الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

N° Réf :……

Centre Universitaire
Mila Abdelhafid Boussouf

Institut des Sciences et Technologie — Département de Mathématiques et Informatique

Mémoire préparé en vue de l'obtention du diplôme de Master

En : Informatique

Spécialité : Sciences et Technologies de l'Information et de la Communication (STIC)

Study and development of a regular carpooling platform in Algeria

Préparé par : BOUDAD Mohammed

Devant le jury :

M. BOUBAKIR Mohammed	MAA	C.U.Abd Elhafid Boussouf	Président
M. MERABET Adil	MAA	C.U.Abd Elhafid Boussouf	Examinateur
Dr. LALOU Mohammed	MCB	C.U.Abd Elhafid Boussouf	Rapporteur

Année Universitaire : 2017/2018

Abstract

In Algeria, transport plays a crucial role in people's lives as it is the major mean of daily commute. Improving this sector will have a direct impact on daily life especially with the remarkable evolution of the road network and the automobile fleet. In fact, the road network in Algeria has been evoluted in last decades, and becomes one of the biggest and densely road networks in Africa. This evolution has caused two main issues. The first one is the large number of problems generated by the increased traffic, such as traffic congestion, energy consumption, global climate change, etc. The second one is the demand for new road transportation solutions which has significantly increased as a logical consequence of these problems. One of the efficient solutions to deal with both issues is carpooling. In fact, the good exploitation of the unoccupied seats of private cars allows traffic to be minimized which minimizes consequently fuel consumption, air pollution, road accidents, etc. In another hand, it ensures a new mean of transportation better than many existing other means such as bus and tramway. This project fits in this objective where we aim at developing an application using new Information and Communication Technology (ICT) to encourage people to share rides, which can improve road transport in our country. This will contribute in protecting human lives as well as the environment by reducing the number of private cars on the road.

Acknowledgements

First and foremost, I would like to thank Almighty God for giving me strength, knowledge, ability and opportunity to take up this project work and complete it with His grace. Second, my sincere appreciation goes to my supervisor Mohammed Lalou, whose guidance, careful reading and constructive comments was valuable. I am extremely grateful to my parents Mahfoud and Nacera for their love, prayers, caring and sacrifices for educating and preparing me for my future. I deeply thank all of my teachers. Last but not least I would like to thank my friends and classmates who gave me great encouragement to accomplish this work.

Mohammed

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The remarkable increase of motor vehicles, particularly private vehicles, throughout the last two decades all around the world has caused many issues concerning traffic congestion, energy consumption and global climate change. To fix these problems, new road transportation solutions are required. Among the proposed ones is the exploitation of the unoccupied seats of private cars, which can be achieved through the use of carpooling (also known as ride-sharing). Carpooling is a collective transport system based on a shared use of private cars, whose objective is to reduce the number of cars in use by grouping people.

There are so many benefits in sharing cars. Indeed, it has minimal incremental costs because it makes use of seats that would otherwise be empty [1]. Also, it allows people to share the cost of gas and parking, cutting the expenses of the people involved and also helping to reduce the costs with construction of new roads, road maintenance and air pollution related health costs [2].

1.1 Motivation

The motivation of this work comes from the observation that despite the advent of ubiquitous, internet-enabled smartphones and different Information and Communication Technologies (ICTs) in Algeria, there is not much contribution of these technologies in sustaining good road transportation. Also, Algeria has been rising prices of subsidized gasoline and diesel for a third straight year as part of the 2018 budget, amid government attempts to compensate for a sharp fall in oil and gas revenues [3], which has created a financial crisis. For this matter we believe that developing a platform for carpooling will have a significant contribution regarding transport sector.

Using a car to go to a shopping center, to visit relatives or friends or to get out of the city is one of the advantages of economic development; many of the costs of these trips are internalized by the owner of the vehicle, mainly in fuel prices, because these trips usually take place at times of low congestion.

Using a car to go to a city center office every day, however, generates external costs in the form of congestion and pollution doing considerable harm to society [4], and it is precisely for this second type of use that our platform can be applied.

1.2 Contribution

In this work we aim to develop a dynamic carpooling platform and put it to use in Algeria with the main objectives of (1) bringing together people with similar origins and destinations, particularly at the same time, and consequently reduce cars on the road which reduces parking demand and air pollution; (2) decreasing traffic congestion and improving accessibility; (3) reducing fuel consumption and saving money on trips; (4) reducing road accident rate and preserve people's lives.

The operating principle of the application is as follows:

- The driver choose to offer a ride, giving information about the meeting point, destination, and the date and time they will leave.
- The passengers search for rides based on the meeting point, destination, and the date and time.
- After that the system retrieves the searched rides, the passenger could send a request to the driver.
- The driver receives request, and choose which passengers they would like to have a trip with.
- The driver and passengers are brought together in the same automobile.

1.3 Outline of thesis

The structure of this thesis is as follows. In chapter 2, we present the state of the use of ICT in road transport all over the world, while focusing on the use of carpooling by giving its characteristics, its different forms of use, its benefits and barriers, etc. Chapter 3 presents the preliminary study, where we talk about road transport in Algeria; its impact on people's lives and the environment; as well as the use of ICT in Algeria's road transport. Chapter 4 details the analysis and conceptual design of our application, from actors identifications to the elaboration of the class diagram. In chapter 5, we detail the whole implementation of the application, where we present the systems architecture; different tools and programming languages used for developing the application, and lastly, we give different screenshots to show the use of the system. Chapter 6 gives a general conclusion to the thesis.

Chapter 2

Road Trasportation and ICT

2.1 Introduction

In this chapter, we will start by giving an overlook of transport and the Impact of ICT on transport. After that we will define carpooling, and will give a brief history of carpooling. Moreover we will look at some carpooling related work, then we will talk about some benefits as well as some barriers of carpooling and we finish with a conclusion.

2.2 Transport

Transport is the movement of people and goods from one place to another using different means of transportation, Transport provides better access to economic opportunities, such as jobs. It also plays a major role in social inclusion. For instance, rural roads can unlock a world of opportunities for isolated communities, while urban transport that benefits low-income communities can promote access to affordable ways to reach jobs. Types of transport include air, water, and land Transport. Transport is responsible of 23% of the world's energy-related CO2 emissions [5].

Beginning from the 20th century the world has seen a significant increase in mobility and higher levels of accessibility as trade was liberalized and technology improved significantly. resulting in a great need for capacity to manage, support and expand movements of passengers and freight [6]. Figure 2.1 shows a considerable increase of automobile production and fleet, starting from 1965 till 2016.

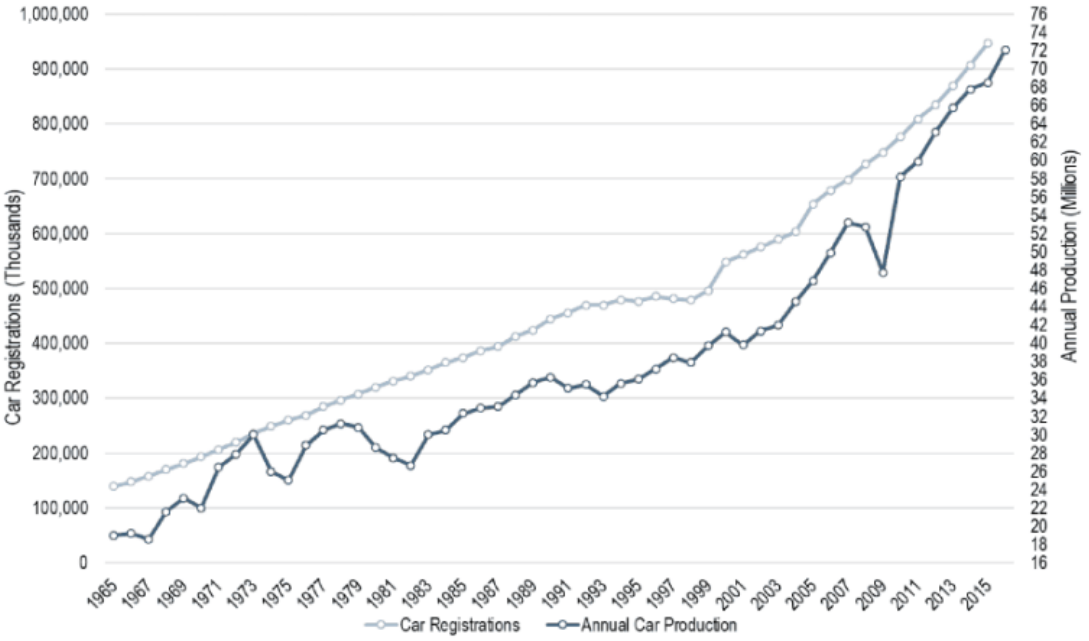


FIGURE 2.1: World Automobile Production and Fleet, 1965-2016.

2.3 Different ICT used in transport

Over the last years, there has been a wide adoption of ICT in the road transport sector all over the world, which has evident benefits in improving the transport infrastructure. In this section, we will explore some of these major innovative technologies.

