination is *FEUP*, from the moment a passenger and driver position is close to each other (less or equal than 0.25km), this means that the passenger got in the driver's car, and their positions will be verified until the end of the trip. It was implemented this way because a driver can start a trip with a passenger or can pick that person in the middle of it. In the reverse scenario, if the trip's departure place is *FEUP*, the passengers' positions must coincide with the driver's from the beginning and when they stop coinciding, it means that the passengers already arrived at their destination.

When the driver stops the trip, a final validation is done in order to assign points to the users. The first validation is to verify the position where the trip stopped. If the driver position was further than 1km from the declared arrival position, the trip is considered invalid and nobody receives points. Else, for each passenger, the percentage of valid entries from the previous step is calculated. The trip is considered valid for that passenger only if at least 90% of the entries are valid. Regarding the driver, if no passenger receives points, which means that no route matched its destination, the driver will also not receive points. This means that if at least one passenger acquires points, the driver will also receive them. Finally, all users involved on the trip will receive a notification stating that they received (or not) points after the validations are conducted.

4.4.3 Reputation System

A reputation system is a way to earn another person's trust on interactions. Particularly on interactions that are not in person, of which strangers on the internet are an example, trust can be a barrier to communication and is not easy to build. When interacting with total strangers, sometimes great opportunities can appear, but there's always a big risk of a person taking advantage of another.

FEUPooling: Carpooling Platform

“A reputation system collects, distributes, and aggregates feedback about participants’ past behavior” [RKZF00]

This project, though, is directed to a closed and known community of users, which means there is a smaller chance of dishonesty or identity theft. Even though, implementing a simple reputation system would be beneficial to all users. That being said, a simple rating star system was implemented, from 0 to 5, that allows users to give feedback about other users with whom they have shared trips. Rating is a quantitative way of creating a reputation system. There are also qualitative ways of doing it, for example not allowing a pseudonym to be used, which on this case does not apply due to the fact that all data about a user is real.

All ratings will be anonymous to users in order to try to make people more honest and open when giving feedback about someone. Even though rating a driver or passenger is not mandatory, and there is a chance that only a few users will use it, this system can help people decide whom to trust and share trips with. Another problem is that users could give high rating to friends or even acquaintances and low rating to people that they dislike. Nevertheless, in the long term the ratings could stabilize and become something very valuable.

The main goal of this system is also to create and encourage good and trustworthy behavior and help people decide, based on the history of past interactions, people with whom they would like to share rides with in the future. Besides that, the rating provides the administrator(s) of the system the opportunity to find disrespectful behaviors and take actions, for example, banning users from the carpooling system.

4.5 FEUPooling Web

FEUPooling Web was developed to be used only by the responsible(s) for the system, which means that it is not intended to be accessed by the carpooling users.

4.5.1 Requirements

Regarding this system, intended to be used by administrator(s), there are two types of users: Authenticated and Unauthenticated.

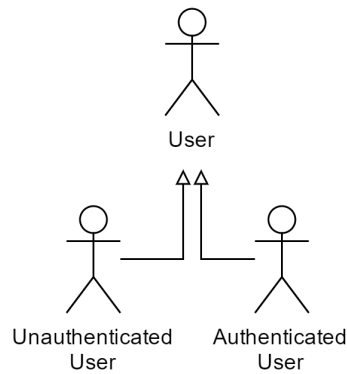


Figure 4.3: Users *UML* for *FEUPooling Web*

Identifier	Description
Authenticated User	Authenticated generic user. Has permissions to execute all administrative operations on the system.
Unauthenticated User	Generic user. Can only authenticate in the system, given it has permissions to it.

Table 4.1: *FEUPooling Web* Actors Description

For this system, the defined user stories are presented on the following tables.

Identifier	Name	Priority	Description	Interface
US101	Login	High	As an unauthenticated user, I want to login to the application, so that I can start using it.	B.1

Table 4.2: *FEUPooling Web* User Stories for Unauthenticated User

FEUPooling: Carpooling Platform

Identifier	Name	Priority	Description	Interface
US201	List Users	High	As an authenticated user, I want to list all the users that use the application, so that I can check which users of FEUP are using the system.	B.2
US202	Ban/Unban user	High	As an authenticated user, I want to be able to ban or unban users, so that I can prevent abusive uses of the application.	B.2
US203	List Trips	High	As an authenticated user, I want to list all the trips that have been created, so that I can check how the system is being used.	B.3
US204	View Trip Details	High	As an authenticated user, I want to see all details about a trip and passengers involved, so that I can know more details about it.	B.4, B.5
US205	Change points to be received per trip	High	As an authenticated user, I want to change the points that drivers and passengers receive for a valid trip, so that they can be more justly rewarded.	B.6
US206	Filter and order the list of users	Medium	As an authenticated user, I want to filter and order the users, so that I can find the users I am looking for.	B.2
US207	View User Profile	Medium	As an authenticated user, I want to see the profile of an user, so that I can get more information about that user and the trips performed.	B.7, B.8, B.9
US208	Add and Remove Points	Medium	As an authenticated user, I want to add or remove points to an specific user, so that I can perform fixes, reward users or exchange the user's points for rewards.	B.7
US209	Filter and order the list of trips	Medium	As an authenticated user, I want to filter and order the trips, so that I can find the trips I am looking for.	B.3
US210	Logout	Medium	As an authenticated user, I want to logout of the system, so that no unauthorized accesses are performed.	Any but login's

Table 4.3: FEUPooling Web User Stories for an Authenticated User

FEUPooling: Carpooling Platform

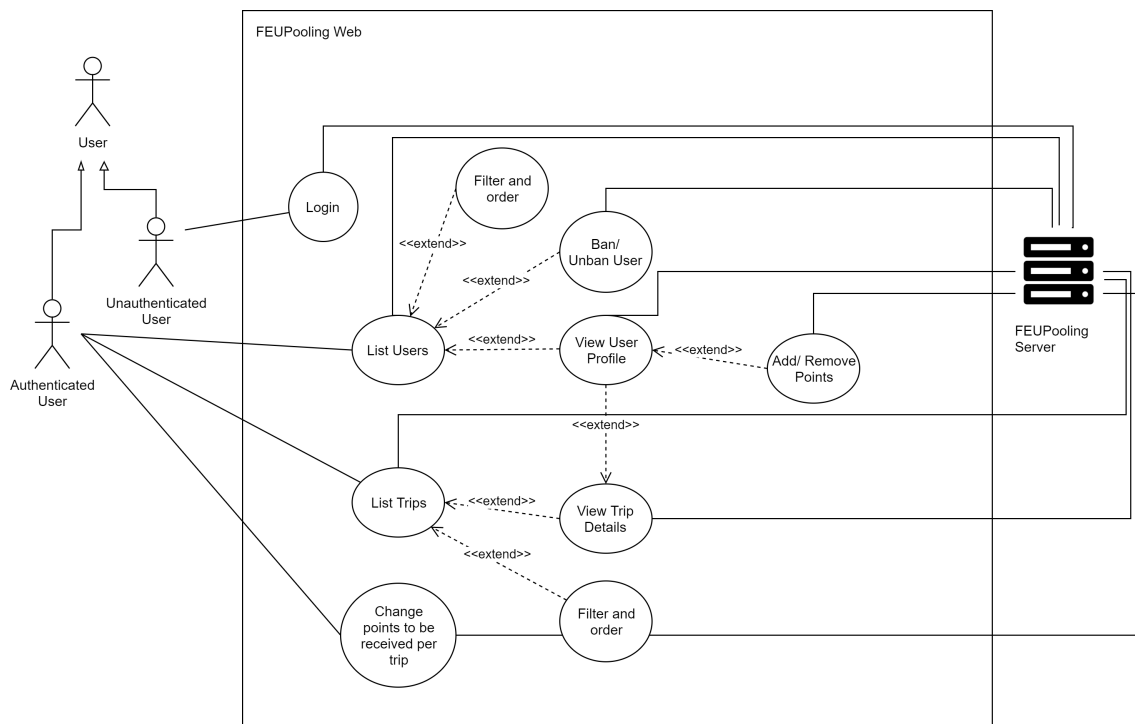


Figure 4.4: UML Use Cases for *FEUPooling Web*

4.5.2 Features

As for the way the application can be used, when the administrator accesses the website, a valid username and password is required in order to be able to perform any functions (B.1). After that, the authenticated user can choose one of the options from the navigation bar: “Users”, “Trips”, “Settings” and “Logout”.

The first option, “Users” (B.2), shows a list of users that already have already used the mobile system. The data displayed can be ordered and filtered by name, email, student number and number of points collected. These options are important, as the administrator may want to search for something specific, or just analyze the available data. There is also the possibility of accessing an individual user’s profile by clicking a button named “Details”. Besides that, there is also the possibility to ban or unban a user, in order to prevent abusive uses of the system.

The user’s individual page (B.7, B.8, B.9) shows all the relevant information about a user on the system. The following details are displayed: name, email, student number, institucional profile web page, mobile number (if provided), points gathered and also driver and passenger rating given by other users that have shared trips with the user. Furthermore, the history of trips as a driver and a passenger and the respective evaluation (a rating from 0 to 5 and who gave it) can also be seen. Finally, on this page there is also a points history, by the means of a list, containing a record of when and how (by being a driver or a passenger) the user received points or exchanged them. If the

FEUPooling: Carpooling Platform

user intends to exchange the accumulated points for rewards, the administrator can perform that action on this page by specifying a description and the quantity of points to be added/removed.

The screenshot displays the user interface for Bruno Gonçalves. It includes a profile summary, a 'Points History' table, an 'Add/Exchange Points' form, a 'Trips as a driver' table, and an 'Evaluation Received' table.

Date	Description	Flow	Total
2017-06-16 11:59:35	Received points for being the driver on a valid trip	+30	153
2017-06-13 17:17:18	Received points for being the driver on a valid trip	+30	123
2017-06-13 15:35:53	Received points for being the driver on a valid trip	+30	93

Date	Departure Place	Arrival Place	Passengers
2017-06-19 09:05	R. Azemha	FEUP	1
2017-06-18 12:35	FEUP	Rio Tinto	1
2017-06-16 09:05	R. Azemha	FEUP	1
2017-06-15 13:15	FEUP	Rio Tinto	1
2017-06-15 14:27	R. Amélia Rodrigues	FEUP	1
2017-06-12 09:05	R. Azemha	FEUP	1
2017-06-10 01:41	Poto	FEUP	0
2017-06-09 09:05	R. Azemha	FEUP	1
2017-06-09 00:00	FEUP	Rio Tinto	1

Date	Name	Student Number	Rating
2017-06-13 13:53:07	Eduardo Almeida	201204989	0 out of 5
2017-06-12 09:07:40	Eduardo Almeida	201204989	1 out of 5

Figure 4.5: User page - FEUPooling Web

Regarding the “Trips” (B.3) option on the navigation bar, it shows a list of all the trips stored on the system. The data displayed can also be ordered and filtered by date, departure and arrival place, driver and number of passengers. Each trip also has an individual page. It shows all the information mentioned before and also the departure time and estimated arrival time and a list of users involved on that trip, whether they are passengers or the driver. A map is also shown, with the departure and arrival place and an estimated route calculated by the system.

The screenshot displays the trip details for a trip on 2017-06-19. It includes a map showing the route from A to B and a list of passengers.

Name	Role
Bruno Gonçalves	Driver
Eduardo Almeida	Passenger

Figure 4.6: Trip page - FEUPooling Web

The following option, “Settings” (B.6), allows the administrator to set the quantity of points to be received by each user for a valid trip, both as a driver and as passenger.

Finally, the last option (“Logout”) allows the administrator to terminate his session with the system.

4.5.3 Implementation

The framework chosen for implementing the application was *Angular 2*. This a relatively recent and popular *JavaScript*-based web framework. It is an improvement from the previous version and is now fully component oriented. It is a framework for developing robust web applications. Furthermore it implements the concept of Two-Way Data Binding, which means that, when proprieties in the model get updated so does the view and vice versa, following the *MVC* software architectural pattern [Per14b].

The recommended language to develop applications that target this framework is *TypeScript*, an open-source programming language created by *Microsoft*, that is a superset of *JavaScript* that compiles to this same language [Esc16].

This framework was considered since the already developed *API* could be used without the need to create duplicated code, as no backend code would need to be developed. The *API* requests could then be almost the same as the mobile client application. Nevertheless, the means of authentication is not the same as the mobile client. A federated authentication is not used, and a database of administrative users is used in its place. If the authentication is successful, the *API* answers with a token that is stored to be sent later in all future calls to the *API*.

In order to aid the implementation of some functionalities, apart from everything that is available on *Angular* itself, a few third-party libraries, which are reusable software componentes, were used. These libraries were:

- *ngx-bootstrap* - Eases the usage of all core *Bootstrap* components in applications powered by *Angular*. *Bootstrap* allows for an easier development of responsive websites, which make them look good across all sorts of devices. Besides the most common elements, “Tabs”, “Modal Windows” and “Alerts” were used. “Tabs” are used to separate information on the user profile page and the trip page. “Modal Windows” are used as a confirmation pop-up, for example, to confirm if the administrator wants to add or remove points from a user or ban/unban a user. Finally, “Alerts” are used to show the result of actions like the ones mentioned before. These message alerts appear for less than five seconds and then disappear automatically [Sof17].
- *angular2-datable* - A table component with sorting and pagination. On the website it is used in all lists of users, trips, history, among others [Fol17].
- Google Maps APIs - Used to represent a map with the departure and arrival location of a trip. Its *Directions API* is also used in order to represent the calculated route between these two points [Sto17e].

4.6 FEUPooling Mobile

FEUPooling Mobile is the core subsystem of the project. It is safe to say that most if not all of *FEUP*'s students carry a smartphone with them all the time. This mobile application is intended to be used by the students that wish to plan trips and share transportation with other colleagues. In order to target the most students, the deployment operating systems are *Android* an *iOS* because they were the two most popular mobile operating systems at the time of development [Vin17].

4.6.1 Requirements

Regarding this subsystem, intended to be used by the *FEUP* student community, two types of users can be identified: Authenticated and Unauthenticated. The last one only can login on the system because only authenticated users can use the application. Regarding the Authenticated user, it can also be divided into two subcategories: Driver and Passenger. The user can be both driver and passenger without needing to change any setting. The role played by the user changes accordingly with the goal when using the application at a certain moment. When a user opens the mobile application in order to find a ride, this user is at that moment considered a Passenger. On the other hand, if the user opens the application with the intent of creating a trip advertisement in order to try to find someone to share a commute with, the user is considered a Driver.

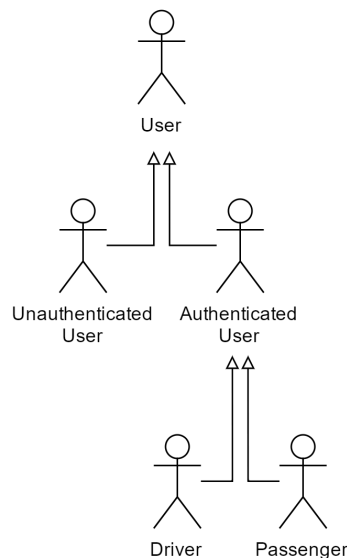


Figure 4.7: Users UML for *FEUPooling Mobile*

FEUPooling: Carpooling Platform

Identifier	Description
Passenger	Authenticated user that is a passenger in any trip or is looking forward to it. The main goal of this user is to find advertisements created by other drivers in order to share rides with them.
Driver	Authenticated generic user that is a driver and offers to share trips with other passengers. This user has a private vehicle and is interested in sharing daily commutes.
Authenticated User	Authenticated generic user. Has generic permissions. It can change into a Passenger or Driver.
Unauthenticated User	Generic user. Can only authenticate in the system if has permissions to do it.

Table 4.4: *FEUPooling Mobile* Actors Description

The user stories for this system are presented in the following tables.

Identifier	Name	Priority	Description	Interface
US301	Login	High	As an unauthenticated user, I want to login to the application, so that I can start using it.	B.12

Table 4.5: *FEUPooling Mobile* User Stories for Unauthenticated User

FEUPooling: Carpooling Platform

Identifier	Name	Priority	Description	Interface
US401	See Profile	High	As an authenticated user, I want to see a profile, so that I can find information about a user.	B.18
US402	Edit Profile	High	As an authenticated user, I want to edit my profile, so that I can keep my mobile number and preferences updated.	B.19
US403	See Trip Page	High	As an authenticated user, I want to see a trip page, so that I can find more information about it.	B.23
US404	List notifications	High	As an authenticated user, I want to list my notifications, so that I can acquire information that may be relevant to me.	B.22
US405	List Trips as Driver	Medium	I want to list the trips a user has performed as driver so that I can know more about the trips history of that user.	B.20
US406	List Trips as Passenger	Medium	I want to list the trips a user has performed as passenger so that I can know more about trips history of that user.	B.21
US407	Delete notifications	Medium	As an authenticated user, I want to delete my notifications, so that I can delete notifications that are not important to me anymore.	B.22

Table 4.6: *FEUPooling Mobile* User Stories for Authenticated User

FEUPooling: Carpooling Platform

Identifier	Name	Priority	Description	Interface
US501	Create Advertisement	High	As a driver, I want to create a new advertisement, so that I can inform other passengers about a trip that I am doing and would like to share.	B.15, B.16
US502	Delete Trip	High	As a driver, I want to delete a trip that I have created, so that no one may attempt to join it.	B.31, B.32
US503	List Requests	High	As a driver, I want to list other users' requests to join a trip that I am a driver, so that I can view which users are interested in sharing a ride with me.	B.27
US504	Accept/Reject Requests	High	As a driver, I want to accept or reject other users' requests to join a trip that I am a driver, so that I can share a ride with who I desire.	B.28
US505	Remove a passenger	High	As a driver, I want to remove a passenger on a trip that I am a driver, so that I no longer need to share a ride with that user.	B.30
US506	See passenger position	High	As a driver, I want to see the actual position of a passenger during a trip, so that I can know where I need to stop next.	B.35
US507	Start a trip	High	As a driver, I want to start a trip, so that I can inform the other passengers on the trip that it has started.	B.33
US508	Stop a trip	High	As a driver, I want to stop a trip, so that the system is aware of this fact and may award me and the passengers with points.	B.35
US509	Edit Trip Page	Medium	As a driver, I want to edit information about a trip that I have created, so that I can keep the information updated.	B.31, B.32
US510	Give feedback	Medium	As a driver, I want to give feedback about other passengers that shared a trip with me, so that I can help create trust in the system.	B.39