2.3.1 Route finding and traffic information

One of the most innovative apps used for finding routes and obtaining information about traffic is Google Maps. It is the most popular maps app in the world, and is an extremely powerful tool for navigation. It offers a lot of great features, among these features we find route-planning options for driving, walking, cycling, public transit, and ride-booking, however some of these features are not available in all regions. In addition to Google maps, Waze which is also owned by Google gives some extra features considering that it's the largest community based traffic and navigation app. It provides real-time information of traffic backups caused by constructions or accidents, road closures not yet seen by Google Maps, dangerous potholes, dicey road conditions, and police speed traps. Another equally important app is Inrix, which gives accurate Estimated Travel Arrival (ETA), using a cloud platform called Autointelligent to calculate travel times based on current travel conditions as well as predict future conditions.[7]

2.3.2 Smart public transit systems

Smart public transport is the connected solutions for shared passenger transport services such as buses, trains and ferries. It includes applications for connected vehicles and related infrastructure, such as passenger information, ticketing and payment systems, cloud and analytics services as well as traffic management and control[8]. There is an important number of apps and websites made specifically to make access to public transport easier and convenient such as [9]:

Citymapper [10]. It is the ultimate transit app. It has support for most transportation means. That includes taxis, subways, trains, ferries, cabs, and even Uber [11] and Lyft [12]. It helps you to check nearby departure time; find the fastest route combining bus, subway, train, ferry, taxi, car share, bike share and walk; follow step by step directions, etc.

Moovit [13]. It is one of the most popular transit apps. It helps passengers to find the nearest train or bus station to their home. We can use it to find train schedules, bus schedules, subway schedules, trams and more. It also helps to figure alternative lines to get to the desired destination. At the time of this writing it supports over 2000 cities accross 80 counties.

Easy [14]. It is a taxi, car and ridesharing app, and it works rather easily. It covers currently over 400 cities in South, Central, and North America, including some cities in the Middle East region. The app finds your GPS location, and hence we can hail a ride to that location. People have the option to store their credit card information for even more features.

Transit [15]. It is a real-time transit app. It supports over 125 cities across the U.S., Canada, Australia and Europe. it gives real-time predictions for buses, subways, and similar modes of transportation. It has other features such as step-by-step navigation, bike sharing stations, and integration with Uber. It syncs up with local transportation agencies in most of the cities it supports.

Most of the above mentioned apps have their own web interface, and almost all of them use Google Maps services for navigation, as well as a dataset to hold the public transport modes schedules of most cities around the world.

2.3.3 Parking service

Besides smart public transit systems, parking services are of great importance in urban area, considering the number of vehicles on the road. Finding a parking spot can be frustrating, time-consuming and can also result in more traffic congestion. Fortunately a considerable number of apps offer parking

services which allow people to find, compare spots and save money and time whenever they want to park [16].

SpotHero [17]. It is one of the top parking apps, recommended by Forbes [18]. It has amazing benefits which make it stand out of the other apps. It helps people get everywhere, easier by making it fast and convenient to find and reserve parking in major cities. It also gives users a number of deals that ensures they can save up to 50% on the parking costs such as coupons codes and promo codes.

ParkMe [19]. It covers a significantly bigger area, available in over 84,000 locations around 64 countries. One of the best things about this app is that it will tell users where to find parking up to two weeks in advance. It is also considered as one of the most accurate apps available at the moment of this writing.

ParkWhiz [20]. It is known for offering the best deals on parking spots. It allows users to find parking spots in any major city and save up to 50% when they book in advance . Much like other parking apps, this one lets users compare various locations and prices.

In the following, we give a detailed description of carpooling, and present its different characteristics.

2.3.4 Carpooling

Carpooling consists in sharing a private car between many people for the same transit. It is one of the most important form of using ICT in transportation. Carpooling began after the introduction of Model-T, America's first automobile, and went through many ups and downs due to many factors. A research conducted by the Sightline Institute in 2008 shows that for an average car with a single driver, the amount of carbon dioxide (CO2) emitted per passenger per mile is 1.1 lbs [21].

2.4 Carpooling

In this section, we give a more detailed description for carpool service. First we give a definition for carpooling, and an overview of its development over the years. Then, we present its different forms, and its benefits and barriers. Furthermore we talk about carpooling practices across the world.

2.4.1 Definition

Carpooling does not have a single definition, and it's broadest definition as it is stated by Morency is "when two or more trips are executed simultaneously, in a single vehicle" [22]. Basically a driver chooses to offer a ride with other people in their own vehicle. We can note that sharing a ride in a private car makes the trip more comfortable and reduces the need to own a special car for regular travel, which is not always feasible, especially for individuals with low income. There are several platforms and websites that provide carpooling services such as: BlaBlaCar [23], www.carpooling.com [24], and others. The term ridesharing is sometimes used to refer to carpooling, although ridesharing is also used to mean other modes of vehicle sharing like vanpooling and sharing of a taxi.

2.4.2 History of carpooling

Carpooling is not a new concept, it dates back to the invention of private automobiles and it has been around since the beginning of the 20th century. It went through four major incidences.

The first incidence began in the U.S. after the introduction of Model-T the "jitney" craze private automobile in 1914, after the U.S. economy fell into a recession with the outbreak of World War I[25] which was intended to shadow streetcar routes. It spread rapidly in few weeks with the help of some national press. In 1915 the jitney craze became an alternative for many Americans that detested streetcars, but could not yet afford an automobile and also a challenge to streetcar companies [26], however carpooling did not take too long to see a decline due to many factors. For instance, concerns over safety and liability of the jitneys automobiles were being reported in the popular press (New York Times, 1915) [27].

The second major period that carpooling had seen a great participation was during World War II, in which the U.S. government encouraged carpooling heavily as a method to conserve resources for the war effort, by requiring that ridesharing arrangements to workplaces be made when no other alternative transport means were available, in addition to the U.S. Office of Civilian Defense asking neighborhood councils to encourage four workers to share a ride in one car to conserve rubber for the war effort [28].

The third major period where a substantial interest in carpooling has arisen in the U.S. was in the early 1970's, in response to the energy and oil crisis due to the Arabian oil embargo of 1973 to 1974. The U.S. federal government begun spending funds on carpool initiatives [27] by making High Occupancy Vehicle (HOV) lanes, which encouraged more people to carpool.

The forth major period began after the emergence of internet-enabled smartphones in the second half of 2000s, dozens of carpooling websites, applications and ridematching platforms have come into existence, which aimed to make carpooling more flexible and simplify the process of sharing rides.

2.4.3 Forms of carpooling

Carpooling can be divided into two main forms: organized and casual carpooling. In organized carpooling there is no third party involvement in arranging a meeting between the driver and passengers. Usually the driver decides to pick up passengers in order to be able to use a High Occupancy Vehicle (HOV) lane or to share the trip cost, and the crews are formed on the spot[29], also by hitchhiking, where a passenger stand on the side of the road asking for a ride from passing vehicles, usually from strangers, for free in most cases. In casual carpooling, also known as slugging, it is the other way around where a third party entity is involved. Casual carpooling involves long-distance and short-distance carpooling.

There are other forms of carpooling like dynamic carpooling, which is a service that makes use of recent technological advances such as: navigation devices to determine a driver's route and arrange the shared ride; smartphones for a traveler to request a ride from wherever they happen to be; social networks to establish trust between drivers and passengers [30].

2.4.4 Carpooling benefits and barriers

Carpooling, as we previously mentioned, has great advantages concerning:

- Reducing traffic congestion and air pollution.
- Reducing individual car costs such as fuel and parking.
- Making it easier to get a ride at the peak hours, and provides an additional form of transport for rural areas.
- Making the ride more comfortable (having a ride in a car is usually more comfortable than public transport).
- Making new acquaintances

Despite the clear benefits of carpooling, there are some barriers which discourages people to consider it as an alternative for the other means of transportation.

The main barrier to carpooling that was discussed by focus group participants was that of a perceived lack of flexibility. One participant mentioned that "getting in at the same time is fine but what about leaving at the same time" with another saying a barrier was "not being able to leave when you're ready". It was perceived that this barrier impacts "especially for people with children" and that "if you have to take a day off, you have to organize it in advance" [31].
Other common barriers to carpooling that were mentioned included:

- Personal safety ("for women" and "stranger danger").

- Not wanting to talk in the morning.
- Having to rely on others (“worrying about punctuality”).
- Already being content with using public transport (“don’t think carpooling will ever be attractive to the city because public transport is so convenient”)

2.4.5 Carpool practices across the world

Ridesharing is growing each passing year. The statistics portal Statista[32] estimated that Ridesharing user penetration worldwide is at 9.8% in 2018 and it is expected to hit 13.3% in 2022. The estimation include: online carpooling platforms that help users organize carpools (e.g. BlaBlaCar); online platforms that let users book rides offered by drivers using their private passenger cars (e.g. Uber); taxi companies that offer their services through an app(e.g. myTaxi). A considerable number of apps offer ridesharing services all around the world. The following are some of the most popular apps [33][34][35]:

Uber [11]. It is the leading app in ridesharing service. It matches passengers requesting rides with drivers based on the GPS location of the passenger’s phone. The app sends driver contact information to the passenger, and a text when the driver arrives. Uber will quote fares and allow payment with credit card. The passenger can leave a review to rate how well the ride goes.

Lyft [12]. It is the second-largest ride-hailing app by volume, offers transport in over 200 cities in the U.S. It has expanded to Toronto in 2017, marking its first expansion into the Canadian market. Lyft offers a range of vehicles to choose from, including average-sized Lyft autos and larger Lyft Plus cars.

Blablacar [23]. It is launched initialy in France. At the time of this writing it is the leading European carpooling platform. It connects regular drivers and passengers who would like to share long trips. People traveling with extra seats offer them up, and passengers then pre-pay to help with fuel and other traveling costs. Drivers are not professionals and they are not making a profit, they are just offsetting their own costs.

Gett [36]: The Israel-based ride-hailing app formerly known as GetTaxi, has a user base that pales in comparison to Uber and Lyft. It covers more than 100 cities globally. It is appreciably more reasonable when it comes to pricing, though: the service never charges a premium during busy hours.

2.4.6 Features of Dynamic Ridesharing

We use the term dynamic ridesharing to describe automated systems that facilitate the process of sharing one-time trips between drivers and passengers, which are close to their desired departure times. We characterize this concept by the following features [37]:

Dynamic. The ride-share can be established on short-notice, which can range from a few minutes to a few hours before departure time. The growing use of Internet-enabled mobile phones allows people to offer and request trips whenever they want, wherever they are. Thus, communication technology is a key enabler to dynamic, on-demand ride-sharing.

Independent. The drivers which provide the rides are independent private entities. This is different from most traditional forms of passenger transportation where a central organization owns vehicles and/or employs drivers. Cost-sharing The variable trip-related costs are reallocated among the ride-share participants in a way that makes it beneficial for them to participate from the perspective of cost reduction. The variable trip cost minimally includes fuel expense, but may also take into account wear and tear on vehicles, parking costs or road fees such as tolls.

Non-recurring. trips Dynamic ride-sharing focuses on single, non-recurring trips. This distinguishes it from traditional carpooling or vanpooling, both of which require a long-term commitment among two or more people to travel together on recurring trips for a particular purpose, often for traveling to work. Single-trip ride-sharing is more flexible because it does not require rigid time schedules or itineraries over time.

Prearranged. The trips are prearranged which means that the participants agree to share a ride in advance, typically while they are not yet at the same location. This is different from the spontaneous, so-called casual ride-sharing in which riders and drivers establish a ride-share on the spot, similar to hitch-hiking or hailing a taxi on the side of the street. In casual ride-sharing, drivers and riders line up at established locations to share rides to other established locations to take advantage of high occupancy vehicle lane time-savings or toll savings. The main limitation of casual ride-sharing is the inflexibility of its routes, which does not allow door-to-door transportation.

Automated matching. To establish ride-shares in a way that requires minimal effort from the participants, ride matching should be automated in a dynamic setting. This means that a system matches up riders and drivers and communicates the matches to the participants. We do not include in our definition simple (online) notice boards where riders and drivers can post desired or planned trips and choose to contact potential ride-share partners themselves.

2.5 Conclusion

As pointed out in the previous sections, there are already numerous smart and innovative technologies in road transport around the world. These technologies have made a considerable contribution to make travel as easy as never before, particularly carpooling for its known benefits in reducing traffic congestion and air pollution.

Chapter 3

Case study and project presentation

In this chapter, we will study the existing situation of road transport in Algeria. We will determine, in one hand, how the road transport in Algeria has been evoluted in last decades, and in another hand, the level of using ICT in road transportation. This constitutes the project preliminary study, based on which the decision to develop such an application has been made.

First we give different statistics and information about the Algeria's road network. Then, we present different impacts of road transport on environment, namely air pollution and scarcity of natural resources consumption, as well as on human life, namely accident and damages to human lives caused by road traffic, costs of fuel consumption, etc. We close up the chapter by talking about different forms of using ICT in the Algerian road transport.

3.1 Road transport in Algeria

Road transport is the major mean of daily commute in Algeria, be it urban or rural areas. Road transport can be divided into two groups: transportation of goods and transportation of people. In this work, we are interested in people transportation.

Algeria has one of the more biggest and densely road networks in Africa of more than 118,306 km of paved roads [38]. This network will be complemented by significant portions in future projects planed by the Algerian government. Figure 3.1 shows the current road network in Algeria.

In what follows, we give general information about the Algeria's road network, namely: different kind of roads existing in this country, the road network capacity and different means of road transport.

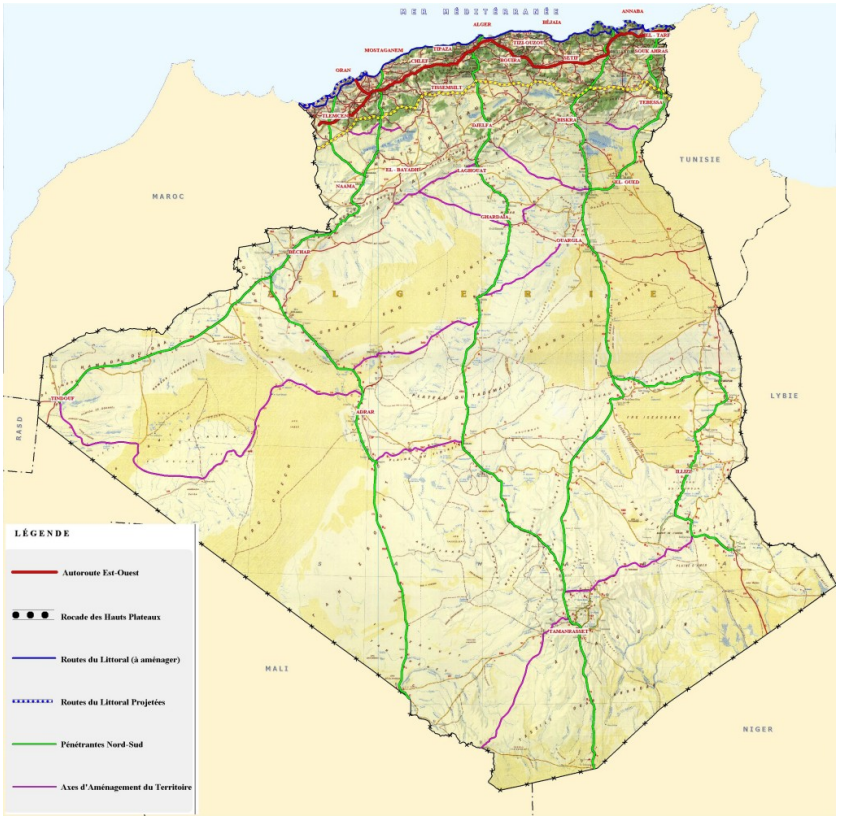


FIGURE 3.1: The Algerian road network.

3.1.1 Road classification

Roads in the Algerian transport network are divided into three main categories, namely: departmental (or national) roads, provincial roads and municipal roads.

- **Departmental roads.** which are roads connecting different cities (Wilayas) of the country. The total length of departmental roads in the whole country is nearly 30 thousand km.
- **Provincial roads.** which are roads connecting different provinces (Daïras) inside the same wilaya. The total length of provincial roads in the whole country is 26,626 km.
- **Municipal roads.** which are roads connecting different agglomerations in the same municipality. The total length of provincial roads in the whole country is 62,100 km.

In addition to these roads, there is the East-West highway, which is the most important road at the moment. It is a 1,216 km linking Annaba at the east to Tlemcen at the west. The project is achieved in 2014 and it is currently operational. Figure 3.2 shows different road categories in the wilaya of Mila.