Table 4.7: FEUPooling Mobile User Stories for Driver Authenticated User

FEUPooling: Carpooling Platform

Identifier	Name	Priority	Description	Interface
US601	Search for trips	High	As a passenger, I want to find trips by date or place, so that I can see if there is any trip I may be interested in joining.	B.13 , B.14
US602	Request to join a trip	High	As a passenger, I want to request to join a trip I am interested in, so that I can share that commute.	B.24
US603	Cancel request	High	As a passenger, I want to cancel a request that I have made to join a trip, so that the trip driver does not think I am interested in joining that trip.	B.26
US604	Remove from a trip	High	As a passenger, I want to remove myself from a trip that I am already a passenger, so that I can show that I am no longer interested in participating on it.	B.30
US605	See driver position	High	As a passenger, I want to see the actual position of the driver during a trip, so that I can know the driver's location.	B.35
US606	Give feedback	Medium	As a passenger, I want to give feedback about the driver and other passengers that shared a trip with me, so that I can help create trust in the system.	B.39

Table 4.8: *FEUPooling Mobile* User Stories for Passenger Authenticated User

FEUPooling: Carpooling Platform

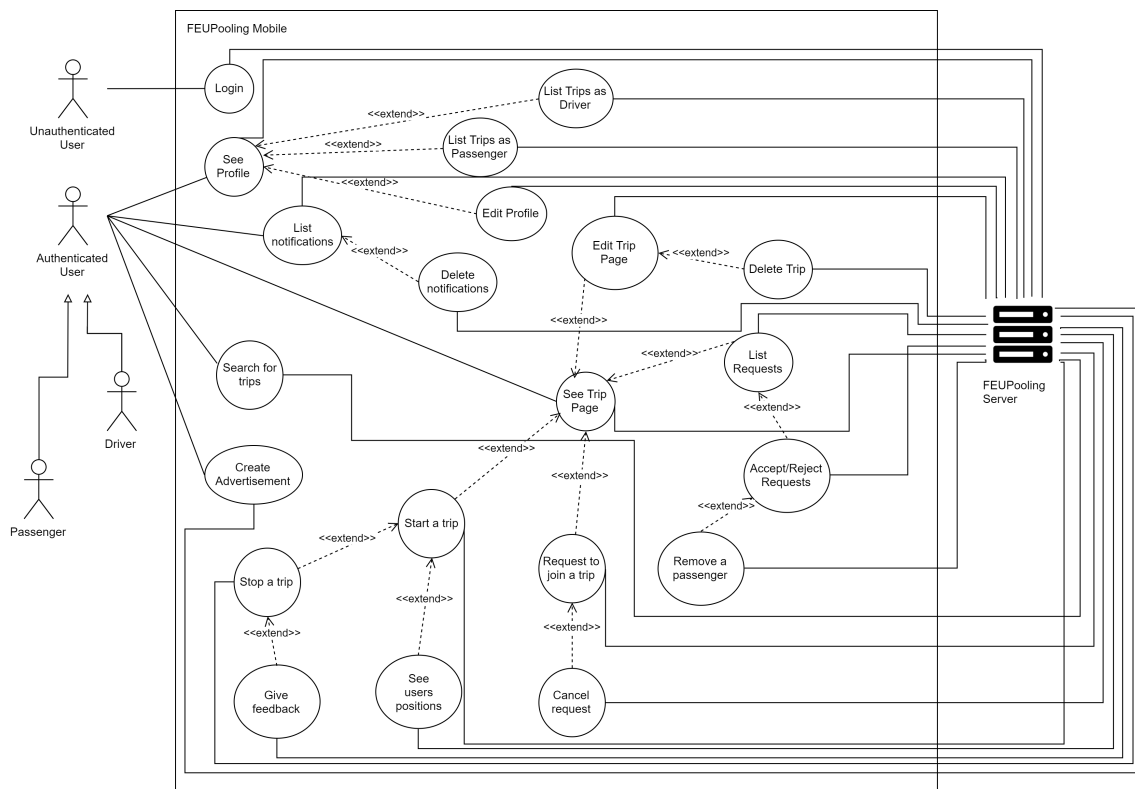


Figure 4.8: UML Use Cases for FEUPooling Mobile

4.6.2 Features

Regarding the functionalities of the system, it is intended for it to be very simple to understand and use in order to have the maximum possible adoption. The first step when the user installs and opens the application is to perform a federated authentication with credentials from *Universidade do Porto* (B.12). This is a one-time step, that is only requested the first time the user opens the application. An internet connection is always required, as data needs to be constantly obtained from the server.

The main screen (B.13) shows a map with the actual user position on it (as long as it is available) and the trip advertisements available for the selected date, which for default is the present day. Upon tapping on the icon of a trip on the map, an information window is shown with the departure date and time and the type, that is, if the trip will start or end in *FEUP*. The route from the departure to the arrival point is also drawn on the map. By tapping that window, a page with more details about the selected trip is displayed. Also on the map screen, it is possible to filter the trips currently being shown on the map by date, by choosing the intended date on the calendar. This page was chosen to be the main screen because it was assumed that most of the users would open the application to search for a ride and, on that case, a “draggable” and “zoomable” map would be the easiest way to

FEUPooling: Carpooling Platform

find a trip corresponding to the needs of the user. From this screen, on the tab bar, it is possible to navigate to the following views: “Map”, “List”, “Create”, “Profile” and Notifications.

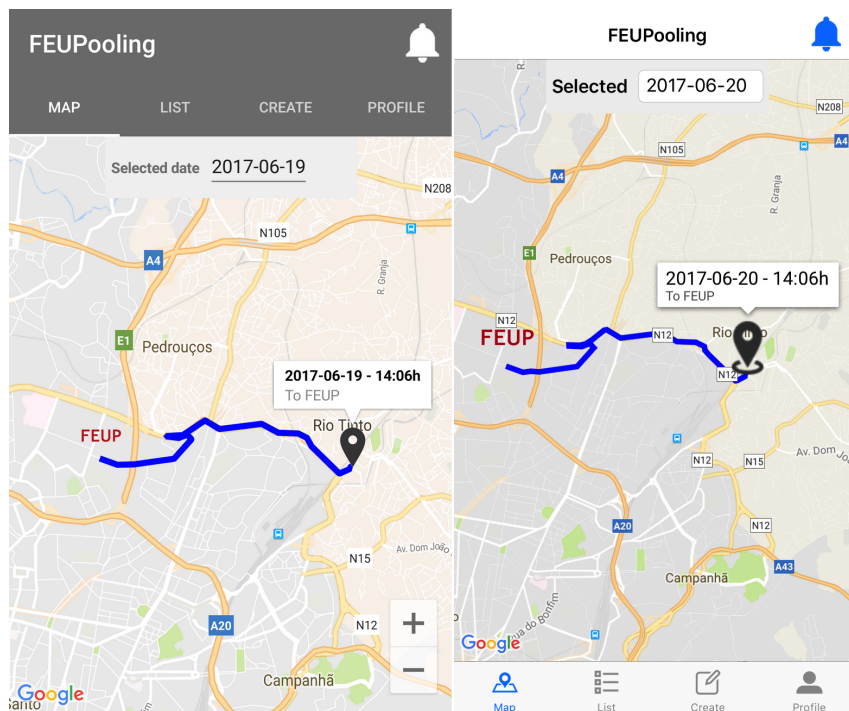


Figure 4.9: Main screen map - *FEUPooling* Mobile

By switching to the option “List” (B.14), a list of the same filtered trips by date is shown, ordered by departure time, with the departure and arrival place and number of available seats at that moment. If the trip has any available seats at that moment, that row will be highlighted in another color. Upon tapping on an item, a page with more information about a trip is also shown.

The following option, “Create” (B.15, B.16), is directed to drivers who pretend to announce a trip they are doing in order to find interested passengers. There are four types of information that need to be provided by the user, which are the date and time of the departure, places, information about the vehicle and possible additional information. The trips are required to start or end in *FEUP* to ensure that users only use it when traveling to college, therefore in the places section the driver must choose the trip type (from or to *FEUP*). The other destination (departure or arrival) must then be selected in a new view (B.17). On that view, there is a map and the possibility to search and select a desired place. Regarding the vehicle, the driver must designate its vehicle, along with the number of free seats, which is the number of passengers the user is willing to take on the trip (apart from the driver). Finally, additional information is the last field and is not mandatory. This field is useful in the cases the driver wants to provide extra information that can not be fulfilled by the other fields, for example if the trip may be delayed due to the schedule of a given class. When all the information is given the user can attempt to create the trip, and in case of success a redirection to the trip page will occur.

FEUPooling: Carpooling Platform

“Profile” (B.18) is the final option on the tab bar and it displays all user information. Besides a picture, it includes the name, student identifier, phone number, email and number of points available. Apart from that, it shows three basic preferences that are: smoking, talking and listening to music during trips. Additionally, it shows the rating the user has been given by other passengers, from 0 to 5, and it is possible to see the trips that have been done both as a driver (B.20) and as a passenger (B.21). Finally, if the user profile corresponds to the user that is logged in, there is the option to edit some information of this page. Editing profile (B.19) allows the user to insert or update a mobile number for other users to contact when interested on a trip and to update their basic preferences.

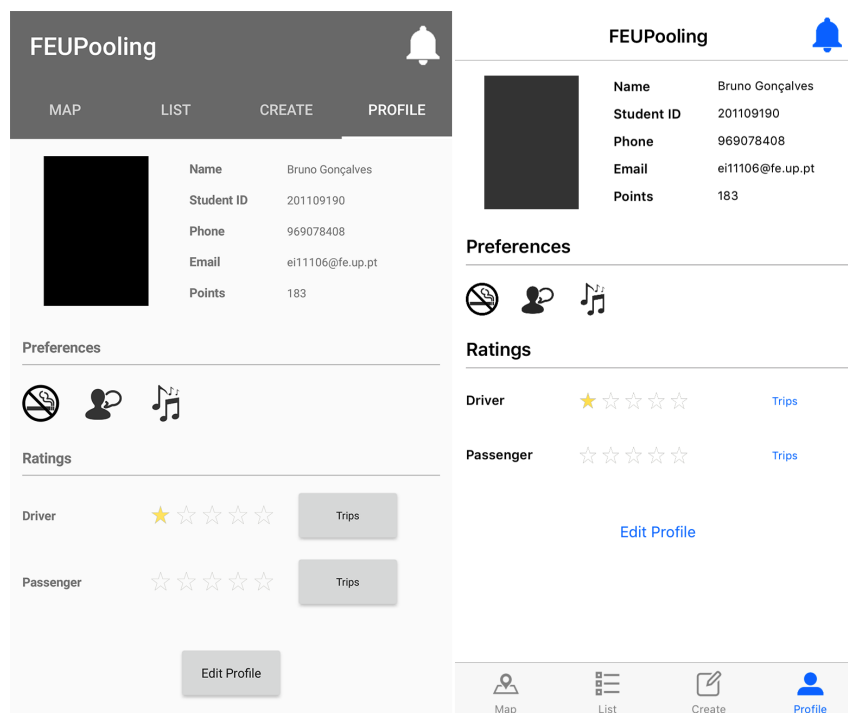


Figure 4.10: User profile - *FEUPooling* Mobile

On the upper right corner of the screen there is an option that allows the user to open a notification page (B.22). By tapping on it, it shows a list of messages directed to the user and the respective date and time they were sent. These messages can be requests to join a trip, a driver editing a trip that the user is part of, a trip that the user is involved has started, an exchange of points, among others. The user can also delete notifications individually or remove them all. When a notification is related to a trip, by tapping on it the trip information page is shown.

Regarding the trip page (B.23), all information about it can be found there — date, departure time and place, estimated arrival time and place, car, number of free seats and description. Besides that, there is a section for passengers, that shows a list of users with the picture, name, and role (driver or passenger) of each one of them. If there are still seats available and the user is not yet part of the trip, the option to request a trip will appear (B.24). This request can be canceled anytime before the driver accepts or rejects it (B.26). If the driver accepts it, the requester becomes

a passenger. The passenger can choose not to be part of the trip anymore or be excluded by the driver (B.29, B.30). If the logged in user is the driver of the trip, a list of pending requests waiting for approval will be shown, along with an option to edit the information of the trip (B.27). This information can then be updated and the trip can even be deleted if the user wishes so (B.31, B.32). If the date is equal or greater than the departure date and time and the number of passengers is not zero, the driver has an option to start the trip (B.33). During a trip, a new view appears when a user tries to access a trip page. This view (B.36) shows a map with the departure and arrival points, driver and passengers positions, along with a route starting at the position of the driver, going through passengers' positions and ending on the arrival point. The driver is the only one who can end the trip (B.35). When the trip ends, the driver and passengers will have the option to give feedback about each other, using a star rating system from zero to five (B.38, B.39).

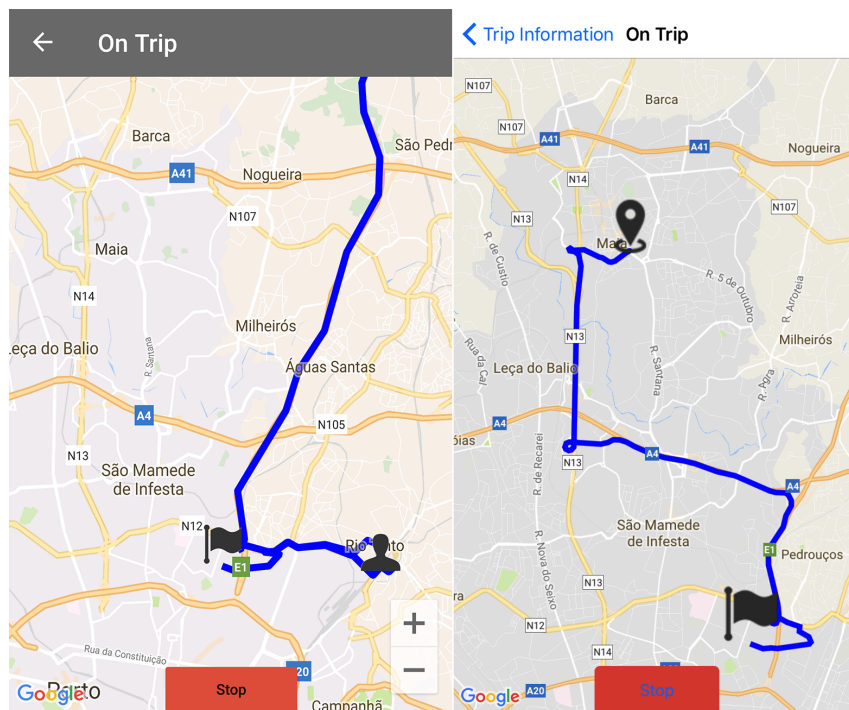


Figure 4.11: On trip mode for the driver - FEUPooling Mobile

4.6.3 Implementation

This application is intended to be used by all interested students. Therefore, in order to target a wide range of students, the application needed to be designed for the two major mobile operating systems: *Android* and *iOS*. In the last quarter of 2016, 99.6 percent of all new smartphones sold ran *Android* or *iOS* [Vin17]. As of that, a cross-platform approach was chosen because there would not be the need to implement much platform-specific code.

Xamarin.Forms was the framework chosen for this project because it allows the development of native apps for *Android*, *iOS* and *Windows Phone* by sharing a single *C#* codebase, without

the need of much platform-specific code. *Windows Phone* was not considered as a target platform because it has a smaller representation [IDC16] on the market and most of the libraries used were not compatible with this operating system.

When the application is opened, a splash screen (B.10), which is a loading screen presented while the application is launching, is shown to the user. For both *Android* and *iOS*, different code had to be written for each platform but the visual content is the same.

The mobile application requires an internet connection (B.11) in order to get data from the server. The only data that is stored locally is the user identifier and a token, which is received after the first login and is used in all future *API* requests with the server.

That being said, after checking for an internet connection, the application verifies if the user has previously logged in and if the stored token is valid. If the answer is no, the user will be presented with a web view in order to perform the federated authentication (B.12). The web view was implemented by a custom renderer for both *Android* and *iOS* because by default, web views on *Xamarin.Forms* are not “zoomable”. The customization, appearance and behaviors of *Xamarin.Forms* default controls can be customized by custom renderers which are effectively native, platform-specific controls. When a valid login is performed, a token and user identifier are received and saved. The application then shows the user its main view.

Although is not necessary to have the device’s *GPS* active in order to use some functionalities, a listener was implemented that acquires the user’s position when necessary. If a driver wants to start a trip, the *GPS* is required to be active, otherwise the action will not be allowed (B.34).

It is important to keep all the information updated, therefore when a view appears, it updates its information by exchanging data with the server. Besides that, almost every view has pull-to-refresh functionality which allows for a manual update of its data.

On the following paragraphs some aspects considered relevant regarding the implementation of some views will be mentioned.

On the trips map view (B.13), the minimum search date is always the present date. When a new date is selected, both the map and list view information are updated. When no trips are found, a message is shown to the user. When a route icon is tapped, a route is drawn by performing a *REST* call to *Google Maps API*.

In order to ensure that the application is only used to commuting from or to *FEUP*, when creating a trip (B.15, B.16) a user can only freely choose one of departure or arrival places, being the other place always *FEUP*. By using *Geocoder*, which converts a street address to geographic coordinates, the user can search for a place or street by name (B.17). Also for creating a trip there was the need to use a custom renderer, since the default time picker only allows for the date to be displayed in AM/PM, and not in 24-hour format. The numbers of seats to be offered by the driver on a trip is limited from one to three because at least one seat needs to be offered (else, the ride

would not be shared) and a maximum of passengers a normal car can carry are three, apart from the driver. The optional additional information was added so that the driver could add some extra information or an alternate way to contact the driver if desired.

In the user profile (B.18), it is possible to see a user's rating as a driver and passenger as given by other users, although it is not possible to know who gave what, as if this information was public there was a chance the rating system would be corrupted. If a user knows that the evaluation will be anonymous, the rating will most certainly be more honest. Nevertheless, as a user can not search for trips that already have been performed or are older than the present date, it is possible for these to be accessed by a given user's page in order to allow others users to view the history of that user. Another important aspect implemented are basic preferences. This is a core functionality present on some of the carpooling platforms already on the market. It is easier to cluster users with common interests, for example, if a user is allergic to smoke or does not like it all, it would not make much sense to share a trip with another one that smokes during journeys.

Regarding the trip page (B.23), some information is only shown to some users, that is, only the driver can check pending requests but can not request a seat, for example. When the driver starts a trip and the other passengers open the trip page, they will be redirected to the on trip mode page (B.36). On this page, which can only be seen by users involved on the trip, there is a timer that connects to the server in order to acquire the other user positions. The route on the map is then redrawn starting on the driver position, passing through all passengers positions and ending on the arrival place. When the user position changes, its current position is sent to the server. If the driver is further away than 1km from the arrival place, and taps the button to stop the trip, it will be warned that if does so, no user will receive points because the validation process will not consider it valid (B.37).