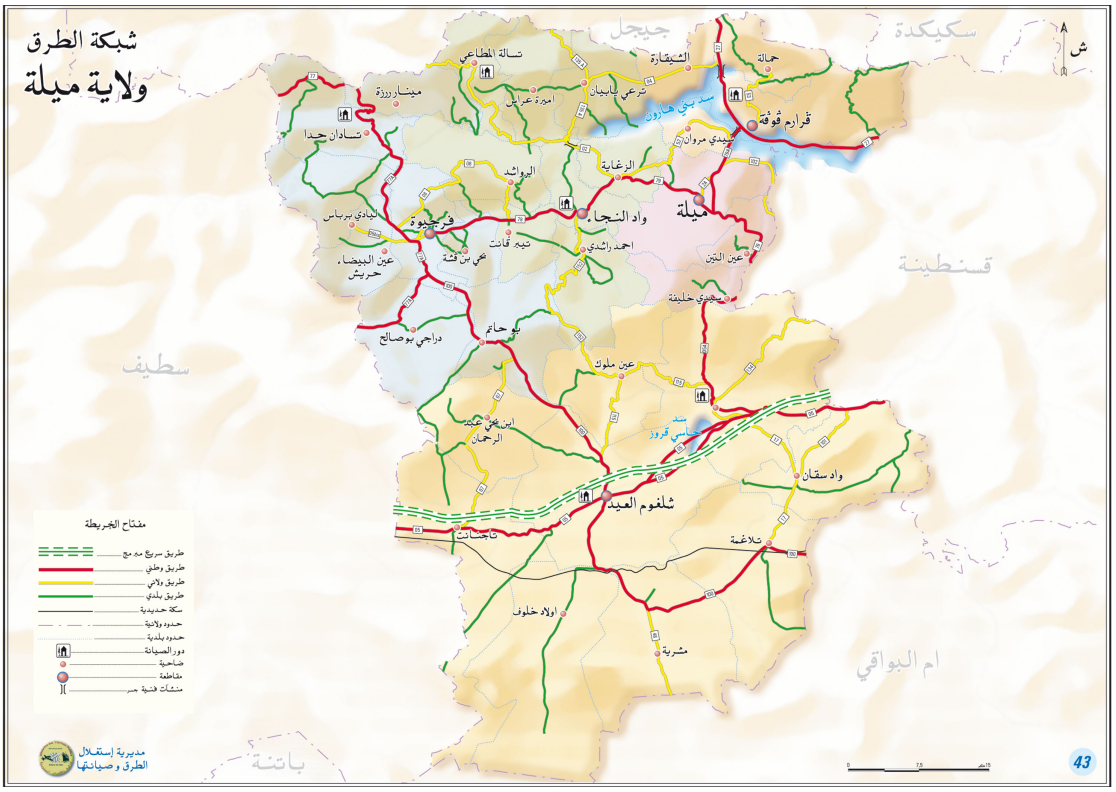


FIGURE 3.2: Different roads categories in the wilaya of Mila.

3.1.2 Road network capacity and future perspectives

The Algeria's road network is constantly growing through the government modernization program of roads and rails of transport. As future projects for expanding the road network, there are:

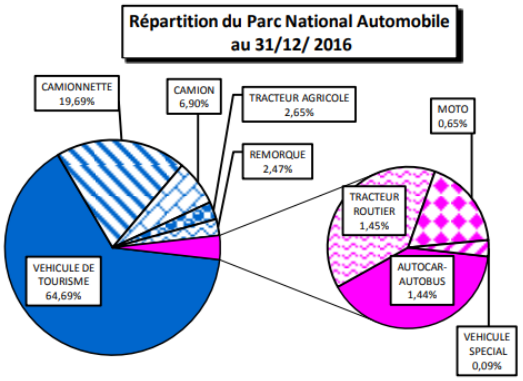
- The Highlands Highway it is of 1,330 km, and it is currently in progress and will be completed in 2015/2025. that aims at opening and linking cities located in the high plateaus regions at the interior of the country.
- The Trans-Saharan road (or the north-south road). It is a road crossing the country from north to south. This road is prompted by the government to increase trade between the six countries crossed by the Trans-Saharan road (Algeria, Mali, Niger, Nigeria, Chad and Tunisia). The project consists of three parts:
  - The west part: From El Aricha to Tiaret, via Saida for 305 km.
  - The center: Going from Tiaret to Batna, passing by M'sila for 495 km.
  - The est: Going from Batna to Tebessa, passing by Khenchela for 220 km.

The expansion of the road network in order to respond to the increased demand leads to more and more traffic, and hence using advanced and innovative methods to reduce the traffic congestion become a necessity. Adopting carpooling can be an efficient way to fix the problem.

FIGURE 3.3: Different roads categories in the wilaya of Mila

3.1.3 Means of road transport

Several modern transportation means used to transport persons as well as goods, such as high-speed train, subway train, tramway, aerial tramway, bus, taxi, etc. However, cars remain the most used mean as we can see in Figure 3.4. Established from statistics of the National Office of Statistics [39].



GENRE DU VEHICULE	Nombre	%
VEHICULE DE TOURISME	3 872 709	**64,69**
CAMION	413 261	**6,90**
CAMIONNETTE	1 178 745	**19,69**
AUTOCAR-AUTOBUS	86 200	**1,44**
TRACTEUR ROUTIER	82 832	**1,38**
TRACTEUR AGRICOLE	158 855	**2,65**
VEHICULE SPECIAL	7 033	**0,12**
REMORQUE	147 613	**2,47**
MOTO	38 933	**0,65**
TOTAL	**5 986 181**	**100**

FIGURE 3.4: Repartition of different means of road transportation in Algeria.

According to the same source, the number of cars is so important and the national car fleet is in progress as we can see in Figure 3.5. The diagram shows an almost monotonous growth in the number of cars between 2005 and 2016.

A constant growth in the national car fleet involves obviously an increase in traffic and hence the impact of this on people's daily lives, namely pollution, bottlenecks, cost, etc. Therefore, adopting new methods to deal with this situation are needed.

Dealing with the increase of traffic may be through reducing and/or managing it. Carpooling has been proven to be effective in reducing traffic congestion and make it easy to manage, especially if the growth is mainly in private vehicles, which is the case in Algeria as we can be seen in Figure 3.6. In fact, carpooling aims at reducing the use of private cars and hence more the traffic is caused by private cars more the carpooling gives good result.

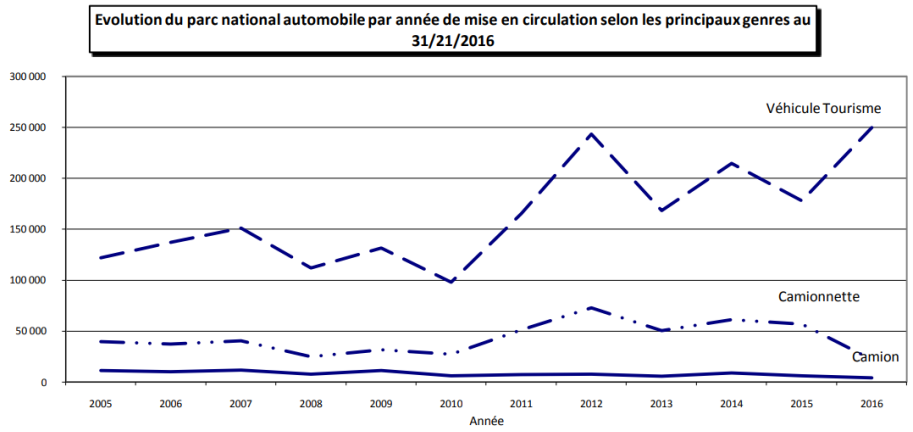


FIGURE 3.5: Evolution of the national automobile fleet from 2005 to 2016.

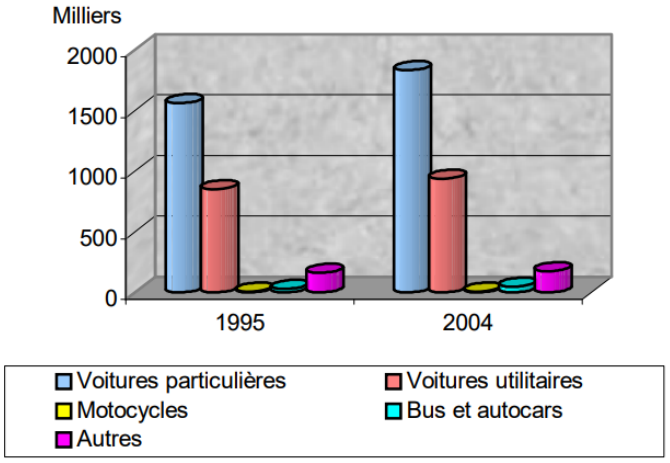


FIGURE 3.6: Evolution of the national automobile fleet by genre (private, public, bus, etc.).

3.2 Human and environmental impacts of road transport

The environmental impacts of road transport are numerous, and they can be direct such as the impact on human safety and social welfare (bottlenecks, road accidents, cost, etc.), or indirect such as the environmental impact (pollution, resource depletion, etc.). In this section, we present different impacts of road transport in people's lives in Algeria.

3.2.1 Human security

According to the World Health Organization report, road accidents are the main cause of death among young people between the ages of 15 and 29 in the world. Road crashes cause more than 1.3 million deaths every year, where 90% of them are in developing countries. In this pattern, road accidents will kill 2.4 million people in 2030 and will become, according to the WHO projections, the seventh cause of death by 2030 [40].

As for Algeria the number of car accidents increases each year and remains horrific as showed through statistics of the National Center for Prevention and Security of Roads. For instance, in 2011, 41,467 traffic accidents have been recorded, leaving 4,598 deaths and 66,361 wounded; 114 incidents, 13 deaths and 182 wounded per day. In comparison to the previous year (2010), the number of accidents has increased by 8594 accidents, i.e. + 14,26%, and the number of deaths by 938, i.e. 63.25%, while the number of injured increased by 13,926, i.e. 56.26% [41].

Figure 3.7 shows statistics of the province (namely the wilaya of Djelfa) ranked in the eighth position in 2014 with regard to the number of accidents. We see obviously that the number of car accidents, deaths and injured persons left by these accidents is in a steady increase from 2004 to 2014.