In order to aid the implementation of some functionalities, a few packages and functions written by other developers were used:

- Xamarin.Forms.GoogleMaps - It is an optimized library for *Google Maps* and provides many functionalities used on the project, as described: Map types, Map events, Custom Pins, Pin drag & drop, Lines, Custom map tiles, among others. The pins are customizable and can have different images. They also have a label field (main title), an address field (a subtitle) and a geocoordinate associated. This information is shown when the user taps a pin which opens a information window. By tapping on this window, an event that opens the trip page is triggered. There are two events being used: the information window clicked and the pin clicked. When tapping on a map a pin can be placed. This functionality is used when creating or editing a trip to select the departure or arrival place. It is also possible to move the map to the user's position when it changes, which is used for the real-time tracking of ongoing trips. In order to represent routes a method of drawing polylines is provided, which are lines composed by several segments. Some parameters can also be customized, of which examples are the stroke color and width [ama17].

FEUPooling: Carpooling Platform

- Geolocator Plugin - Plugin used to get the user's *GPS* position [Mon17a].
- Xamarin.Forms-Samples - Code samples with behaviors. A sample was adapted and was used on the rating star system at the end of a trip [Nis17].
- Settings Plugin - A plugin to create and access settings from shared code in a fast and reliable way. On this project, it is used after the first valid user login performed by the user in order to save the user identifier and token to be used on future *API* calls with the server [Mon17c].
- RestSharp - A *REST* Client. It is used for all communications with the server. The responses can be automatically deserialized from *JSON* to an instance of a class [res17].
- ACR User Dialogs - A library that allows for more customized user dialogs. In this case, it is used to show error messages that are presented when attempting to connect to the internet, to get a *GPS* signal, among others [Rit17].
- Xamarin.Forms-PullToRefreshLayout - Library whose goal is to implement a pull to refresh layout. On the project it was used so that the user can easily request for an update of the data on the views [Mon17b].

Chapter 5

Validation

In this chapter, usability, performance and energy and trip validation tests that were performed are mentioned. These tests were made in order to try to understand if the desired goals were achieved. Nevertheless, only a few, small tests were conducted. In order to test the robustness of the product there is the need to test with a larger scale of users, like mentioned below on future work.

5.1 Usability

As stated before, the mobile application aimed to be very simple to use in order to have the greatest possible adoption by the students. Therefore, when presented with the main screen, the user should be able to perform most actions with the minimum taps possible. This makes the application efficient, since most actions can be performed quickly. Besides that, it also makes it easier to learn and remember how to use the application. If a user has to perform many unnecessary taps before an action can be done, there is a chance the user may become frustrated and end up not using the application.

That being said, on the table below most actions and the minimum number of taps required to achieve them from the main screen are represented. Some actions can be achieved from different paths but the number of taps is almost the same. For the counting of taps, variable input lengths were not considered. For example, when creating a trip, a tap was considered when selecting an input field, but not the taps that on the keyboard that appears when the input field is selected. All the remaining taps were included though, including, but not limited to confirmations.

Validation

Action	Number of taps
Login	3
See Profile	1
Edit Profile	2
See Trip Page	2
See Trip Route	1
List notifications	1
List Trips as Driver	2
List Trips as Passenger	2
Delete notifications	2
Create Advertisement	11
Delete Trip	5
List Requests	2
Accept/Reject Requests	4
Remove a passenger	5
Start a trip	4
Stop a trip	4
Edit Trip Page	4
Give feedback	6
Request to join a trip	4
Cancel request	5
Remove from a trip	5

Table 5.1: Actions and number of taps required

The medium number of taps for each action is 3.6. This means that, from the main screen, a user can perform any action by tapping the screen, in median, 3.6 times. It is also important to state that the creation of an advertisement requires a great number of taps because there are many fields that need to be provided by the user.

5.2 Performance and Energy

One of the main functionalities of the mobile application is the ongoing trip mode. This mode provides real-time tracking of all users in a given trip. This continuously uses the *GPS* of the

Validation

device and therefore it was important to try to minimize the battery spent. Mobile phones batteries usually have a not so long time of usage before they need to be recharged again. If this application drained the battery (consumed energy) too fast, users might end up not using it. That being said, performance tests were conducted in order to verify the battery consumption.

It was decided to perform test trips using different mobile phones. The smartphones used were an *LG G4* running *Android 6*, an *OnePlus One* running *Android 7* and an *iPhone 7 Plus* running *iOS 11*. In one way the trip was performed using *FEUPooling* and on the reverse way using *Google Maps*. As *Google Maps* provides navigation it was used as a benchmark to compare the battery spent on both applications.

The trip's distance was about 26km and the duration was more than half an hour.

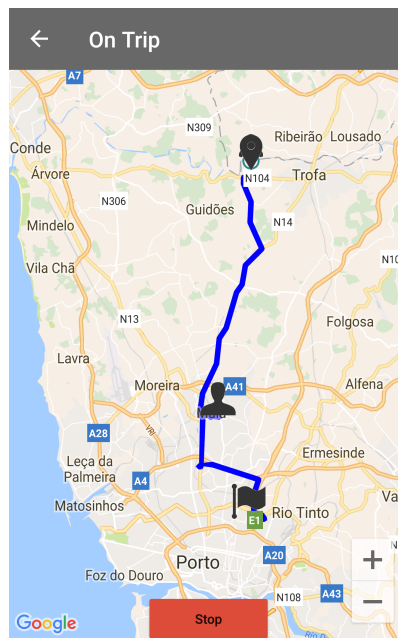


Figure 5.1: Test trip 1

The battery consumption on each smartphone is presented on the table below.

	<i>FEUPooling</i>	<i>Google Maps</i>
<i>LG G4</i>	3%	5%
<i>OnePlus One</i>	2%	5%
<i>iPhone 7</i>	5%	6%

Table 5.2: Battery consumption comparison

Validation

Taking into account the values obtained, it can be concluded that the battery spent with *FEUPooling* is similar to *Google Maps*'.

Besides that, the data consumption on each smartphone was also verified. The results are presented on the table below.

	<i>FEUPooling</i>
<i>LG G4</i>	4MB
<i>OnePlus One</i>	5MB
<i>iPhone 7</i>	7MB

Table 5.3: Data consumption comparison

The data used was also within accepted values taking into account that the maps needed to be downloaded and that the application is constantly communicating with the server.

5.3 Trip Validation

In order to test the validation of the trips, a trip was performed, with two users, on both ways. On the first scenario it was expected for the trip to be valid, as both users had the application open the whole time, and on the second scenario this was not expected to happen since one of the users was supposed to close the application early on.

The raw data obtained from the validation Scenario 1 was:

user_id	trip_id	valid	created_at
9	79	True	2017-06-12 12:16:01
9	79	True	2017-06-12 12:17:01
9	79	True	2017-06-12 12:18:02
9	79	True	2017-06-12 12:19:01
9	79	True	2017-06-12 12:20:01
9	79	True	2017-06-12 12:21:02
9	79	True	2017-06-12 12:22:02
9	79	True	2017-06-12 12:23:01

Table 5.4: Validation Scenario 1

Validation

Regarding Scenario 2 it was:

user_id	trip_id	valid	created_at
9	80	True	2017-06-12 13:10:02
9	80	True	2017-06-12 13:11:01
9	80	True	2017-06-12 13:12:02
9	80	True	2017-06-12 13:13:01
9	80	True	2017-06-12 13:14:01
9	80	True	2017-06-12 13:15:02
9	80	True	2017-06-12 13:16:01
9	80	False	2017-06-12 13:17:02
9	80	False	2017-06-12 13:18:01
9	80	False	2017-06-12 13:19:01
9	80	False	2017-06-12 13:20:01
9	80	False	2017-06-12 13:21:02

Table 5.5: Validation Scenario 2

As expected the first trip was considered valid because the passenger's route matched 100% with the drivers' and the second scenario was not because it only matched 58%. Therefore, only the first scenario awarded points to the users since the driver's route needs to match with any passenger at least 90%.

Validation

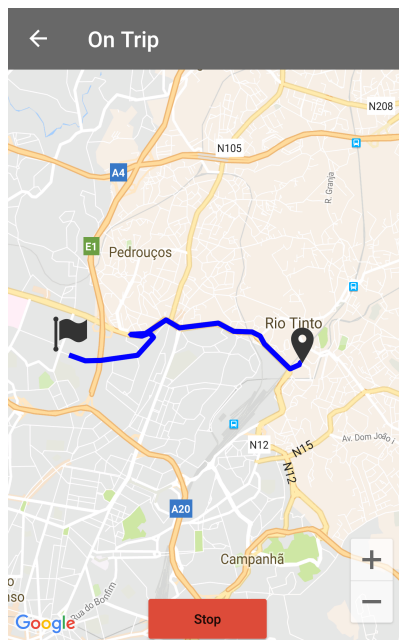


Figure 5.2: Test trip 2

Chapter 6

Conclusions

The demand for transportation, particularly on private cars, has increased, causing serious problems at an environmental, social and economic level, which need to be solved or at the very least minimized.

Increasingly, it seems untenable to maintain the current individual levels of usage of transportation, especially within major cities. It then becomes fundamental to:

“Be part of the solution and not part of the problem” [Cam02]

It is important to invest on alternative transport solutions, namely vehicle sharing, so that the vacancy rate of the vehicles in circulation will be increased.

6.1 Contributions

Taking into account the current state of the art with regards to the mobility platforms, it can be concluded that carpooling is a current and relatively popular topic. Due to specific requirements, such as federated authentication and a reward system that other platforms do not offer, there was a need to develop a product combining some of the functionalities already implemented in other platforms, but also new ones specific just to this case.

Therefore, a mobile solution (*Android* and *iOS*) was developed for planning the sharing of daily commutes of the students of *FEUP*, a web client application for management of the system and a backend.

Some of the more important implemented aspects on the mobile application are:

- Authentication with the information services of the faculty, which makes the service more secure since the person with whom transportation will be shared can be known and verified;

Conclusions

- Possibilities to advertise trips that the user would be interested in sharing;
- Definition of some basic preferences regarding with whom the user would want to share transport with;
- Rewards system as a form of incentive for users to join and use the platform.
- Reputation system in order to encourage a good behavior among students.

The developed platform can bring numerous benefits, as previously mentioned, to the *FEUP*'s community. This platform was intended to have innovative aspects so that, if it is of interest, it may be adopted by other educational institutions so as to increase the likelihood of finding people with similar paths and interested in sharing transport.

Finally, it can be concluded that the main goal of the dissertation, which was to develop a product to fulfill the specific needs of a institutional carpooling platform, was attained.

6.2 Future Work

Although the applications are ready to start being used on the following school year, on a first phase the whole platform should be tested in a larger scale and in the form of a prototype, by the *FEUP* student community. Some small tests have been performed on the platform and it was deemed functional but there is still room for improvements. Besides that, as what happens with the majority of products, it is probable that some fixes may be required after a larger scale testing.

There are some steps that must be defined by the stakeholders of the system. After the system leaves the pre-production phase and is migrated to an institutional server, some adjustments to the authentication system should be made and new required *API* keys to use *Google APIs* should be generated, and there should be a more detailed testing phase.

A medium sized group of students willing to use and test the system should be found. Some kind of reward could be offered, perhaps a number points on the system, in order to have a good amount of volunteers. This phase is crucial in order to understand and correct some problems on the system.

After using the system for a short period of time, these students should be inquired about what they thought about the user experience, problems that occurred while using it and suggestions to improve the application. A good way to do this evaluation would be presentially because a direct interaction could be more beneficial than simply asking the testers to fill surveys.

The feedback gathered from this testing is very important, as the product should be iterated based on that feedback. Besides that, in order to have a great adoption, students must be happy with the product and the way it must be used, so all feedback should be taken into account. Results should be analyzed and iterative redesigns should be made while regarding some recommendations.

Conclusions

After the initial phase, that should not take longer than a month, and some product iteration, if needed, a launch plan for the product must be defined. Updates and maintenance must also be considered as part of the plan.

When launching the product, marketing will be crucial to the success of the application. The platform should be advertised by email and also on the main college official social networks in order to reach the maximum number of students. Applying also some mechanisms of gamification, some starter points or double points for performing trips could be offered for a limited time.

The success of the platform will only depend on the students. If only a few students end up using it, there is a smaller chance of these having common routes in order to make it possible to share trips among them. On the other hand, if the right amount of students is captivated to the system from the beginning and the platform becomes popular, the main goal can be attained.

Regarding improvements, the mobile system could benefit with the addition of interesting new functionalities. A few were considered while developing but ended up not being implemented due to lack of time and prioritization of tasks. Nevertheless, they can still be implemented on future updates. Some of them are:

- Push Notifications, which is a messaging system that delivers to the user messages from the backend to the mobile devices. With this system users could be notified, for example, when a trip they were part of started without having to open the application and check the notifications manually. This functionality started to be implemented, but due to it involving much platform-specific code and mainly due to the lack of time, it ended up not being finished.
- Background support for the on trip mode. With this functionality, users would not need to have the application open in order to have their own position sent to the server. This needs to be implemented in a different way for both *Android* and *iOS*, since both operating systems have a different way of dealing with applications on the background.
- The on trip mode could get notified when the users positions changes instead of having to poll the server for the positions of the other users. This would reduce the data exchanged with the server.
- A messaging system between users could also be implemented so that users could interact directly on the application in order to decide if they wish to share a commute.
- Ecological footprint. As one of the possible features of this system, it would be pleasant to have a record of the ecological footprint, which is the quantity of greenhouse gases that were avoided to be sent to the atmosphere by doing carpooling with colleagues.
- List of cars. If each user has a list of cars they own it would ease the creation of a trip. It would also help calculation the ecological footprint, which is influenced by the type of vehicle.

Conclusions

Finally, the web system could also be improved. Some interesting functionalities could be implemented, such as:

- Administrator(s) could see more details about performed trips. The data calculated for the trip validation could be shown, which is the percentage of route that matched between the passengers and the driver.
- Public web platform. Users could use the web version to, at least, create and plan trips. Nevertheless, the on trip mode could still be exclusive of the mobile applications.
- Leaderboards as part of the gamification system. There could be a leaderboard that would rank the users by points. That could encourage friendly competition between users.
- Rewards List. It could be a list of rewards, editable at any time by the administrator(s), and the cost for each reward could be viewed by the users.
- News List. The possibility for administrator(s) to post news about the system or promotional events that could be checked by the students would also be interesting.

In short, at the present moment of the writing of this report, there is still much to be done, however the core of the platform is finished.

Appendix A

FEUPooling Server API

In this appendix, the *API* calls for the *FEUPooling Server* are described and the documentation for each call is presented.

All the *API* calls, except the ones used to authenticate, need an *<Authorization Token>*, that is provided by the system after the authentication. Besides that, all the valid *API* calls return as a response a *success* boolean parameter that indicates the success of the call. In case of a non-successful operation, an *error* parameter will be also sent with the description of the error. Furthermore, a major part of the calls return objects with data.

A.1 Authentication

- **POST** /users/verifyToken – Verifies if a token is valid for a user.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Response** (as JSON) – { success: boolean }

A.2 User Routes

- **GET** /users/:id – Returns information about a specific user in the system.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Parameters** (as URL parameters) – id: string
 - **Response** (as JSON) – { success: boolean, error: string, user: UserObject }
- **POST** /users/profile– Updates the information of the current user.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”

- **Parameters** (as form data) – mobile_number: string, smoking: boolean, listening_to_music: boolean, talking: boolean
- **Response** (as JSON) – { success: boolean, error: string, user: UserObject }

A.2.1 Notification Routes

- **GET** /users/notifications/ – Returns all notifications belonging to a user.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Response** (as JSON) – { success: boolean, error: string, notifications: [NotificationObject] }
- **DELETE** /users/notifications/:id – Deletes a (specific) user notification.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Parameters** (as URL parameters) – id: string
 - **Response** (as JSON) – { success: boolean, error: string }
- **DELETE** /users/notifications/ – Deletes all user notifications.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Parameters** (as URL parameters) – id: string
 - **Response** (as JSON) – { success: boolean, error: string }

A.2.2 Rating Routes

- **POST** /users/:userID/ratings/trip/:tripID – Evaluate other user on a specific trip.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Response** (as JSON) – { success: boolean, error: string }
- **GET** /users/ratings/trip/:tripID/rated – Verifies if the user has rated the other passengers on the trip.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Response** (as JSON) – { success: boolean, error: string, rated: boolean }

A.3 Trip Routes

- **GET** /trips/activeTrip – Returns a “ongoing trip” identifier on which the user is a passenger or driver. If there is no trip currently ongoing for the present user, the call returns 0 as the tripID.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”

- **Response** (as JSON) – { success: boolean, error: string, tripID: int }
- **POST** /trips – Creates a new trip.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Parameters** (as form data) – departure_coords: string, arrival_coords: string, date: string, departure_time: string, description: string, car: string, num_free_seats: integer
 - **Response** (as JSON) – { success: boolean, error: string, trip: TripObject }
- **GET** /trips – Returns all trips on the system.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Response** (as JSON) – { success: boolean, error: string, trips: [TripObject] }
- **GET** /trips/:id – Returns information about a specific trip.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Parameters** (as URL parameters) – id: string
 - **Response** (as JSON) – { success: boolean, error: string, trip: TripObject }
- **POST** /trips/:id – Updates information of a trip.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Parameters** (as URL parameters) – id: string
 - **Parameters** (as form data) – departure_coords: string, arrival_coords: string, date: string, departure_time: string, description: string, car: string, num_free_seats: integer
 - **Response** (as JSON) – { success: boolean, error: string, trip: TripObject }
- **DELETE** /trips/:id – Deletes a trip.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Parameters** (as URL parameters) – id: string
 - **Response** (as JSON) – { success: boolean, error: string }
- **GET** /trips/date/:date – Returns trips occurring on a specific date, ordered by departure time.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Parameters** (as URL parameters) – date: string
 - **Response** (as JSON) – { success: boolean, error: string, trips: [TripObject] }
- **GET** /trips/users/:id – Returns all trips that a user is the driver in.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Parameters** (as URL parameters) – id: string

- **Response** (as JSON) – { success: boolean, trips: [TripObject] }
- **GET** /trips/users/:id/passenger – Returns all trips that a user is a passenger in.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Parameters** (as URL parameters) – id: string
 - **Response** (as JSON) – { success: boolean, trips: [TripObject] }
- **DELETE** /trips/:tripID/user/:userID – Remove a passenger from a trip.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Parameters** (as URL parameters) – tripID: string, userID: string
 - **Response** (as JSON) – { success: boolean, error: string }
- **POST** /trips/:id/status – Changes the status of a trip.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Parameters** (as URL parameters) – id: string
 - **Parameters** (as form data) – status: string
 - **Response** (as JSON) – { success: boolean, error: string }
- **POST** /trips/:id/userPosition – Updates the coordinates of a user involved on a specific trip that is ongoing.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Parameters** (as URL parameters) – id: string
 - **Response** (as JSON) – { success: boolean, error: string }
- **GET** /trips/:id/usersPosition – Returns the coordinates of all users involved on a specific trip that is ongoing.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Parameters** (as URL parameters) – id: string
 - **Response** (as JSON) – { success: boolean, error: string, stopped: boolean, positions: [UserPositionObject] }

A.4 Ride Request Routes

- **POST** /trips/:id/riderequest – Performs a request to join a trip.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Parameters** (as URL parameters) – id: string

- **Response** (as JSON) – { success: boolean, error: string }
- **POST** /trips/:tripID/riderequest/user/:userID – Accepts or rejects a request to join a trip.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Parameters** (as URL parameters) – userID: string, tripID: string
 - **Parameters** (as form data) – approved: boolean
 - **Response** (as JSON) – { success: boolean, error: string }
- **POST** /trips/:tripID/riderequest – Cancels a request to join a trip.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Parameters** (as URL parameters) – tripID: string
 - **Response** (as JSON) – { success: boolean, error: string }
- **GET** /trips/:id/riderequests – Returns the users that have requested to join a trip.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Parameters** (as URL parameters) – id: string
 - **Response** (as JSON) – { success: boolean, error: string, users: [UserObject] }
- **GET** /trips/:id/riderequested – Verifies if a user has a pending request to join a trip.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Parameters** (as URL parameters) – id: string
 - **Response** (as JSON) – { success: boolean, error: string, requested: boolean }

A.5 Admin Routes

- **GET** /admin/login – Authenticates an administrator on the administrative panel.
 - **Parameters** (as form data) – username: string, password: string
 - **Response** (as JSON) – { success: boolean, error: string, token: string }
- **GET** /admin/logout – Logs out an administrator from the system.
 - **Parameters** (as form data) – username: string, password: string
 - **Response** (as JSON) – { success: boolean, error: string, token: string }
- **GET** /users – Returns information about all users in the system, ordered by name.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Response** (as JSON) – { success: boolean, users: [UserObject] }

FEUPooling Server API

- **POST** /admin/banUser/:id – Ban a user from the system.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Response** (as JSON) – { success: boolean, error: string }
- **POST** /admin/unBanUser/:id – Unban a user from the system.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Response** (as JSON) – { success: boolean, error: string }
- **GET** /admin/settings/pointsPerTrip – Returns the number of points users can win for sharing trips.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Response** (as JSON) – { success: boolean, pointsPerTrip: [PointsPerObject] }
- **POST** /admin/settings/pointsPerTrip – Updates the number of points users can win for sharing trips.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Response** (as JSON) – { success: boolean, error: string }
- **POST** /users/:id/points – Updates a user’s points.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Parameters** (as URL parameters) – id: string
 - **Parameters** (as form data) – description: string, type: string, flow: integer, total: integer
 - **Response** (as JSON) – { success: boolean, error: string }
- **GET** /users/:id/driverRating – Returns the driver rating history of a specific user.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Parameters** (as URL parameters) – id: string
 - **Response** (as JSON) – { success: boolean, error: string, userDriverRatings: [UserRatingObject] }
- **GET** /users/:id/passengerRating – Returns the passenger rating history of a specific user.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Parameters** (as URL parameters) – id: string
 - **Response** (as JSON) – { success: boolean, error: string, userPassengerRatings: [UserRatingObject] }
- **GET** /users/:id/pointsHistory – Returns the points history of a specific user.
 - **Parameters** (as headers) – Authorization: “FP-TOKEN <Authorization Token>”
 - **Parameters** (as URL parameters) – id: string
 - **Response** (as JSON) – { success: boolean, error: string, pointsHistory: [PointsHistoryObject] }

A.6 Models

- **UserObject**

- **id** : integer – User identifier
- **name** : string – User's name
- **email** : string – User's email
- **student_number** : string – User's student number
- **mobile_number** : string – User's mobile phone number
- **points** : integer – User's number of gathered points
- **smoking** : boolean – User's interest of share rides with people that smoke
- **listening_to_music** : boolean – User's interest on sharing rides with people that enjoy listening to music while on a trip
- **talking** : boolean – User's interest on sharing rides with people that like to talk on a trip
- **driver_rating** : integer – User's rating as a driver, given by other users
- **passenger_rating** : integer – User's rating as a passenger, given by other users

- **UserRatingObject**

- **date** : string – Date of the evaluation
- **rating** : integer – Rate given from 0 to 5
- **user_id** : integer – User identifier of the user that received the evaluation
- **trip_id** : integer – User's email
- **evaluator** : UserObject

- **PointsHistoryObject**

- **date** : string – Date of the transaction
- **description** : string – Description of the gain or loss of the points
- **type** : string – Assumes the value "+" for gaining points and "-" for the exchange of points
- **flow** : integer – Quantity of points added or removed
- **total** : integer – Total of points after the transaction

- **NotificationObject**

- **id** : integer – Notification identifier
- **date** : string – Creation date of the notification
- **user_from** : string – Name of the user that started the notification. If the notification was created by an administrator, this field will be blank.

- **description** : string – Description of the notification
- **trip** : integer – Trip identifier. May take the value 0 when no trip is associated to the notification

- **TripObject**

- **id** : integer – Trip identifier
- **date** : string – Date of the trip
- **departure_place** : PlaceObject – Departure place of the trip
- **departure_time** : string – Departure time in hours
- **arrival_place** : PlaceObject – Arrival place of the trip
- **arrival_time** : string – Estimated arrival time (in clock time)
- **description** : string – Extra information about the trip (when provided)
- **car** : string – Car associated with the trip
- **num_free_seats** : integer – Number of free seats in the car (between 1 and 3)
- **status** : string – Status of the trip. May acquire the values “pre” (before start), “on” (during trip) and “pos” (after)
- **driver** : UserObject – Driver of the trip
- **passengers** : [UserObject] – Passengers of the trip

- **PlaceObject**

- **id** : integer – Place identifier
- **name** : string – Name of the place
- **city** : string – City where the place belongs
- **coord** : string – Geographic coordinates of the place in the form of Latitude,Longitude

- **UserPositionObject**

- **user** : UserObject – Passenger
- **latitude** : string – Latitude of the person’s location
- **longitude** : string –Longitude of the person’s location
- **date** : string – Date and time when the position was registered

- **PointsPerObject**

- **points_as_a_driver** : integer – Points that user can win as driver
- **points_as_a_passenger** : integer – Points that user can win as passenger


Appendix B

Screenshots

B.1 *FEUPooling* Web

FEUPooling Log In

Authenticate

Username 


Password 

Figure B.1: Login - *FEUPooling* Web

Screenshots

FEUPooling Users Trips Settings Welcome, admin! Log Out

Users

Filter:

Name ▲	Email	Student Number	Mobile Number	Points		
Ana Rocha	ei13006@fe.up.pt	201309190	917310323	25	Details	Ban
Bruno Gonçalves	ei11106@fe.up.pt	201109190	969078408	153	Details	Ban
Bruno Silva	ei12206@fe.up.pt	201209190	911079308	56	Details	Ban
Eduardo Almeida	ei12018@fe.up.pt	201204989	917775555	73	Details	Ban
João Pereira	ei12182@fe.up.pt	201209182	937310323	0	Details	Ban
Manuel Oliveira	ei10206@fe.up.pt	201009190	969079308	0	Details	Ban

5 10 15

Figure B.2: Users' list - FEUPooling Web

Screenshots

FEUPooling Users **Trips** Settings Welcome, admin! Log Out

Trips

Filter:

Date ▼	Departure Place	Arrival Place	Driver	Passengers	
2017-06-19 09:05	R. Azenha	FEUP	Bruno Gonçalves	1	Details
2017-06-18 12:35	FEUP	Rio Tinto	Bruno Gonçalves	1	Details
2017-06-16 06:20	R. Maria Pia	FEUP	Eduardo Almeida	1	Details
2017-06-16 09:05	R. Azenha	FEUP	Bruno Gonçalves	1	Details
2017-06-15 13:15	FEUP	Rio Tinto	Bruno Gonçalves	1	Details
2017-06-15 14:27	R. Amália Rodrigues	FEUP	Bruno Gonçalves	1	Details
2017-06-15 00:00	Porto	FEUP	Bruno Silva	0	Details
2017-06-12 09:05	R. Azenha	FEUP	Bruno Gonçalves	1	Details
2017-06-10 01:41	Porto	FEUP	Bruno Gonçalves	0	Details
2017-06-09 09:05	R. Azenha	FEUP	Bruno Gonçalves	1	Details

« 1 2 3 4 » 5 10 15

Figure B.3: Trips' list - FEUPooling Web

Screenshots

FEUPooling Users Trips Settings Welcome, admin! Log Out

Map Passengers

Date	2017-06-19
Departure Time	09:05
Departure Place	A
Arrival Time	09:36
Arrival Place	B
Driver	Bruno Gonçalves

The map displays a route starting at Trofa (marked 'A') and ending at São Mamede de Infesta (marked 'B'). The route is highlighted in blue and passes through several towns including Covelas, Folgosa, and Nogueira. The map includes various road types (N, M, A) and geographical features like rivers and green spaces. The Google Maps interface is visible at the bottom of the map area.

Figure B.4: Trip page - FEUPooling Web

Screenshots

The screenshot displays the 'Trips' page in the FEUPooling web application. The top navigation bar includes 'FEUPooling', 'Users', 'Trips', and 'Settings'. On the right, it shows 'Welcome, admin!' and a 'Log Out' button. The main content area is divided into two sections: 'Map' and 'Passengers'. The 'Map' section is currently inactive. The 'Passengers' section contains a table with two columns: 'Name' and 'Role'. The table lists two entries: Bruno Gonçalves (Driver) and Eduardo Almeida (Passenger). Each entry has a 'Profile' button next to it. To the left of the passengers table, there is a table of trip details.

Date	2017-06-19
Departure Time	09:05
Departure Place	A
Arrival Time	09:36
Arrival Place	B
Driver	Bruno Gonçalves

Name ▲	Role	
Bruno Gonçalves	Driver	Profile
Eduardo Almeida	Passenger	Profile

Figure B.5: Trip page - *FEUPooling* Web

Screenshots

The screenshot shows the 'Settings' page of the FEUPooling web application. The navigation bar at the top includes 'FEUPooling', 'Users', 'Trips', and 'Settings' (which is highlighted). On the right side of the navigation bar, it says 'Welcome, admin!' and 'Log Out'. The main heading is 'Points received per trip'. Below this, there are two input fields. The first is labeled 'As a driver' and has a value of 30. The second is labeled 'As a passenger' and has a value of 20. Each input field has a red minus button on the left and a green plus button on the right. At the bottom center, there is a green 'Save' button.

Role	Points received per trip
As a driver	30
As a passenger	20

Figure B.6: Settings page - *FEUPooling* Web

Screenshots

FEUPooling Users Trips Settings Welcome, admin! Log Out

Points Driver Passenger

Points History

Date	Description	Flow	Total
2017-06-16 11:59:35	Received points for being the driver on a valid trip	+30	153
2017-06-13 17:17:18	Received points for being the driver on a valid trip	+30	123
2017-06-13 15:35:53	Received points for being the driver on a valid trip	+30	93

Add/Exchange Points

Description

Quantity

- 10 +

Add Remove

User Profile:

- Name: Bruno Gonçalves
- Email: ei11106@fe.up.pt
- Student Number: 201109190
- SiFeup Page: [URL](#)
- Mobile Number: 969078408
- Points: 153
- Driver Rating: ★★★★★
- Passenger Rating: ★★★★★

Figure B.7: User page - FEUPooling Web

Screenshots

FEUPooling Users Trips Settings Welcome, admin! Log Out

Profile Information:

Name	Bruno Gonçalves
Email	ei11106@fe.up.pt
Student Number	201109190
SiFeup Page	URL
Mobile Number	969078408
Points	153
Driver Rating	★ ★ ★ ★ ★
Passenger Rating	★ ★ ★ ★ ★

Trips as a driver

Points Driver Passenger

Date	Departure Place	Arrival Place	Passengers	Details
2017-06-19 09:05	R. Azenha	FEUP	1	Details
2017-06-18 12:35	FEUP	Rio Tinto	1	Details
2017-06-16 09:05	R. Azenha	FEUP	1	Details
2017-06-15 13:15	FEUP	Rio Tinto	1	Details
2017-06-15 14:27	R. Amália Rodrigues	FEUP	1	Details
2017-06-12 09:05	R. Azenha	FEUP	1	Details
2017-06-10 01:41	Porto	FEUP	0	Details
2017-06-09 09:05	R. Azenha	FEUP	1	Details
2017-06-09 00:00	FEUP	Rio Tinto	1	Details

Evaluation Received

Date	Name	Student Number	Rating
2017-06-13 13:53:07	Eduardo Almeida	201204989	0 out of 5
2017-06-12 09:07:40	Eduardo Almeida	201204989	1 out of 5

Figure B.8: User page - FEUPooling Web

Screenshots

FEUPooling Users Trips Settings Welcome, admin! Log Out

Profile Information:

- Name: Bruno Gonçalves
- Email: ei11106@fe.up.pt
- Student Number: 201109190
- SiFeup Page: URL
- Mobile Number: 969078408
- Points: 153
- Driver Rating: ★★★★★
- Passenger Rating: ★★★★★

Trips as a passenger

Date	Departure Place	Arrival Place	Passengers	
2017-06-16 06:20	R. Maria Pia	FEUP	1	Details
2017-06-09 07:00	FEUP	R. do Adro	1	Details