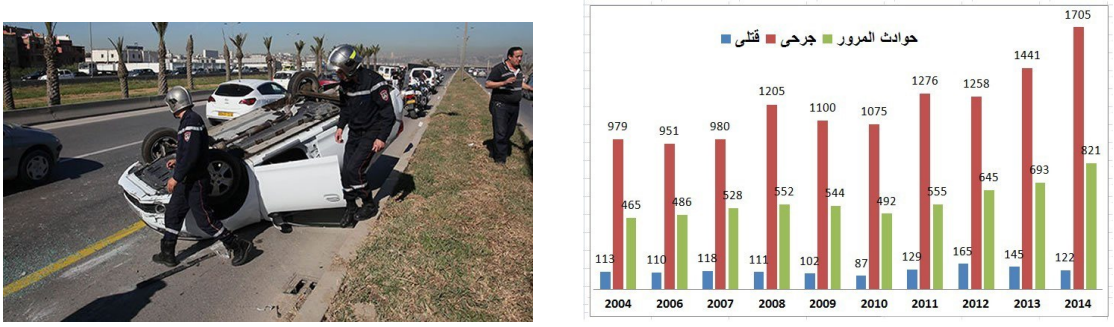


FIGURE 3.7: Daily bottling especially at the entrance to big cities which constitute bottlenecks.

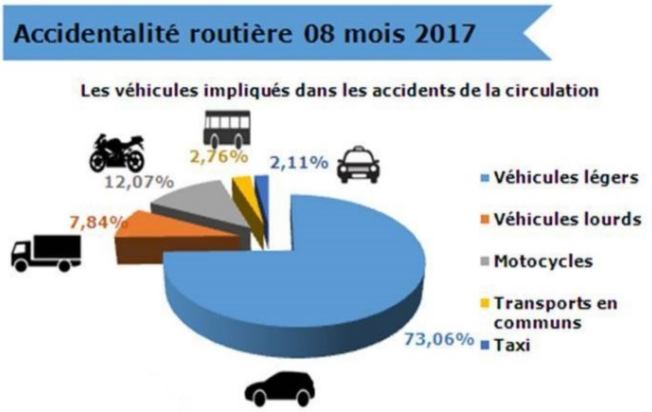


FIGURE 3.8: Daily bottling especially at the entrance to big cities which constitute bottlenecks.

Statistics in Figure 3.8 show that people transportation remains the main cause of traffic accidents with rate of more than 88%, where more than 73% of accidents are caused by tourist cars. Adopting carpooling as a solution can significantly reduce the use of private cars, in such a way that it ensures transport of persons in groups.

3.2.2 Road traffic congestion

Traffic congestion is one of the main issues in urban areas all over the world, causing several daily problems such as fuel overconsumption, delayed work, increased stress, great air pollution, etc. Among

the most congested cities in Algeria are: Alger, Bejaia, Tizi Ouzou, Blida, Oran, Constantine, Annaba. A study on the city of Algiers, with a population and a fleet ten times higher and the result is a 13,772 ton/d/veh value of CO2 release [42]. Substantial efforts have been made in order to reduce it by paving new roads, optimizing traffic-light management, improving bus services, etc. However, the problem continue to remain resulting in additional travel time and fuel wastage. Innovative solutions such as carpooling that aims to reduce the number of vehicles in the road has proven to be effective. Figure 3.9 shows traffic congestion caused in the entrance to cities.

FIGURE 3.9: Daily bottling especially at the entrance to big cities which constitute bottlenecks.

3.2.3 Atmospheric pollution

Air is one of the essential elements to live in this world. It is permanently consumed by all living beings. Now, it suffers from a progressive degradation caused serious life threatening problems such as global warming, greenhouse effect, etc. In fact, environmental resources (air, water, etc.) are mainly influenced by polluant gases such as SO2 (Oxides of sulfur), NOX (Nitrogen oxides), CO (Carbon monoxide), etc. According to Global Carbone Project statistics [43], CO2 emissions in the whole world is 36.4 billion kg in 2016, which indicates a really stresful situation. These emissions increase year by year as shown in Figure 3.10.

There are several sources of pollutant gases emissions, such as housing (heating and electricity), energy industry, Manufacturing industries and construction, etc. One of the major source of pollutant gases is transport and fuel consumption.

Indeed, for instance total CO2 emissions in 2000 were 25 billion tonnes, the energy production represented 37% of this total, industry 20%, transport 27%, housing and agriculture 14%. The 27% emissions of the transport sector are mainly due to the road which represented 20% of emissions, including 12% for passenger cars, 3% for light commercial vehicles and 5% for heavy goods vehicles [44].

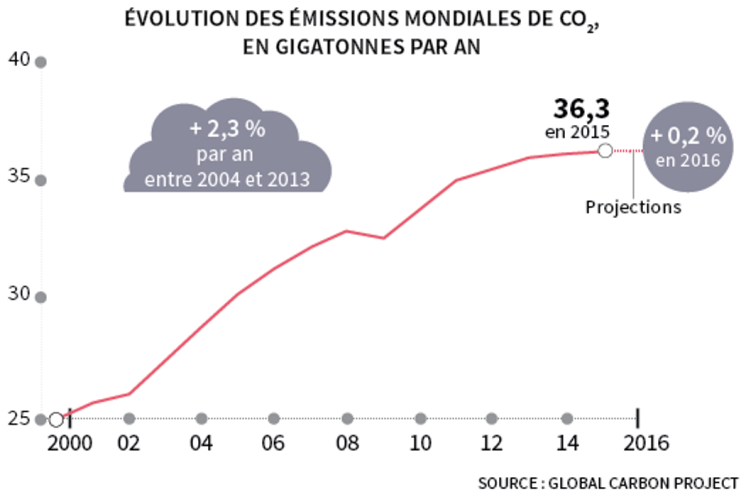


FIGURE 3.10: Evolution of global CO2 emissions in the world between 2000 and 2016.

As for Algeria car traffic represents a major portion of total pollutant gas emissions. In fact, we can see in Table 3.11 that the atmospheric pollution inside cities is essentially caused by road traffic. For example, between 1995 and 2001 there is a 63% increase in nitrogen oxide emissions caused by cars traffic.

Unité : 1000 tonne

	SO2	NOX		PM		COVNM		CO		Pb	
Secteur d'activité	**1995**	**1995**	**2001**	**1995**	**2001**	**1995**	**2001**	**1995**	**2001**	**1995**	**2001**
Energie											
Industries du secteur de l'énergie	0,05	35,83		0,46		1,02		4,49			
Industries manufacturières et construction [1]	44,80	17,85		1 128,23		13,04		39,53			
Transports routiers	4,16	123,63	201,7	8,71	15,79	249,60	135,00	996,90	904,20	1,04	0,73
Déchets											
Décharges de déchets solides	0,21	1,22		3,32		8,95		17,44			
Total des émissions	**49,21**	**178,53**		**1 140,72**		**272,62**		**1 058,36**			

Source : Rapport sur l'état et l'avenir de l'environnement 2000 et 2003 / MATE

FIGURE 3.11: Evolution of pollutant emissions in the air by sector of activity.

3.2.4 Fuel consumption

Transportation has a great importance in modern society, by allowing people to travel across cities, countries or even continents to obtain services. Also it helps to ship goods from one place to another, but this always comes at a cost. According to National Energy Balance Sheet [45], transport accounts for 35.1% of the total national consumption, with 14,293 Tonne of Oil Equivalent (TOE) for road transport and 482 TOE for aerial transport, as we can see in Figure 3.12.

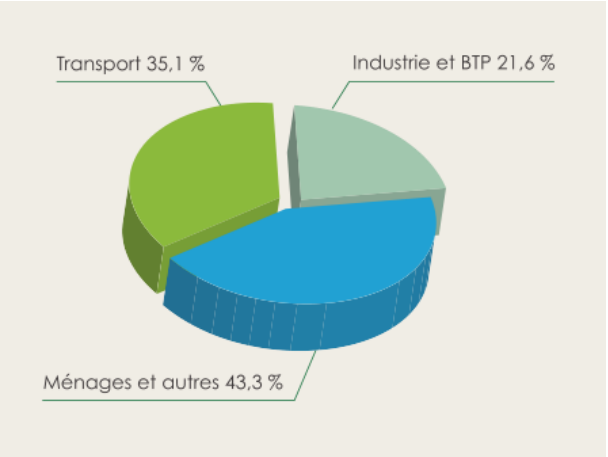


FIGURE 3.12: Consumption by activity area.

3.3 Algeria's road transport and ICT

In this section, We will give an overview of Algeria's ICT infrastructure, in which we cite some of the existing apps and websites regarding road transport sector. Afterwards, we present our project work.

3.3.1 ICT infrastucture

Over the last decade there has been a noticeable improvement in Algeria's ICT infrastructure, notably the launching of 4G network with the main service provider Algérie Télécom. The number of smartphones, tablets and other internet-enabled devices is in a rapid increase and it has became a necessity in our daily lives, offering countless apps which cover almost every area of our lives. Table 3.13 shows that Internet penetration is at 44.2% with 18,580,000 users for December 2017 [46].

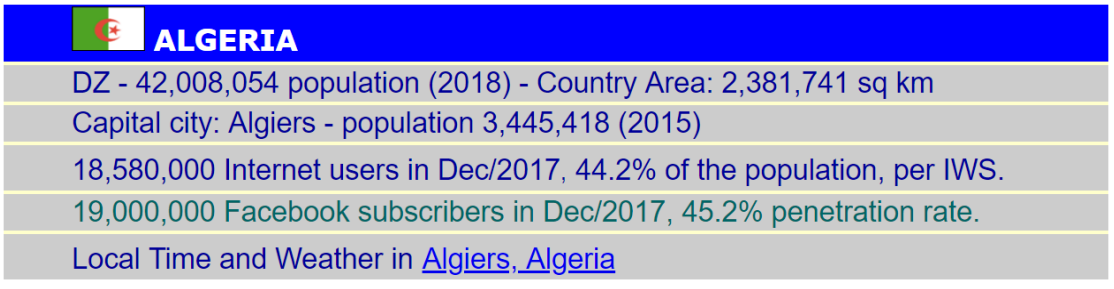


FIGURE 3.13: Internet penetration for December 2017.

According to Statista [47], the number of mobile cellular subscription is increasing rapidly from 2000 to 2016 as it shown in 3.14 .

The evolution of ICT infrastructure is an encouraging factor for developing solutions with new technologies to fix transport problems. In fact, solutions using new technologies require mobile devices

to run the applications and a continuous Internet service to ensure the system operation, and this is provided by means of the current infrastructure.

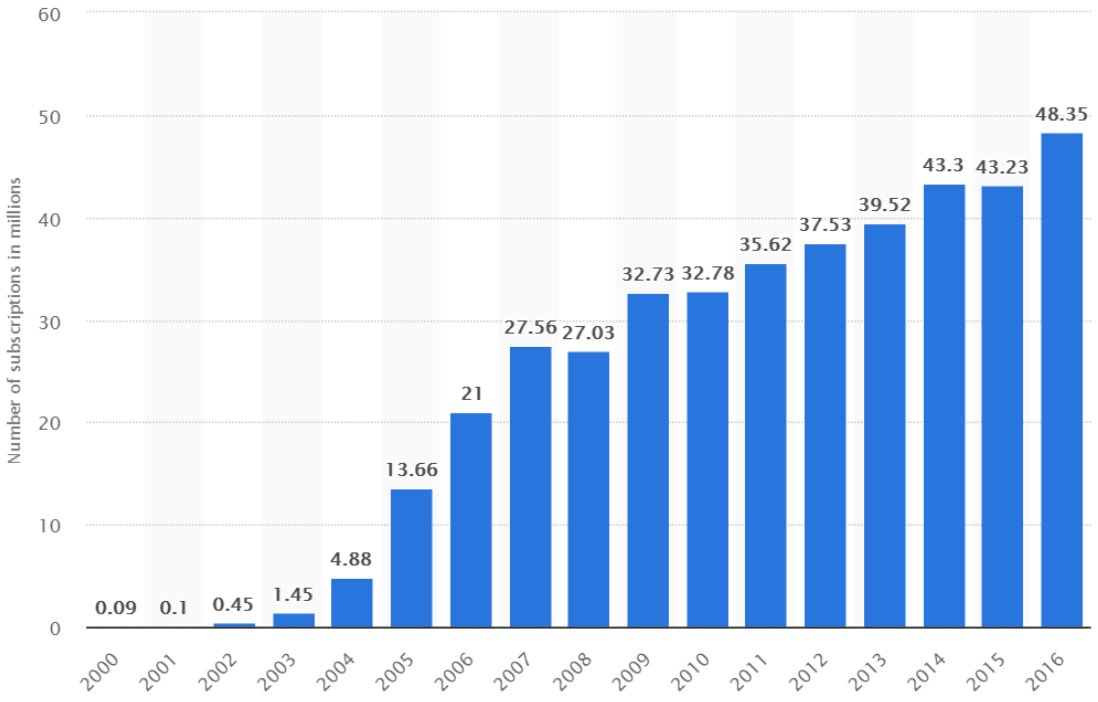


FIGURE 3.14: Number of mobile cellular subscriptions in Algeria from 2000 to 2016 (in millions).

3.3.2 Mobile applications for road transport

Algeria has a small number of mobile apps and websites dedicated for transport sector, and which operate locally. Some of them are available only in big cities like Algiers, Oran. The following are the well-known ones:

- **Yassir.** Created by a group of young Algerian entrepreneurs. It is one of the successful apps in transport sector. Yassir works similar to the giant American Uber app, by offering ride-share service. The app allows people to see the number and location of taxi drivers close to them.
- **Tariki.** An application and a website developed by the National Gendarmerie for navigation and information on road traffic. Tariki allows people to be always updated by the road condition.
- **TemTem.** It is a car hailing app that operates only in the wilaya of Algiers and Oran at the moment of this writing. TemTem allows people to book a ride in advance, and track the driver using GPS on the driver's phone.

3.4 Project presentation

The focus of our work is to allow people with similar origins and destinations to share their journeys so that more passengers travel in a car, which will help to: reduce cars on the road; decrease traffic congestion and improve accessibility; reduce parking demand and air pollution; reduce fuel consumption and save money on trips.

This work is about a carpooling system which consists of a mobile application that is going to be used by both passengers and drivers, and a web-based application for administration, with the aim of matching passengers with drivers in a seamless and fast way.

3.5 Conclusion

From the situation of road transport in Algeria, we can see the necessity of using ICT to address different transport challenges, especially that transport presents the most important axe in daily life. Thus, we believe that developing a carpooling application will help a lot at improving this sector, and will also have a good impact in the way the road transport is used. This analysis allowed us to identify functional and operational requirements, which helped us in elaborating the application design presented in the next chapter.

Chapter 4

Analysis and Conceptual design

4.1 Introduction

In this chapter, we will present the most important phase in our project, namely the application design. First, we determine the functional requirements of the system and describe the behavior of our application. In doing so, we will use the Unified Modeling Language (UML [48]) as a modeling language, while presenting different needed diagrams. We will end the chapter by determining some technical needs for developing our application, namely the logical architecture needed for the system, and the database model used for developing the application.

4.2 Actor identification and functional requirements

In software and systems engineering, a use case describes the interaction between an actor and the system in order to accomplish a particular goal, which can be modeled using UML. A use case is make up of three elements:

- **Actor:** The actor is an entity that interacts with the system through the use cases. The actor can be human or non-human.
- **System:** The system is the required procedures for accomplishing the final goal.
- **Goal:** The goal is the actor's aim or desired result.

4.2.1 System actors

After a thorough analysis of the system we ended up with five main actors for our system's use case. The actors of this system are as follows:

- **Visitor:** The visitor is any person who can download and install the application. They can create an account.
- **User:** The user can rate as well as manage their account. these use cases need the user to be authenticated.
- **Passenger:** The passenger can search for rides; request rides; manage requested rides as well as inherit the user use cases. These use cases need the passenger to be authenticated, so above all the passenger has to sign in to be able to use different available features of the system.
- **Driver:** The driver most importantly, can post rides; confirm requests; manage posted rides; view and edit his profile as well as inherit the user use cases. The driver needs to be authenticated to work correctly.
- **Administrator:** The system administrator is in charge of the management of both user accounts (View, search, deactivate, activate and delete users), and rides (Search, view and delete rides). They can also manage their account and view some statistic about users and rides. The administrator is needed to make a centralized control ensuring a secured functioning of the system.

4.2.2 Use case diagram

The use case diagram of our system is presented in the following figure.

In what follows, we give a detailed description for each use case. We start by the use cases of the visitor, then the use cases of the user, then the use cases of the passenger and those of the driver, and finally the use cases of the administrator. For each use case, we give a text description, a sequence diagram and an activity diagram.