Evaluation Received

No evaluation received as a passenger

Figure B.9: User page - FEUPooling Web

B.2 FEUPooling Mobile



Figure B.10: Splash screen - *FEUPooling* Mobile

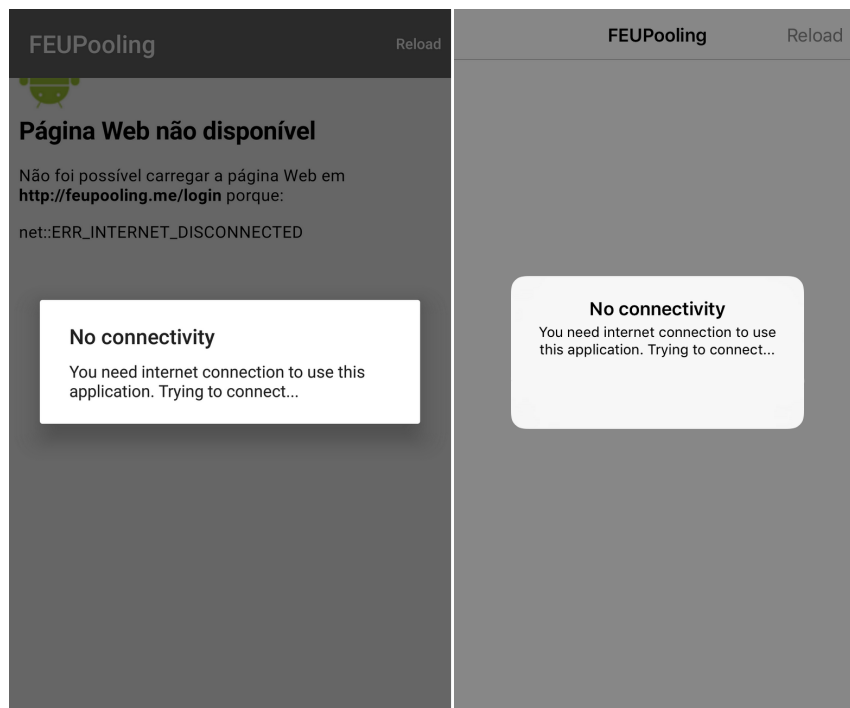


Figure B.11: Login page without internet connection - *FEUPooling* Mobile

Screenshots

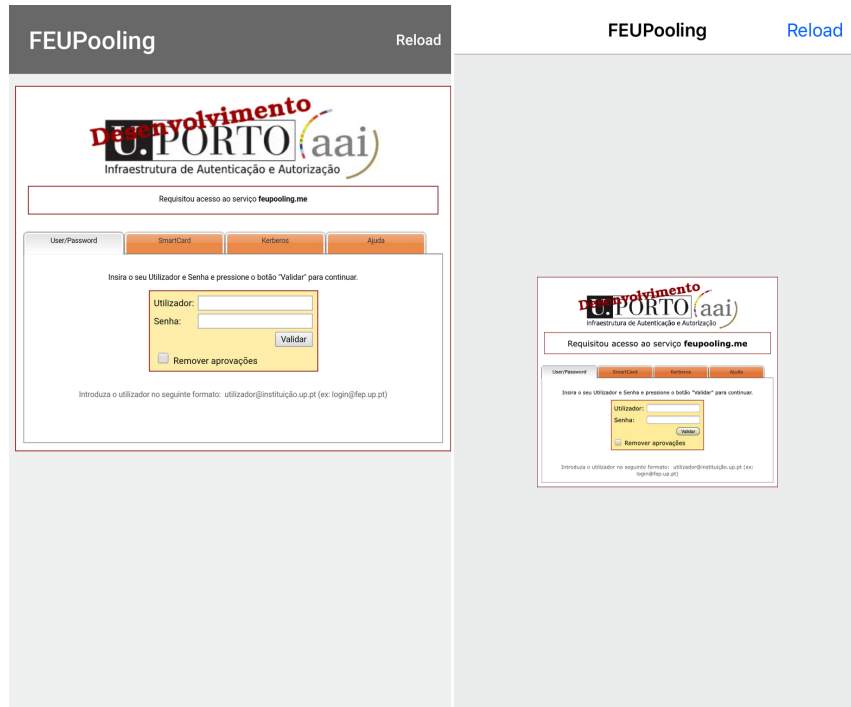


Figure B.12: Login page - FEUPooling Mobile

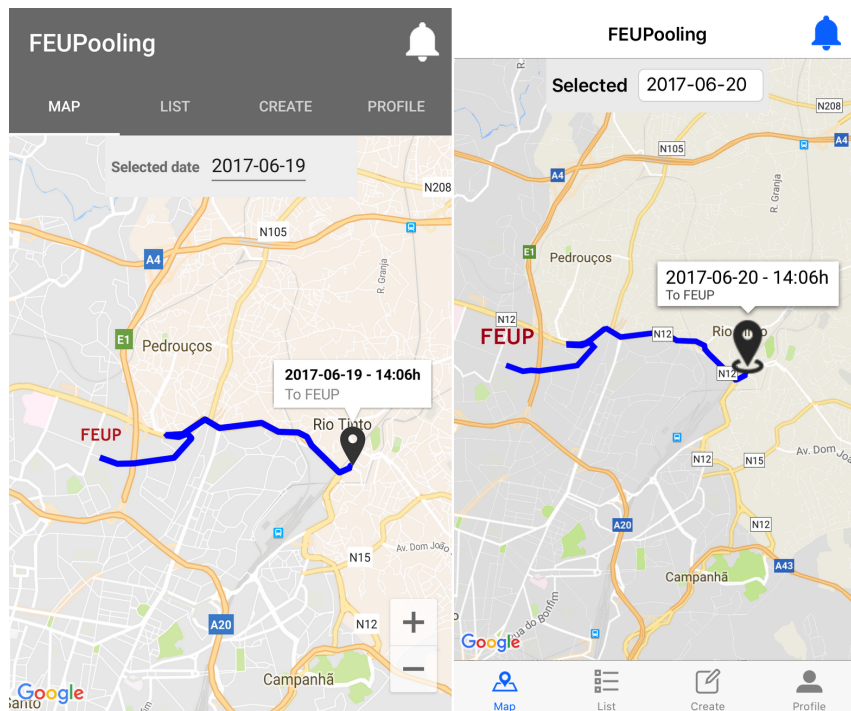


Figure B.13: Main screen map - FEUPooling Mobile

Screenshots

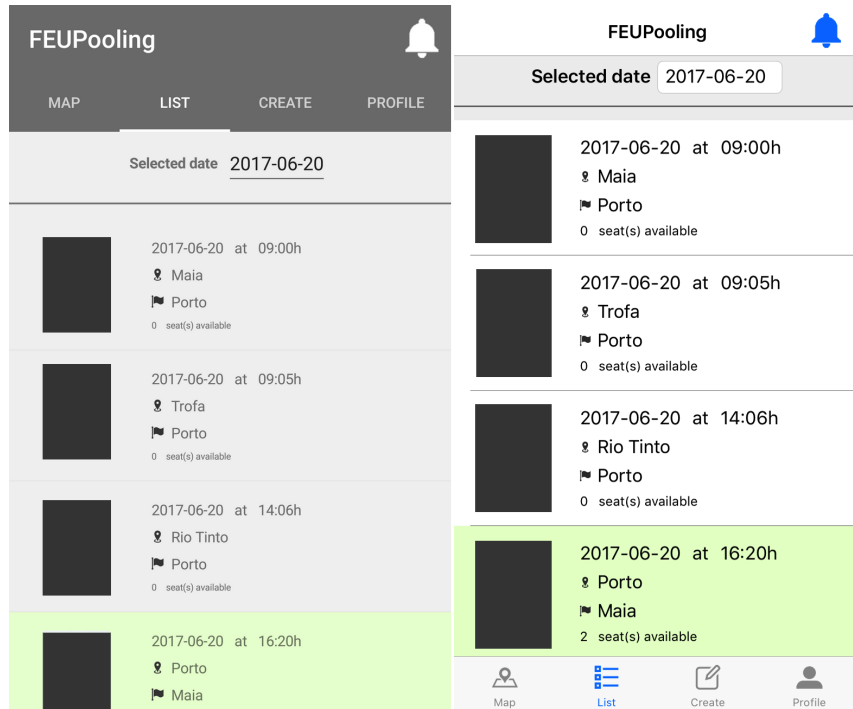


Figure B.14: Trip list - *FEUPooling* Mobile

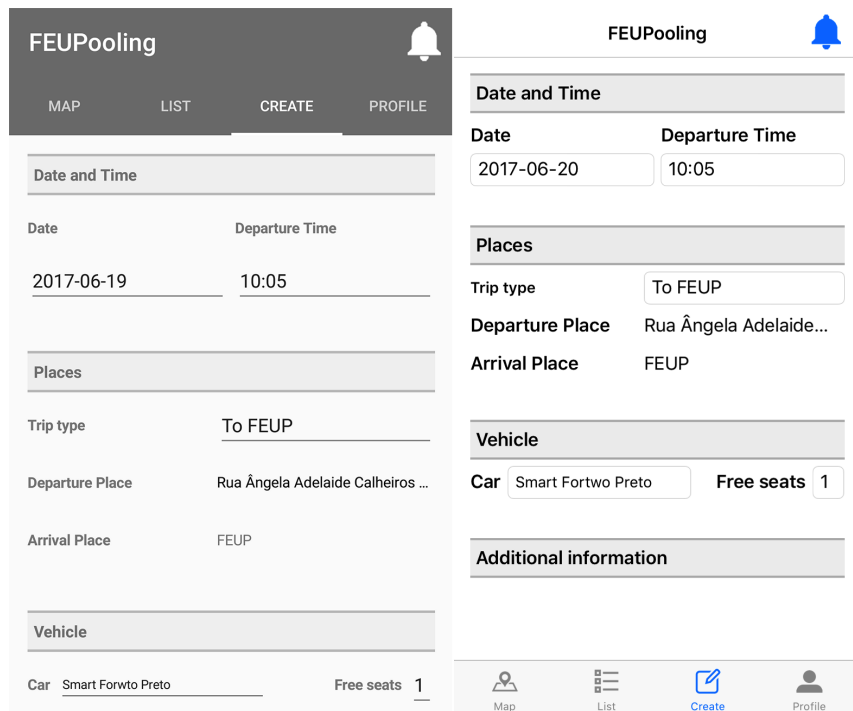


Figure B.15: Create trip part 1 - *FEUPooling* Mobile

Screenshots

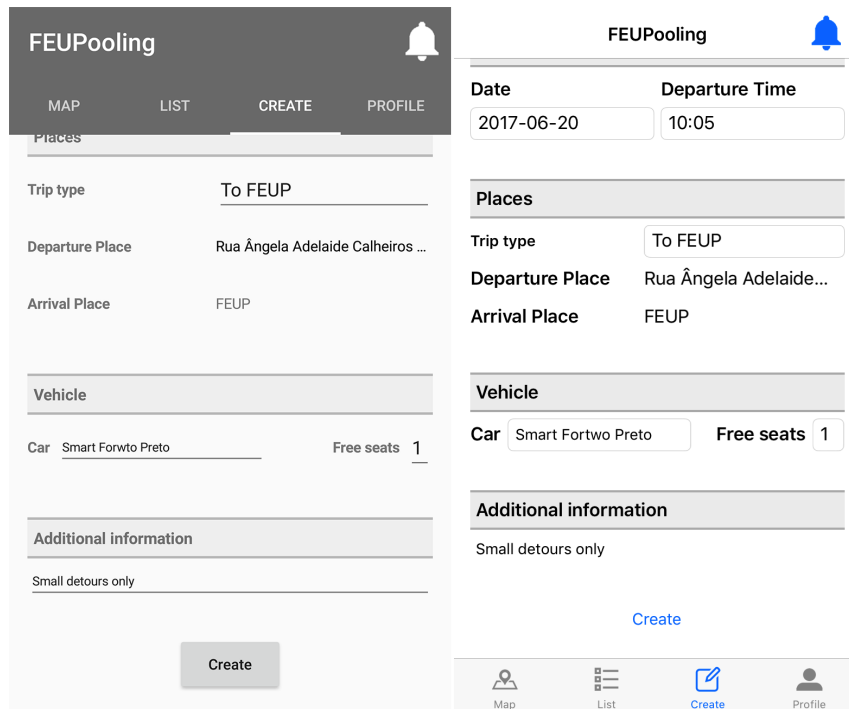


Figure B.16: Create trip part 2 - FEUPooling Mobile

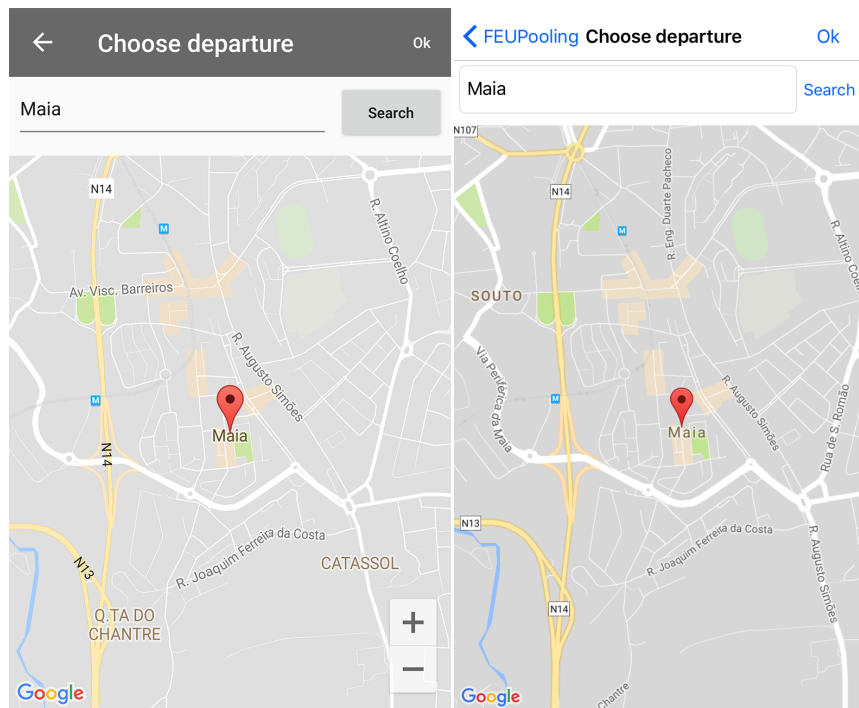


Figure B.17: Departure or arrival place choosing view - FEUPooling Mobile

Screenshots

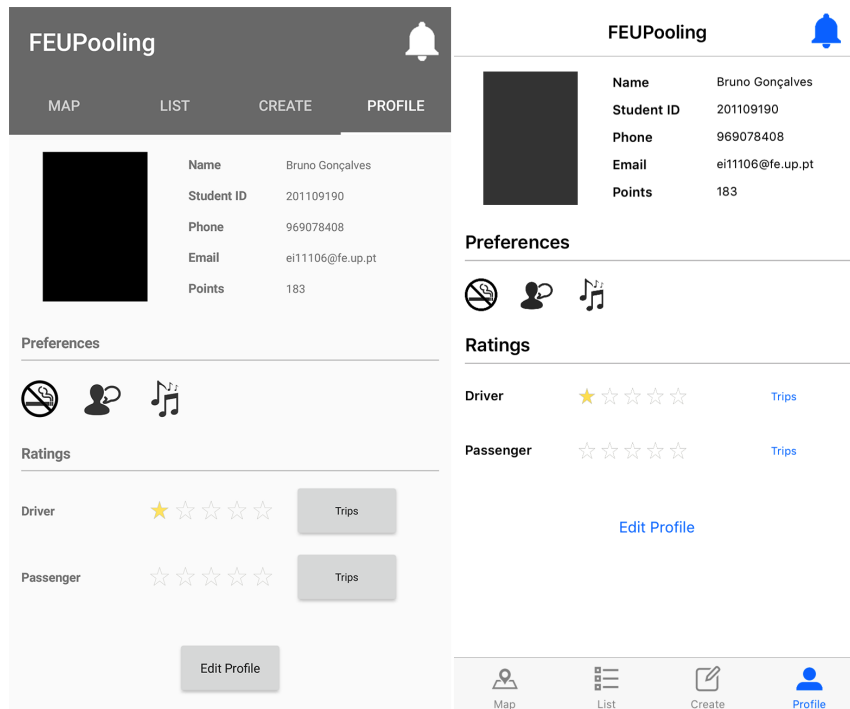


Figure B.18: User profile - *FEUPooling* Mobile

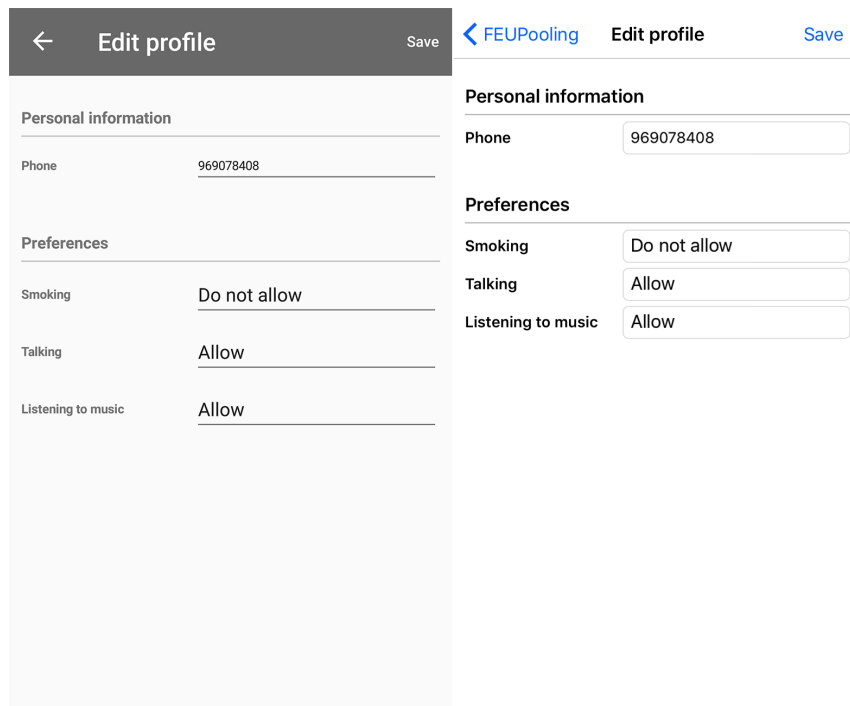


Figure B.19: Edit user profile - *FEUPooling* Mobile

Screenshots

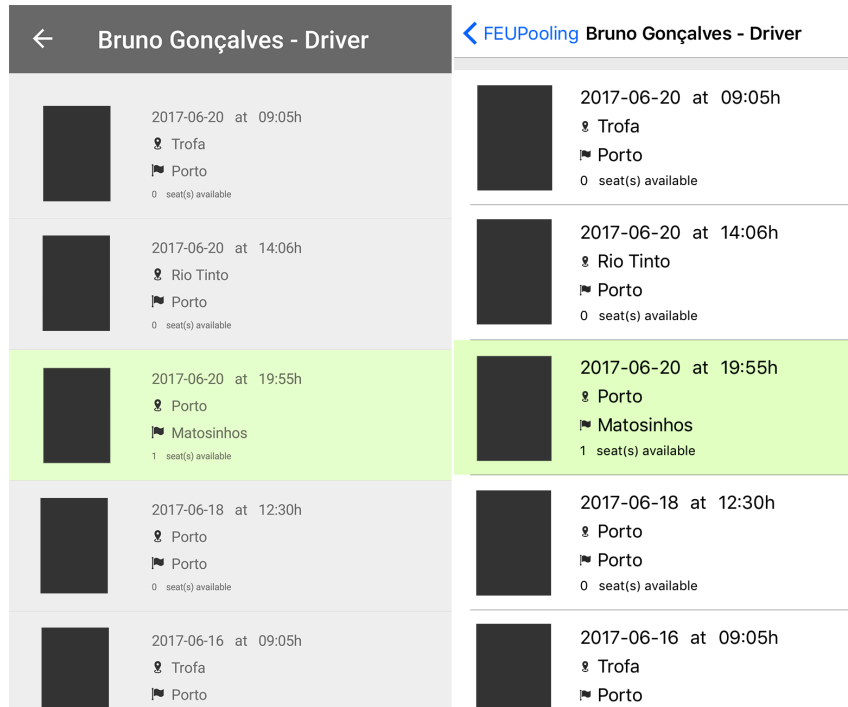


Figure B.20: Trips list of a user as a driver - *FEUPooling* Mobile

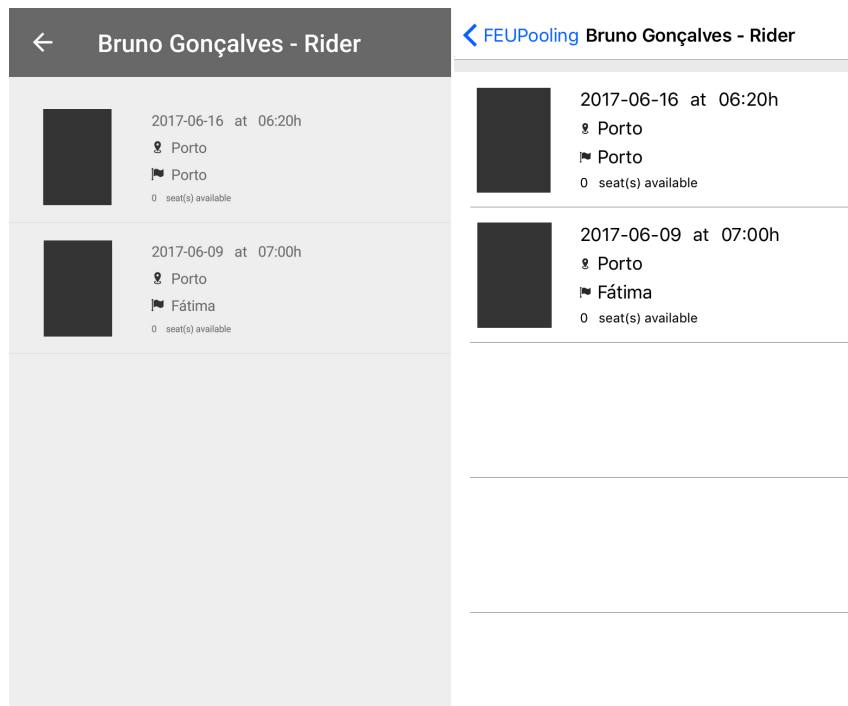


Figure B.21: Trips list of a user as a passenger - *FEUPooling* Mobile

Screenshots

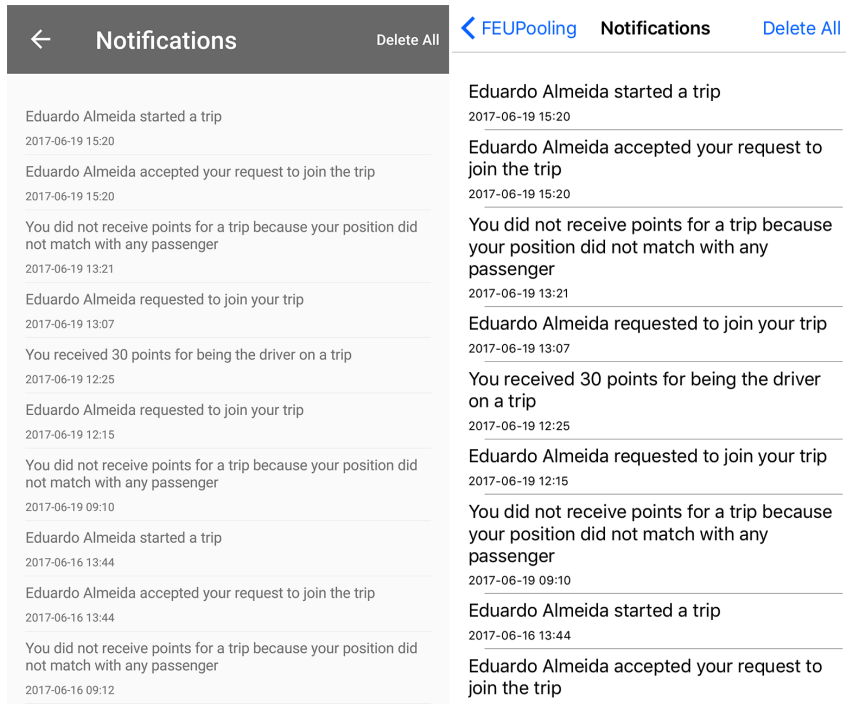


Figure B.22: Notifications - FEUPooling Mobile

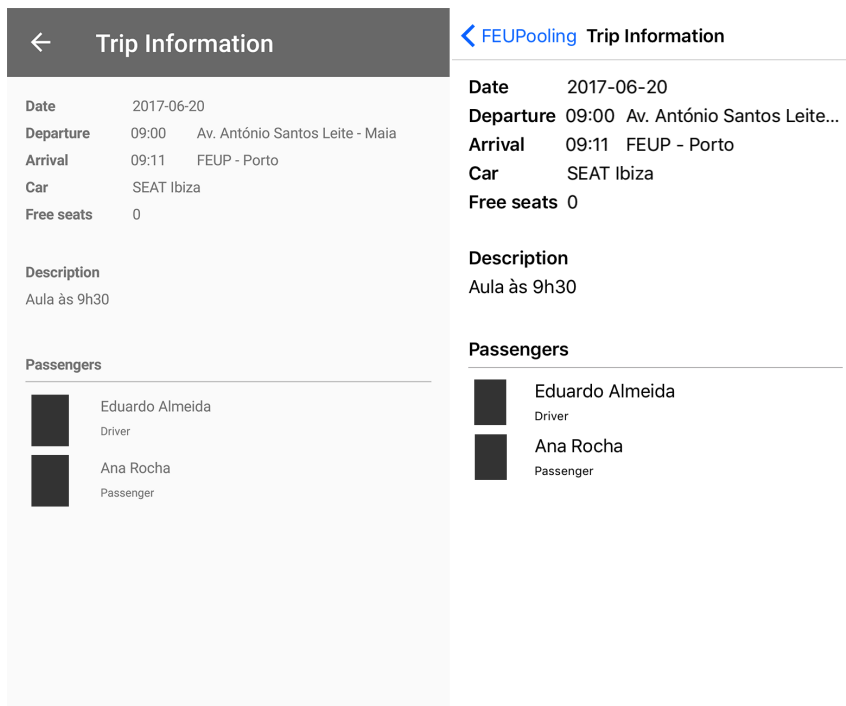


Figure B.23: Trip page - FEUPooling Mobile

Screenshots

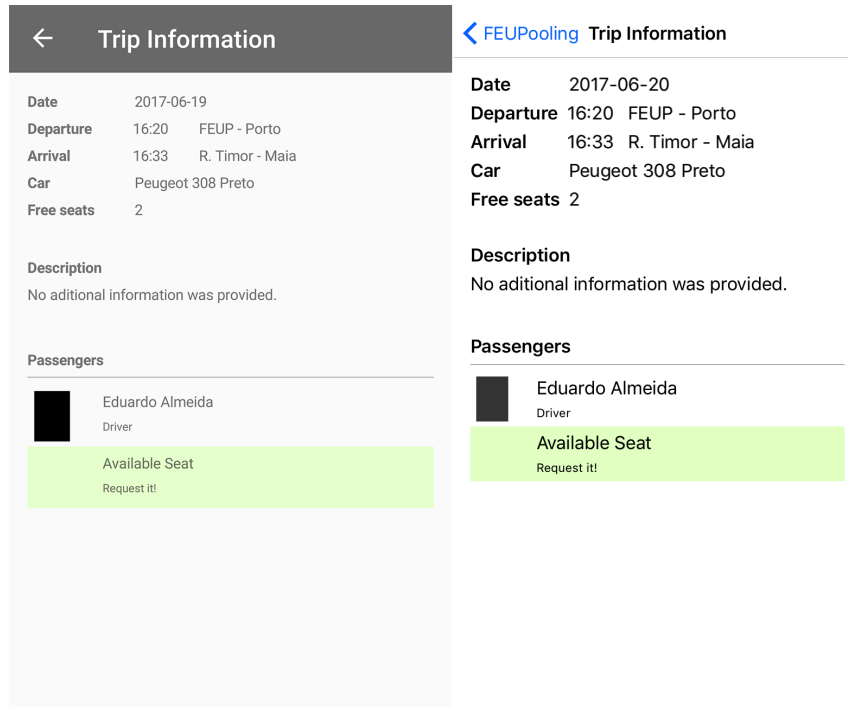


Figure B.24: Trip page with available seats - *FEUPooling* Mobile

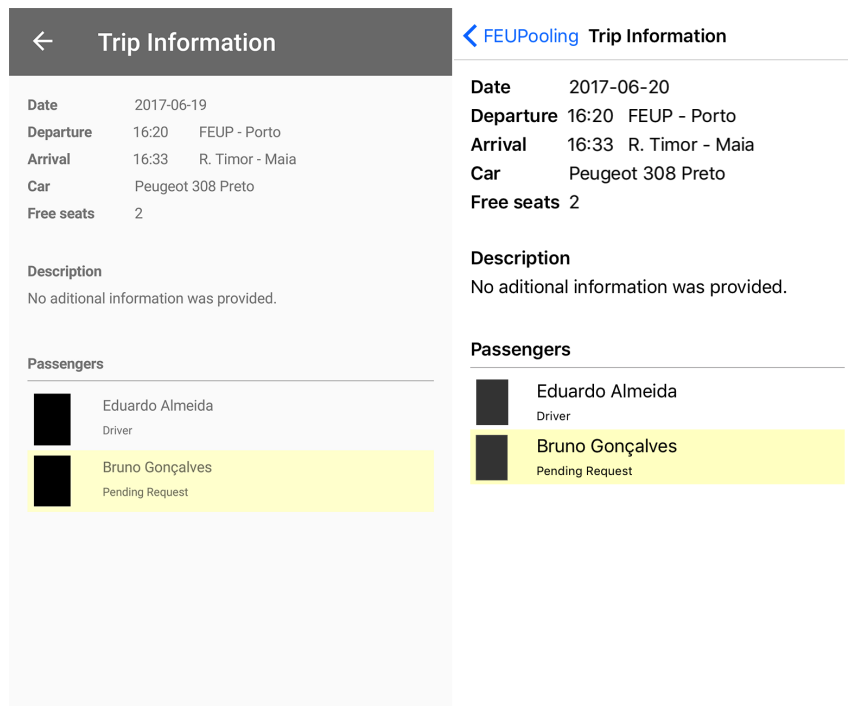


Figure B.25: Trip page with pending request made by the user - *FEUPooling* Mobile

Screenshots

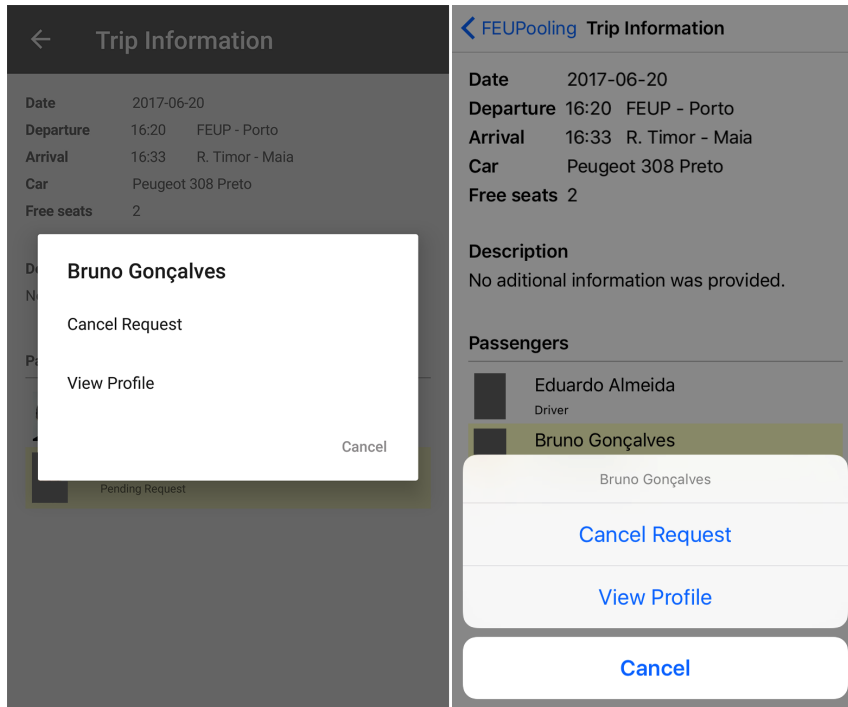


Figure B.26: Cancel request made by the user - *FEUPooling* Mobile

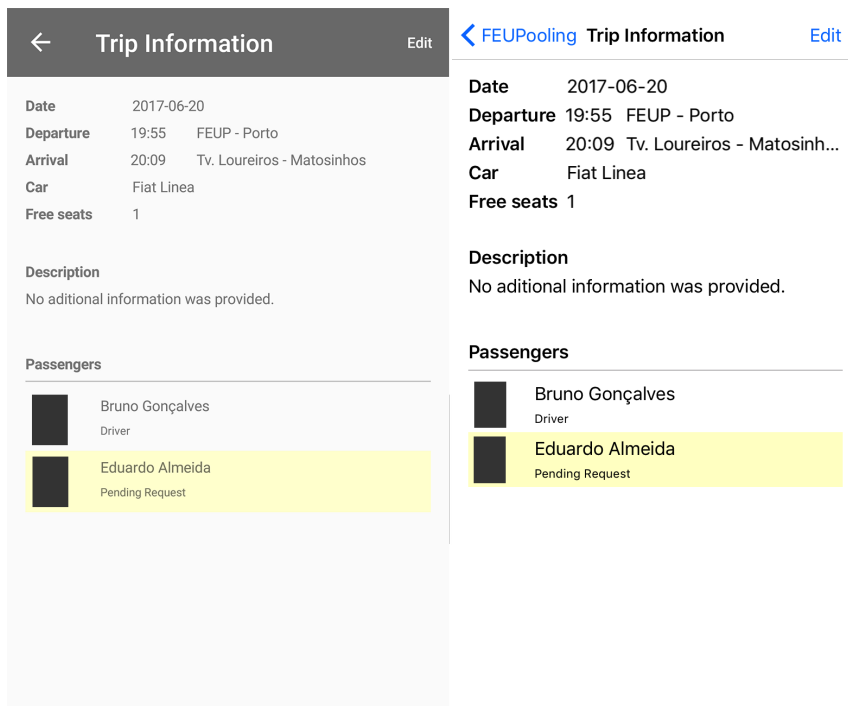


Figure B.27: Trip page with pending request waiting for the driver's approval - *FEUPooling* Mobile

Screenshots

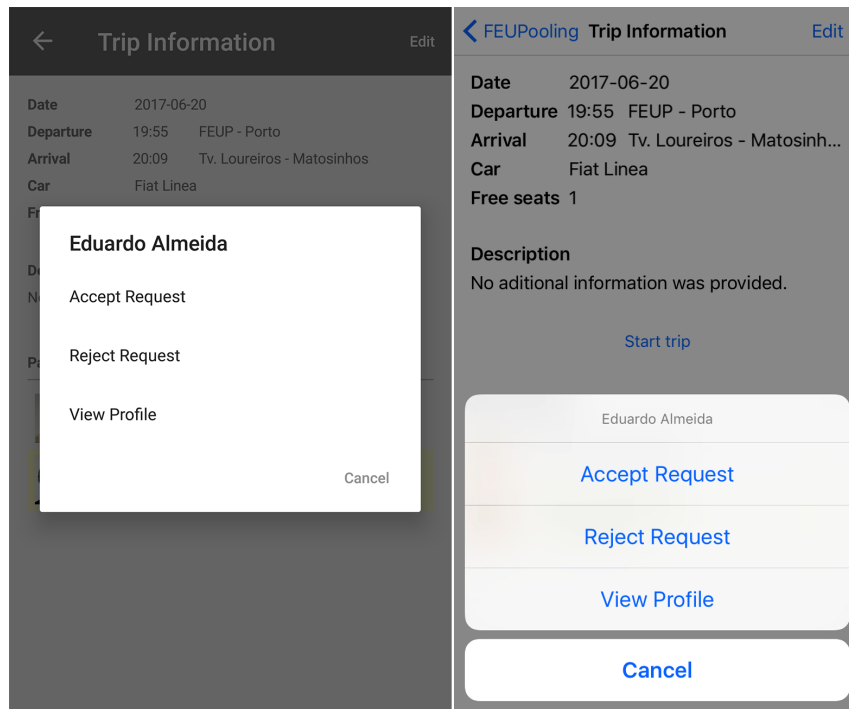


Figure B.28: Approval options for the driver's approval - *FEUPooling* Mobile

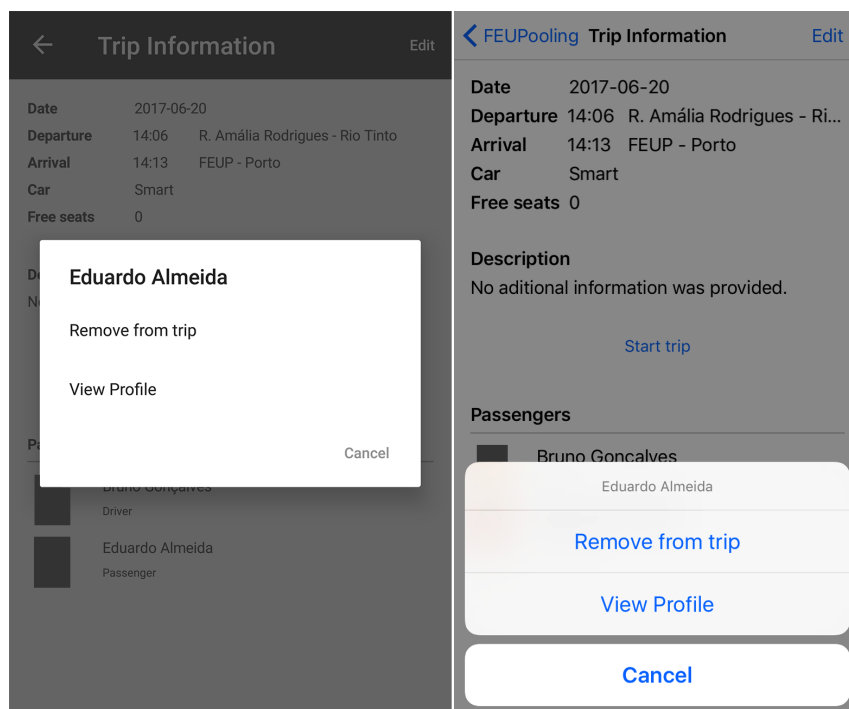


Figure B.29: Driver's removal option from a trip - *FEUPooling* Mobile

Screenshots

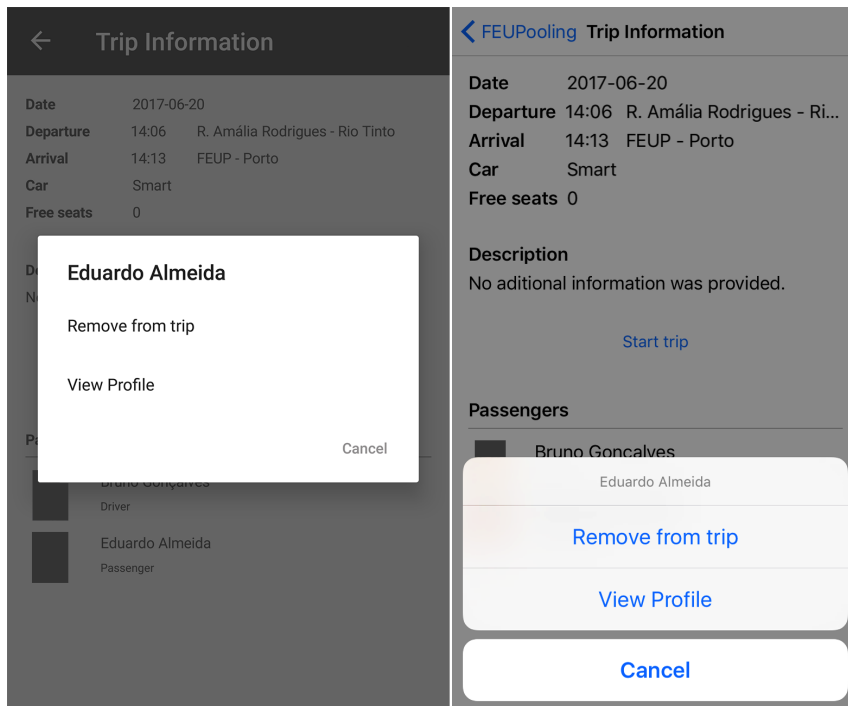


Figure B.30: Passenger's removal option from a trip - *FEUPooling* Mobile

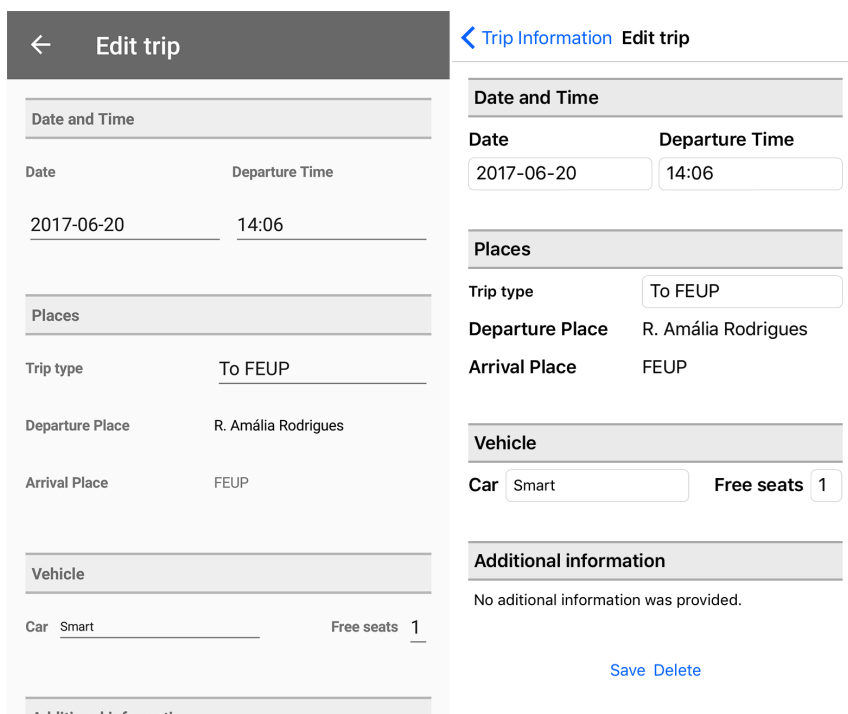


Figure B.31: Edit trip page part 1 - *FEUPooling* Mobile

Screenshots

Left Screenshot: Edit trip

Places

Trip type: To FEUP

Departure Place: R. Amália Rodrigues

Arrival Place: FEUP

Vehicle

Car: Smart Free seats: 1

Additional information

No additional information was provided.