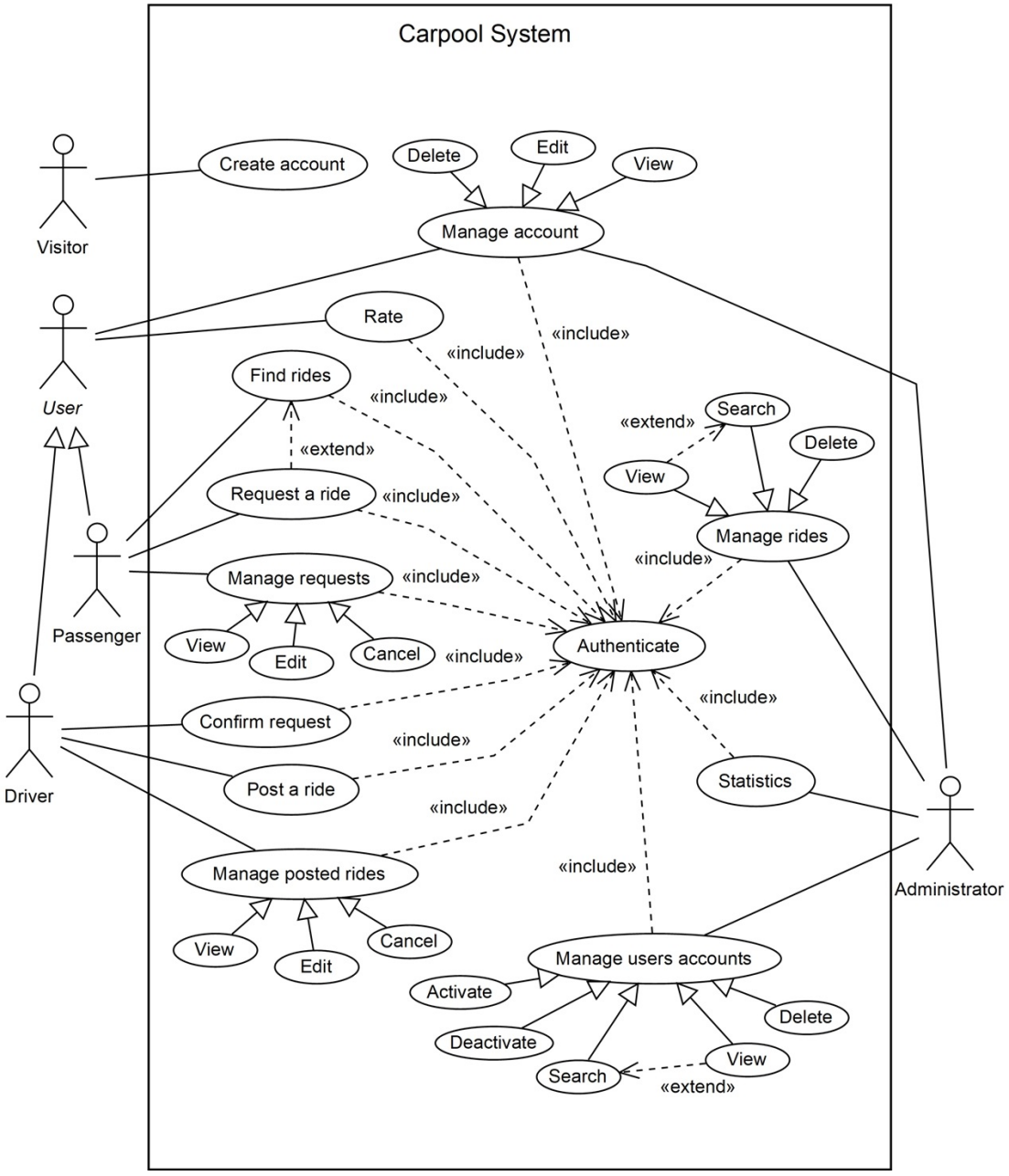


FIGURE 4.1: Use case diagram.

4.2.2.1 Visitor use cases description

Name	**Create account**
Brief Description	The ”Create account” use case allows the user to create a login and become a registered user.
Actors	Visitor
Preconditions	None.
Basic Flow	1. The visitor selects ”Create account”. 2. The system displays a form to fill. 3. The visitor fills in the necessary information. 4. The system verifies the information. 5. The system creates a new account. 6. The system notifies the new user of the creation of the account.
Alternate Flows	4. The visitor did not fill all or some required fields, or entered inaccurate information. 5. The visitor corrects the errors. 6. The system restarts from process (4) after the information had been corrected by the visitor.
Postconditions	The user entered data is stored in the user’s account. Confirmation is sent to the appropriate email address.

TABLE 4.1: Use case description for ”Create account”.

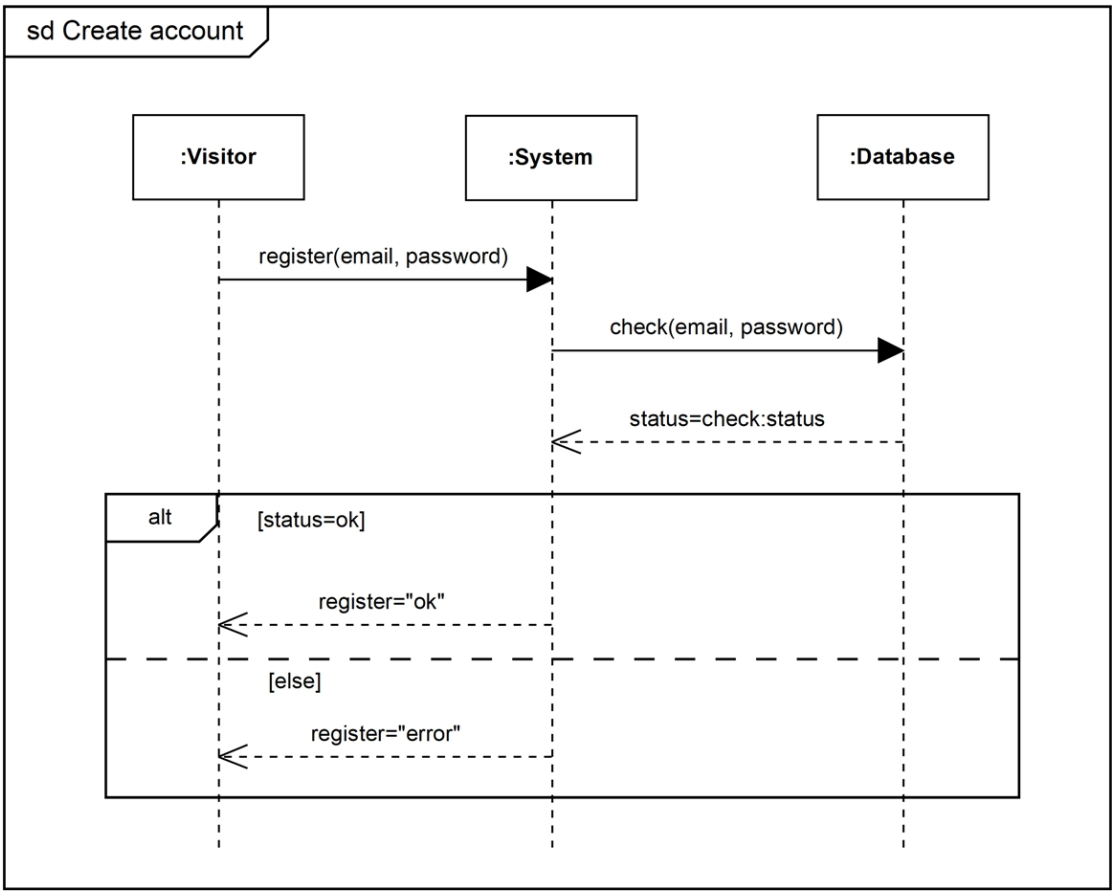


FIGURE 4.2: Sequence diagram for ”Create account”.

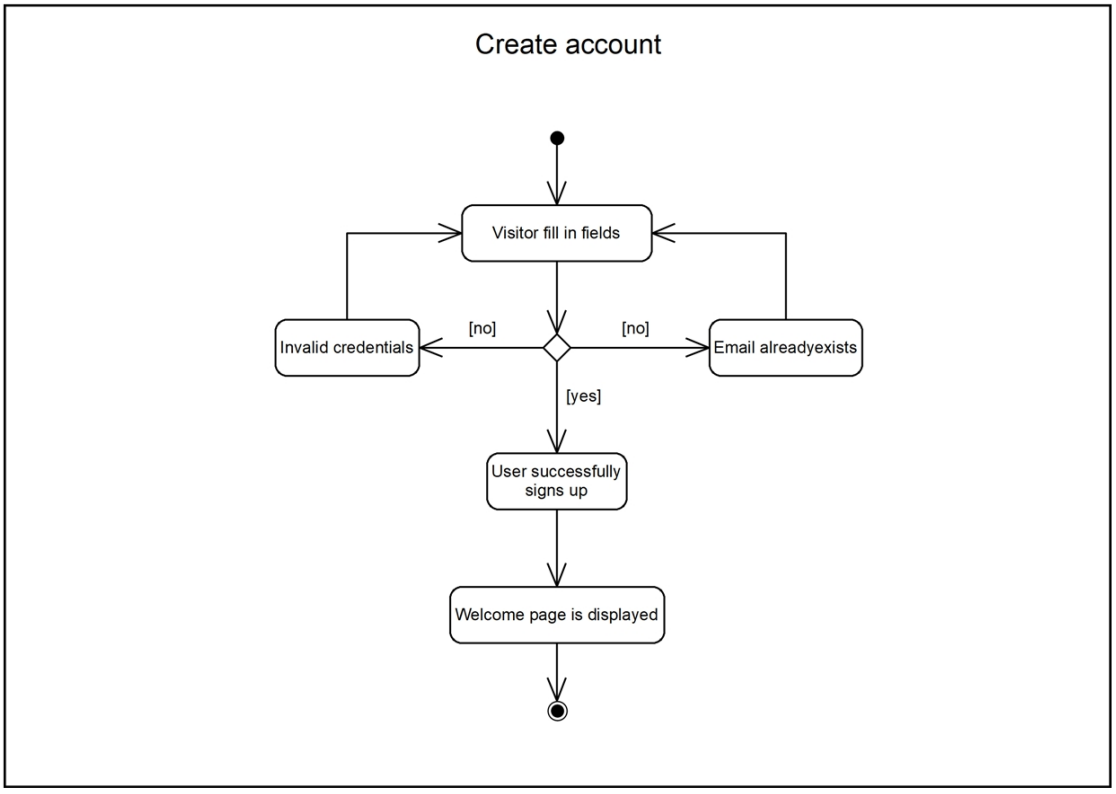


FIGURE 4.3: Activity diagram for ”Create account”.