Save Delete

Right Screenshot: Trip Information

Date and Time

Date: 2017-06-20 Departure Time: 14:06

Places

Trip type: To FEUP

Departure Place: R. Amália Rodrigues

Arrival Place: FEUP

Vehicle

Car: Smart Free seats: 1

Additional information

No additional information was provided.

Save Delete

Figure B.32: Edit trip page part 2 - *FEUPooling* Mobile

Left Screenshot: Trip Information

Date: 2017-06-19

Departure: 09:05 R. Azenha - Trofa

Arrival: 09:36 FEUP - Porto

Car: Smart

Free seats: 0

Description

No additional information was provided.

Start trip

Passengers

Bruno Gonçalves
Driver

Eduardo Almeida
Passenger

Right Screenshot: FEUPooling Trip Information

Date: 2017-06-20

Departure: 09:05 R. Azenha - Trofa

Arrival: 09:36 FEUP - Porto

Car: Smart

Free seats: 0

Description

No additional information was provided.

Start trip

Passengers

Bruno Gonçalves
Driver

Eduardo Almeida
Passenger

Figure B.33: Trip page ready to start - *FEUPooling* Mobile

Screenshots

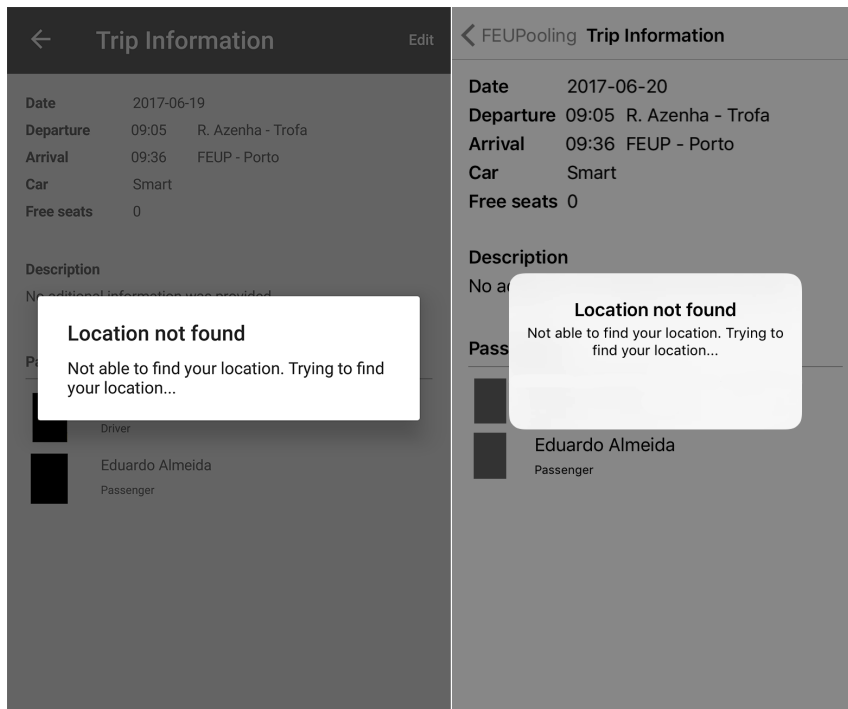


Figure B.34: Trying to find user position before start trip - FEUPooling Mobile

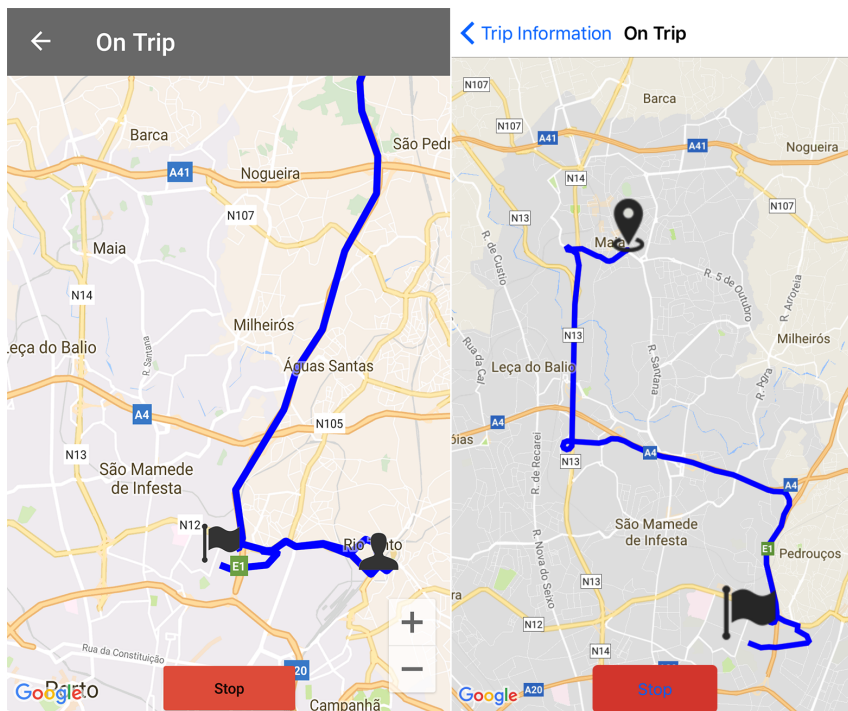


Figure B.35: On trip mode for the driver - FEUPooling Mobile

Screenshots

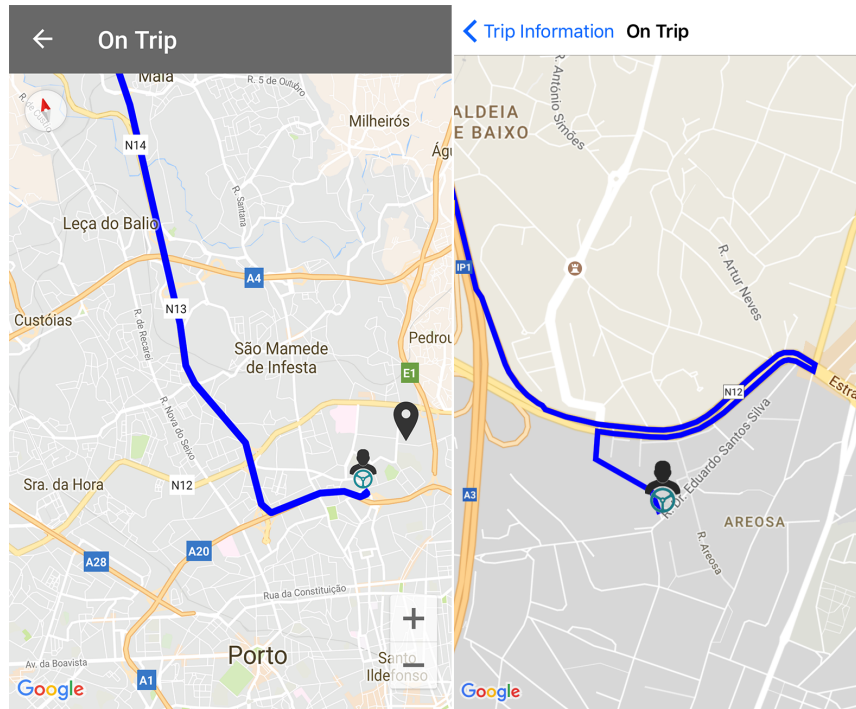


Figure B.36: On trip mode for passengers - FEUPooling Mobile

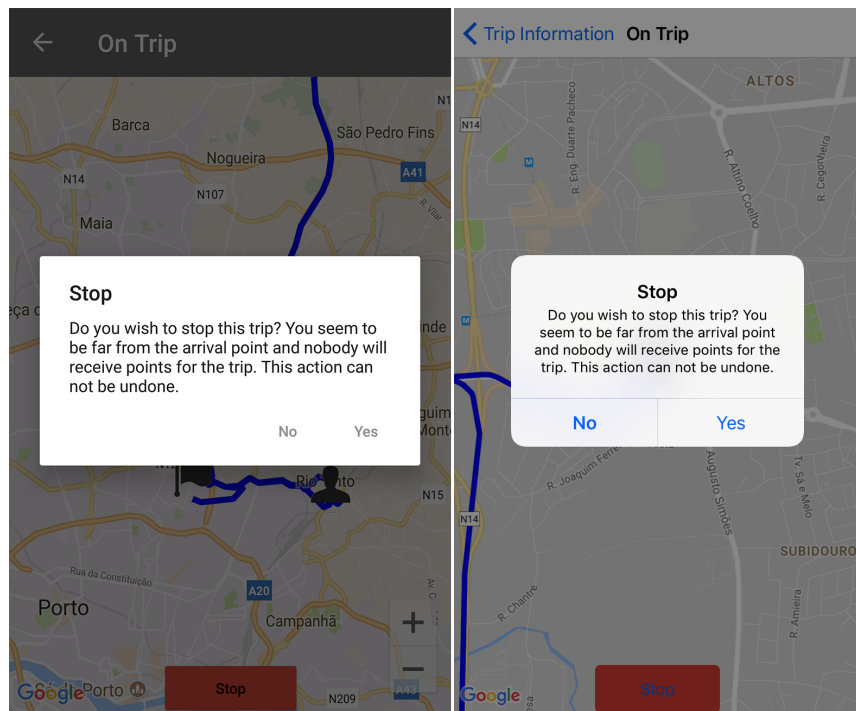


Figure B.37: Trying to stop a trip far from the arrival location - FEUPooling Mobile

Screenshots

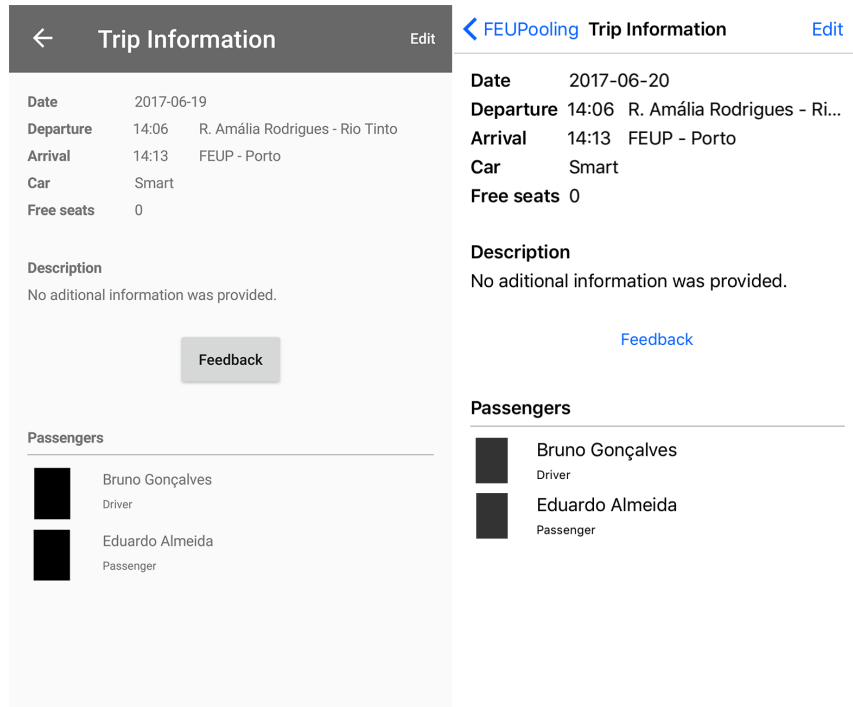


Figure B.38: Trip page that already ended - *FEUPooling* Mobile

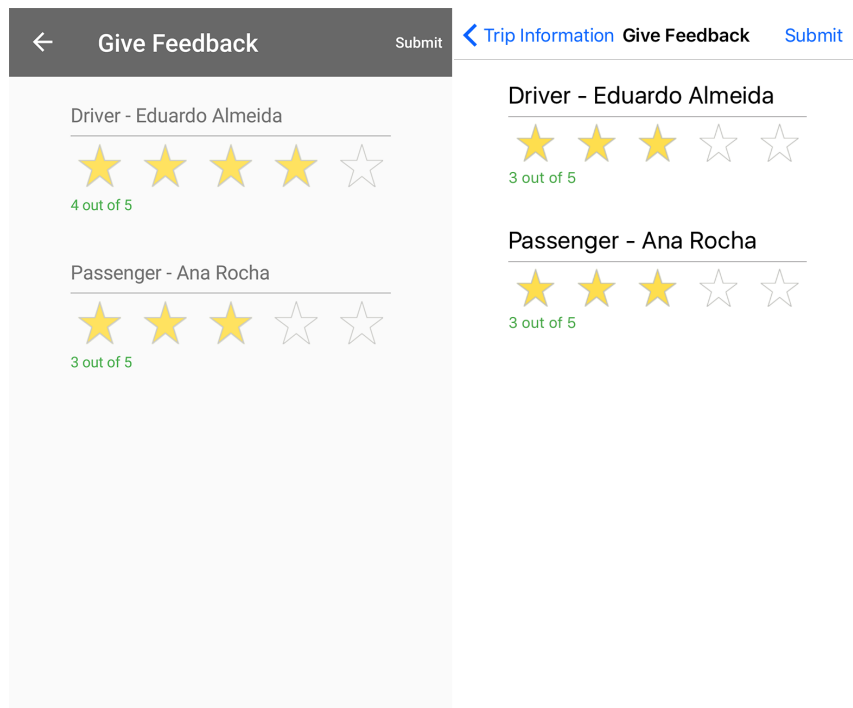


Figure B.39: Rating page - *FEUPooling* Mobile

Appendix C

FEUPooling Server Database Schema

A database was necessary to store all the data required for both the mobile and web parts of *FEUPooling*. The relational database management system used was *MySQL*. The database schema, which is its structure, was the following:

- *admin_login_tokens* – Stores login tokens to administrators of the *FEUPooling Web*.
 - *id* : ObjectId
 - *admin_id* : ObjectId(*user_admins*)
 - *token* : String
 - *created_at* : Date
 - *updated_at* : Date

- *login_tokens* – Stores login tokens for the users of *FEUPooling Mobile*.
 - *id* : ObjectId
 - *user_id* : ObjectId(*users*)
 - *token* : String
 - *created_at* : Date
 - *updated_at* : Date

- *notifications* – Stores notifications for users of *FEUPooling Mobile*. These notifications can be triggered by another user or by an administrator. Notifications triggered by other users have a trip associated, while administrative notifications are triggered by the administrators of the system, which an example is the acquisition or exchange of points.
 - *id* : ObjectId
 - *user_to_id* : ObjectId(*users*)

FEUPooling Server Database Schema

- user_from_id : ObjectId(users)
- trip_id : ObjectId(trips)
- description : String
- created_at : Date
- updated_at : Date
- on_trip_positions – Stores user positions for ongoing trips.
 - id : ObjectId
 - trip_id : ObjectId(trips)
 - user_id : ObjectId(users)
 - latitude : String
 - longitude : String
 - created_at : Date
 - updated_at : Date
- places – Stores places associated with trips.
 - id : ObjectId
 - name : String
 - city : String
 - coord : String
 - created_at : Date
 - updated_at : Date
- points_histories – Stores the history of points associated with a user, which can be of a negative or positive type.
 - id : ObjectId
 - user_id : ObjectId(users)
 - description : String
 - type : String
 - flow : Integer
 - total : Integer
 - created_at : Date
 - updated_at : Date
- ride_requests – Stores the ride requests, triggered by the users when these request to join a trip.

FEUPooling Server Database Schema

- id : ObjectId
- user_id : ObjectId(users)
- trip_id : ObjectId(trips)
- approved : Boolean
- created_at : Date
- updated_at : Date
- settings – Stores some system settings, such as the points to be acquired per trip as a driver and as a passenger.
 - id : ObjectId
 - type : String
 - value : String
 - created_at : Date
 - updated_at : Date
- trips – Stores trips created by the users.
 - id : ObjectId
 - departure_id : ObjectId(places)
 - arrival_id : ObjectId(places)
 - date : Date
 - departure_time : Time
 - arrival_time : Time
 - description : String
 - car : String
 - num_free_seats : Integer
 - driver_id : ObjectId(users)
 - status : String
 - created_at : Date
 - updated_at : Date
- trip_passengers – Stores the passengers of trips.
 - id : ObjectId
 - trip_id : ObjectId(trips)
 - user_id : ObjectId(users)

FEUPooling Server Database Schema

- created_at : Date
 - updated_at : Date
- trip_validations – Stores all the users' position checks for an ongoing trip.
 - id : ObjectId
 - user_id : ObjectId(users)
 - trip_id : ObjectId(trips)
 - valid : Boolean
 - created_at : Date
 - updated_at : Date
- users – Stores all the users of the system.
 - id : ObjectId
 - name : String
 - email : String
 - student_number : String
 - mobile_number : String
 - image : Text
 - smoking : Boolean
 - listening_to_music : Boolean
 - talking : Boolean
 - banned : Boolean
 - created_at : Date
 - updated_at : Date
- user_admins – Stores all the administrators of the system.
 - id : ObjectId
 - name : String
 - username : String
 - password : String
 - email : String
 - created_at : Date
 - updated_at : Date
- user_driver_ratings – Stores all the ratings received by the users as drivers.

FEUPooling Server Database Schema

- id : ObjectId
 - driver_id : ObjectId(users)
 - evaluator_id : ObjectId(users)
 - trip_id : ObjectId(trips)
 - rating : Integer
 - created_at : Date
 - updated_at : Date
- user_rider_ratings – Stores all the ratings received by the users as passengers.
 - id : ObjectId
 - rider_id : ObjectId(users)
 - evaluator_id : ObjectId(users)
 - trip_id : ObjectId(trips)
 - rating : Integer
 - created_at : Date
 - updated_at : Date

FEUPooling Server Database Schema

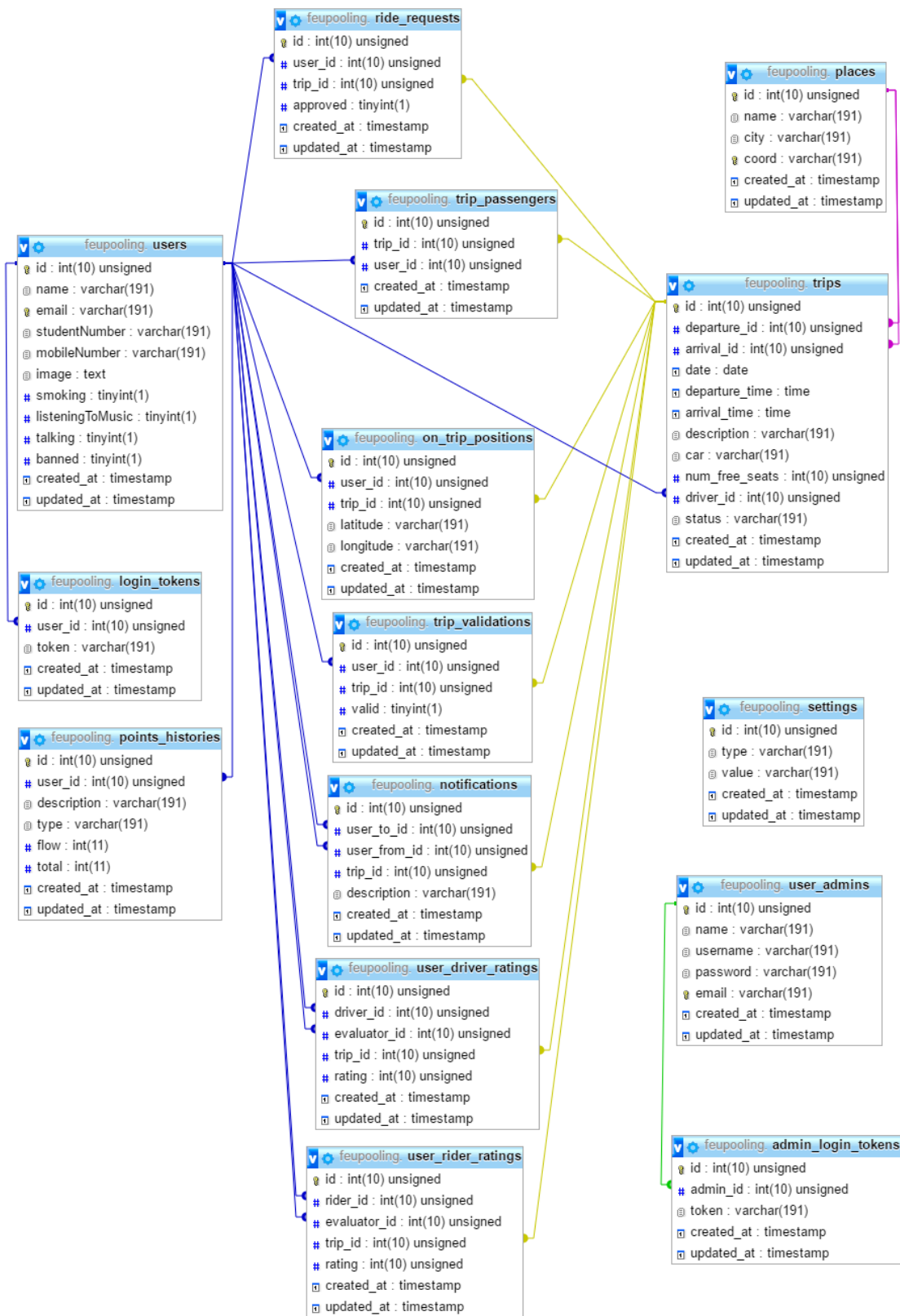


Figure C.1: Database Schema

References

- [ama17] amay077. Xamarin.Forms.GoogleMaps, 2017. Repository on GitHub. URL: <https://github.com/amay077/Xamarin.Forms.GoogleMaps> [accessed June 16th, 2017].
- [BB09] Matthew Barth and Kanok Boriboonsomsin. Traffic congestion and greenhouse gases. *ACCESS Magazine*, 1(35), 2009.
- [Bla17] BlaBlaCar. BlaBlacar, 2017. URL: <https://www.blablacar.pt/> [accessed January 17th, 2017].
- [BSB⁺10] Ron N Buliung, Kalina Soltys, Randy Bui, Catherine Habel and Ryan Lanyon. Catching a ride on the information super-highway: toward an understanding of internet-based carpool formation and use. *Transportation*, 37(6):849–873, 2010.
- [Cab17] Cabify. Cabify, 2017. URL: <https://www.cabify.com> [accessed January 20th, 2017].
- [Cam02] Márcia Campos. Mobilidade Urbana - Carpooling. Master’s thesis, Faculdade de Engenharia da Universidade do Porto, 10 2002.
- [Cen] IBM Knowledge Center. Security Assertion Markup Language (SAML) and application security. URL: https://www.ibm.com/support/knowledgecenter/SS8CCV_7.6.0.3/com.ibm.mbs.doc/securgroup/c_saml_security.html [accessed June 16th, 2017].
- [Cit15] Citydrive. Citydrive, 2015. URL: <https://www.citydrive.pt> [accessed January 27th, 2017].
- [cN17] car2go NA. car2go - Hop in and drive, 2017. URL: <https://www.car2go.com/US/en/> [accessed January 22nd, 2017].
- [Col17] Peninsula College. Peninsula College Institutional Procedure - Carpool Program , 2017. URL: <http://www.pencol.edu/sites/default/files/Carpool%20Guide.pdf> [accessed January 27th, 2017].
- [Cot17] Alejandro Cotroneo. laravel-saml2, 2017. Repository on GitHub. URL: <https://github.com/aacotroneo/laravel-saml2> [accessed June 16th, 2017].
- [DB16] Maria Dias and José Borges. Understanding and conceptualizing interaction. *Interação Pessoa-Computador, MIEIC - FEUP — 2016/2017 Class Handouts*, 2016.
- [Dic] Oxford Dictionaries. Definition of carpool in English. URL: <https://en.oxforddictionaries.com/definition/carpool> [accessed January 20th, 2017].

REFERENCES

- [dN16] Jornal de Notícias. Taxistas contra legalização da Uber, 2016. URL: <http://www.jn.pt/economia/interior/taxistas-contra-legalizacao-da-uber-5261478.html> [accessed January 18th, 2017].
- [dN17] Jornal de Notícias. UberGREEN salta de Matosinhos para todo o mundo, 2017. URL: <http://www.jn.pt/inovacao/interior/ubergreen-salta-de-matosinhos-para-todo-o-mundo-5469071.html> [accessed January 18th, 2017].
- [Esc16] Sebastian Eschweiler. *Angular 2: A Practical Introduction to the new Web Development Platform*. LeanPub, 2016. URL: <https://books.google.pt/books?id=MUelBQAAQBAJ>.
- [Fer16] Ana Bela Ferreira. Uber lança viagens partilhadas na Web Summit, 2016. URL: <http://www.dn.pt/sociedade/interior/uber-lanca-viagens-partilhadas-na-web-summit-5474880.html> [accessed January 18th, 2017].
- [Fol17] Mariusz Foltak. angular2-datatable, 2017. Repository on GitHub. URL: <https://github.com/mariuszfoltak/angular2-datatable> [accessed June 16th, 2017].
- [Gal17] Galpshare. Galpshare, 2017. URL: <http://www.galpshare.pt/> [accessed January 27th, 2017].
- [Gmb16] Intelligent Apps GmbH. mytaxi, 2016. URL: <https://pt.mytaxi.com> [accessed January 22nd, 2017].
- [HTZ11] Khandker M Nurul Habib, Yuan Tian and Hamid Zaman. Modelling commuting mode choice with explicit consideration of carpool in the choice set formation. *Transportation*, 38(4):587–604, 2011.
- [IDC16] IDC. Smartphone OS Market Share, 2016 Q3, 2016. URL: <https://www.idc.com/promo/smartphone-market-share/os> [accessed January 22nd, 2017].
- [iS17a] iTunes Store. Blablacar - boleias, 2017. URL: <https://itunes.apple.com/pt/app/blablacar-boleias/id341329033> [accessed January 17th, 2017].
- [iS17b] iTunes Store. Cabify, 2017. URL: <https://itunes.apple.com/us/app/cabify-your-private-driver/id476087442> [accessed January 20th, 2017].
- [iS17c] iTunes Store. car2go, 2017. URL: <https://itunes.apple.com/us/app/car2go/id514921710> [accessed January 22nd, 2017].
- [iS17d] iTunes Store. Citydrive, 2017. URL: <https://itunes.apple.com/us/app/citydrive-pt/id1105388901> [accessed January 27th, 2017].
- [iS17e] iTunes Store. MEO Táxi, 2017. URL: <https://itunes.apple.com/pt/app/meo-taxi/id857873394> [accessed January 20th, 2017].
- [iS17f] iTunes Store. mytaxi, 2017. URL: <https://itunes.apple.com/app/id357852748> [accessed January 22nd, 2017].

REFERENCES

- [iS17g] iTunes Store. Pool My Ride - Carpool Rideshare, 2017. URL: <https://itunes.apple.com/br/app/pool-my-ride-carpool-rideshare/id789351056> [accessed January 21st, 2017].
- [iS17h] iTunes Store. Uber, 2017. URL: <https://itunes.apple.com/pt/app/uber/id368677368> [accessed January 18th, 2017].
- [iS17i] iTunes Store. Waze Rider - Get a Ride, 2017. URL: <https://itunes.apple.com/il/app/waze-rider-get-a-ride/id1091029104> [accessed January 22nd, 2017].
- [lar17] laravelbook. Architecture of Laravel Applications, 2017. URL: <http://laravelbook.com/laravel-architecture/> [accessed June 16th, 2017].
- [Lda16a] Lindoweb Lda. Boleia.net, 2016. URL: <http://www.boleia.net/> [accessed January 17th, 2017].
- [Lda16b] Lindoweb Lda. Dragoes.boleia.net, 2016. URL: <http://dragoes.boleia.net/> [accessed January 17th, 2017].
- [Lda16c] Lindoweb Lda. Empresas.boleia.net, 2016. URL: <http://empresas.boleia.net/> [accessed January 17th, 2017].
- [lea17] learnnc. When you ride alone, you ride with Hitler, 2017. URL: <http://www.learnnc.org/lp/multimedia/13293> [accessed January 31st, 2017].
- [lie17] liebig. Cron, 2017. Repository on GitHub. URL: <https://github.com/liebig/cron> [accessed June 16th, 2017].
- [Log14] Delve Logic. Poolmyride, 2014. URL: <https://poolmyride.com/> [accessed January 21st, 2017].
- [LPS99] Timo Lajunen, Dianne Parker and Heikki Summala. Does traffic congestion increase driver aggression? *Transportation Research Part F: Traffic Psychology and Behaviour*, 2(4):225–236, 1999.
- [Ltd17] Chitrani Technologies Pvt. Ltd. ToGoCarpool - Together We Go, 2017. URL: <http://togocarpool.com/> [accessed January 21st, 2017].
- [Mic17] Microsoft. Uber, 2017. URL: <https://www.microsoft.com/pt-pt/store/p/uber/9wzdnrcrfhxr> [accessed January 18th, 2017].
- [Mon17a] James Montemagno. Geolocator Plugin for Xamarin and Windows, 2017. Repository on GitHub. URL: <https://github.com/jamesmontemagno/GeolocatorPlugin> [accessed June 16th, 2017].
- [Mon17b] James Montemagno. PullToRefreshLayout for Xamarin.Forms, 2017. Repository on GitHub. URL: <https://github.com/jamesmontemagno/Xamarin.Forms-PullToRefreshLayout> [accessed June 16th, 2017].
- [Mon17c] James Montemagno. Settings Plugin for Xamarin And Windows, 2017. Repository on GitHub. URL: <https://github.com/jamesmontemagno/SettingsPlugin> [accessed June 16th, 2017].

REFERENCES

- [Nis17] Nish. Xamarin.Forms-Samples, 2017. Repository on GitHub. URL: <https://github.com/nishanil/Xamarin.Forms-Samples> [accessed June 16th, 2017].
- [Per14a] José Carlos Portela Pereira. Gamification on Users Daily Activities. Master's thesis, Faculdade de Engenharia da Universidade do Porto, 2014.
- [Per14b] M.H.R. Pereira. *AngularJS: Uma abordagem prática e objetiva*. Novatec Editora, 2014. URL: <https://books.google.pt/books?id=MUelBQAAQBAJ>.
- [Pip17] Pipa. uberGREEN: um passo em direção a uma mobilidade mais sustentável, 2017. URL: <https://newsroom.uber.com/portugal/ubergreen/> [accessed January 18th, 2017].
- [Por15] FC Porto. Acompanha o FC Porto À Boleia De Dragões, 2015. URL: <http://www.fcporto.pt/pt/noticias/Pages/Acompanha-o-FC-Porto-%C3%A0-boleia-de-Drag%C3%B5es.aspx> [accessed January 18th, 2017].
- [Rei11] Manuel Reis. Gestão da mobilidade: Plano de mobilidade da FEUP. Master's thesis, Faculdade de Engenharia da Universidade do Porto, 7 2011.
- [res17] restsharp. RestSharp - Simple .NET REST Client, 2017. Repository on GitHub. URL: <https://github.com/restsharp/RestSharp> [accessed June 16th, 2017].
- [Rid17] RideAmigos. Rideamigos, 2017. URL: <https://rideamigos.com/> [accessed January 22nd, 2017].
- [Rit17] Allan Ritchie. ACR User Dialogs for Xamarin and Windows, 2017. Repository on GitHub. URL: <https://github.com/aritchie/userdialogs> [accessed June 16th, 2017].
- [RKZF00] Paul Resnick, Ko Kuwabara, Richard Zeckhauser and Eric Friedman. Reputation systems. *Commun. ACM*, 43(12):45–48, December 2000. URL: <http://doi.acm.org/10.1145/355112.355122>, doi:10.1145/355112.355122.
- [Rou16] Margaret Rouse. single sign-on (SSO), 2016. URL: <http://searchsecurity.techtarget.com/definition/single-sign-on> [accessed June 16th, 2017].
- [SC07] Susan Shaheen and Adam Cohen. Growth in worldwide carsharing: An international comparison. *Transportation Research Record: Journal of the Transportation Research Board*, (1992):81–89, 2007.
- [SG08] Daniel Sperling and Deborah Gordon. Two billion cars: transforming a culture. *TR news*, (259), 2008.
- [shi17] Shibboleth, 2017. URL: <https://shibboleth.net/> [accessed January 20th, 2017].
- [Sil16] Filipa Silva. FEUP reconhece que parquímetros "penalizam" estudantes, 2016. URL: <https://jpn.up.pt/2016/07/05/feup-reconhece-parquimetros-penalizam-estudantes/> [accessed January 31st, 2017].

REFERENCES

- [Sof17] Valor Software. ngx-bootstrap, 2017. Repository on GitHub. URL: <https://github.com/valor-software/ngx-bootstrap> [accessed June 16th, 2017].
- [SRV13] Jorge Simões, Rebeca DíAz Redondo and Ana FernáNdez Vilas. A social gamification framework for a k-6 learning platform. *Computers in Human Behavior*, 29(2):345–353, 2013.
- [Sto17a] Google Play Store. Blablacar - boleias, 2017. URL: https://play.google.com/store/apps/details?id=com.comuto&hl=pt_PT [accessed January 17th, 2017].
- [Sto17b] Google Play Store. Cabify, 2017. URL: <https://play.google.com/store/apps/details?id=com.cabify.rider> [accessed January 20th, 2017].
- [Sto17c] Google Play Store. car2go, 2017. URL: <https://play.google.com/store/apps/details?id=com.car2go>.
- [Sto17d] Google Play Store. Citydrive, 2017. URL: <https://play.google.com/store/apps/details?id=com.mobiag.citydrive>.
- [Sto17e] Google Play Store. Google Maps APIs, 2017. URL: <https://developers.google.com/maps/> [accessed January 20th, 2017].
- [Sto17f] Google Play Store. MEO Táxi, 2017. URL: <https://play.google.com/store/apps/details?id=com.geolink.android.meotaxi> [accessed January 20th, 2017].
- [Sto17g] Google Play Store. mytaxi, 2017. URL: <https://play.google.com/store/apps/details?id=taxi.android.client> [accessed January 22nd, 2017].
- [Sto17h] Google Play Store. Pool My Ride - Carpool Rideshare, 2017. URL: <https://play.google.com/store/apps/details?id=com.delvelogic.carpool.Activity> [accessed January 21st, 2017].
- [Sto17i] Google Play Store. ToGo Carpool (Ride Sharing), 2017. URL: <https://play.google.com/store/apps/details?id=com.app.togo> [accessed January 21st, 2017].
- [Sto17j] Google Play Store. Uber, 2017. URL: https://play.google.com/store/apps/details?id=com.ubercab&hl=pt_PT [accessed January 18th, 2017].
- [Sto17k] Google Play Store. Waze Rider - Get a Ride, 2017. URL: <https://play.google.com/store/apps/details?id=com.ridewith> [accessed January 22nd, 2017].
- [Stu17] Student Activities. Carpool at Texas A&M University, 2017. URL: <http://carpool.tamu.edu/mvv> [accessed 2017-02-01].
- [Sum17] Web Summit. Web Summit, 2017. URL: <https://websummit.com/> [accessed January 18th, 2017].
- [Tá15] MEO Táxi. MEO Táxi - Chame o seu, 2015. URL: <https://www.meo.pt/universo-meo/mais-servicos/meo-taxi> [accessed January 20th, 2017].
- [Tea87] Roger F Teal. Carpooling: who, how and why. *Transportation Research Part A: General*, 21(3):203–214, 1987.

REFERENCES

- [Ube16] Uber. Ponha a sua empresa em movimento - com o líder global em soluções de transporte terrestre para empresas, 2016. URL: <https://www.uber.com/business/> [accessed January 18th, 2017].
- [Ube17a] Uber. A Uber Em Portugal, 2017. URL: <http://uberportugal.pt/about/portugal> [accessed January 18th, 2017].
- [Ube17b] Uber. Sobre A Uber, 2017. URL: <http://uberportugal.pt/about/uber> [accessed January 18th, 2017].
- [Ube17c] Uber. uberPOOL - Partilhe a sua viagem e poupe, 2017. URL: <https://www.uber.com/pt-PT/ride/uberpool/> [accessed January 18th, 2017].
- [UPO10] UPORTOaai, 2010. URL: https://sigarra.up.pt/up/pt/web_base.gera_pagina?P_pagina=1001917 [accessed June 16th, 2017].
- [Vin17] James Vincent. 99.6 percent of new smartphones run Android or iOS, 2017. URL: <https://www.theverge.com/2017/2/16/14634656/android-ios-market-share-blackberry-2016> [accessed May 30th, 2017].
- [Waz17] Waze. Waze Carpool, 2017. URL: <https://www.waze.com/pt-PT/carpool> [accessed January 22nd, 2017].