4.2.2.2 User use cases description

Name	**Authenticate**
Brief Description	The ”Authenticate” use case allows the user to log into the application.
Actors	User, Administrator
Preconditions	None.
Basic Flow	1. The user fills in the necessary information. 2. The system verifies the information. 3. The system logs the user in.
Alternate Flows	2. The user did not fill all or some required fields, or entered inaccurate information. 3. The system notifies the user about incorrect fields. 4. The user corrects the invalid information. 5. The system restarts from process (2) after the user corrects the information. 6. The system dismisses login attempts after the user reaches 5 times.
Postconditions	The user is authenticated and the system displays all features available for the role that the user is associated with as defined in their user’s account.

TABLE 4.2: Use case description for ”Authenticate”.

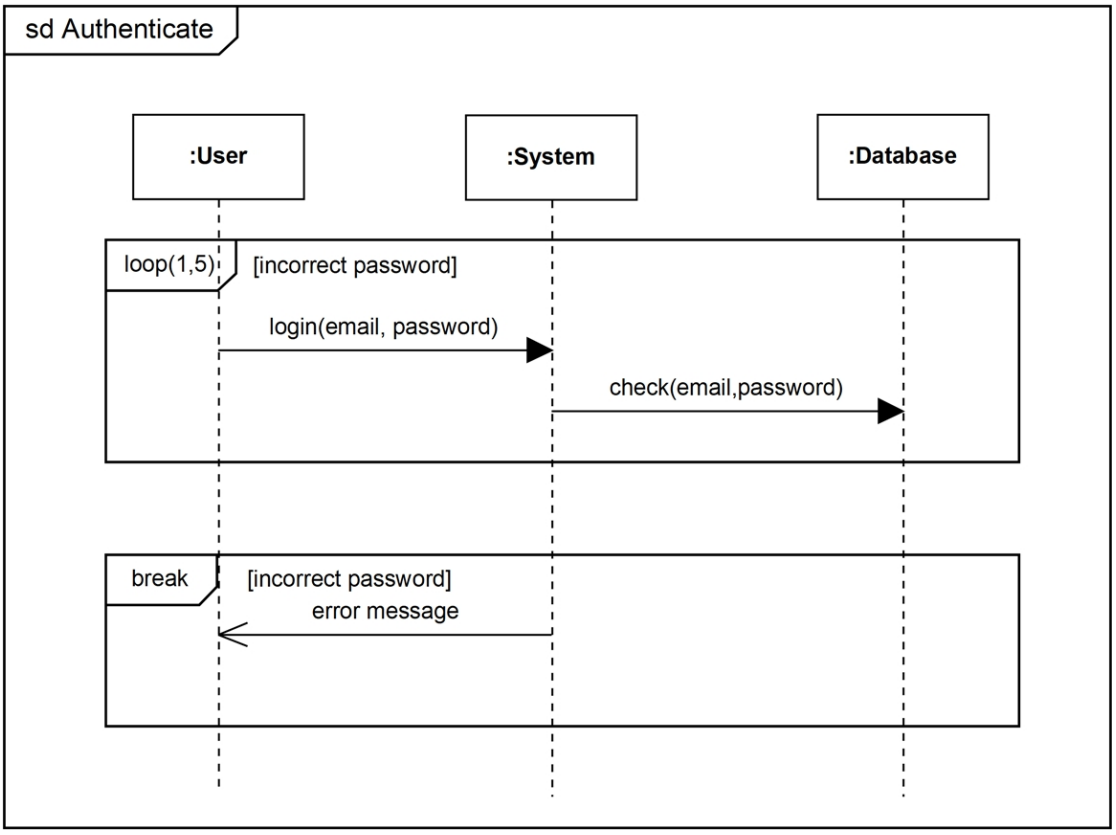


FIGURE 4.4: Sequence diagram for ”Authenticate”.

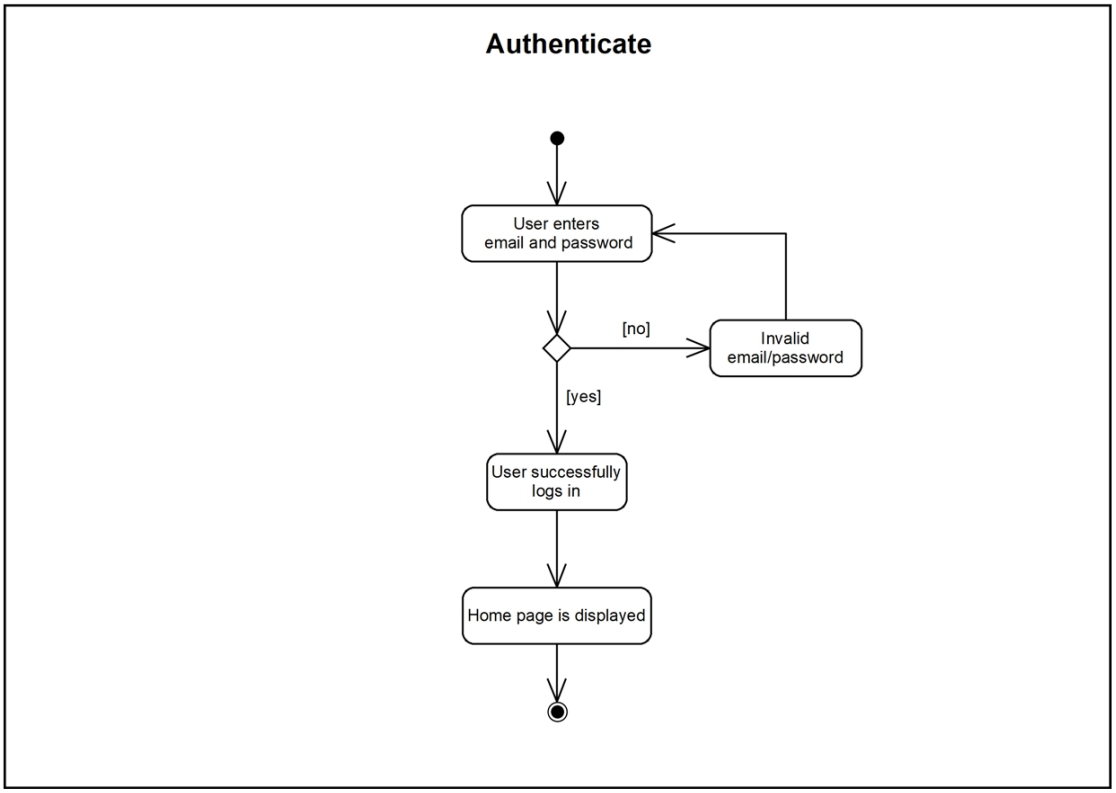


FIGURE 4.5: Activity diagram for ”Authenticate”.

Name	**Manage account**
Brief Description	The ”Manage account” use case allows the user to view, edit and delete accounts.
Actors	User, Administrator
Preconditions	The user has to be authenticated.
Basic Flow	1. The system displays the following options: • View account. • Edit account. • Delete account.
Alternate Flows	None
Postconditions	The changes made to requests are saved to the user’s account.

TABLE 4.3: Use case description for ”Manage account”.

Name	**Manage account - view**
Brief Description	The ”Manage account - view” use case allows the user to view their account.
Actors	User, Administrator
Preconditions	The user has to be authenticated.
Basic Flow	1. The user asks to view their account. 2. The system displays the account.
Alternate Flows	None
Postconditions	The user viewed their account.

TABLE 4.4: Use case description for ”Manage account - view”.

Name	**Manage account - edit**
Brief Description	The ”Manage account - edit” use case allows the user to edit their account.
Actors	User, Administrator
Preconditions	The user has to be authenticated.
Basic Flow	1. The system displays all account details. 2. The user makes the needed changes and asks the system to save them. 3. The system saves the changes and notifies the user about the updates.
Alternate Flows	None
Postconditions	The changes made to the account are saved to the user’s account.

TABLE 4.5: Use case description for ”Manage account - edit”.

Name	**Manage account - delete**
Brief Description	The ”Manage account - delete” use case allows the user to delete their account.
Actors	User, Administrator
Preconditions	The user has to be authenticated.
Basic Flow	1. The user asks to delete their account. 2. The system deletes the account and notifies the user about the account removal.
Alternate Flows	None
Postconditions	The user’s account is removed.

TABLE 4.6: Use case description for ”Manage account - delete”.

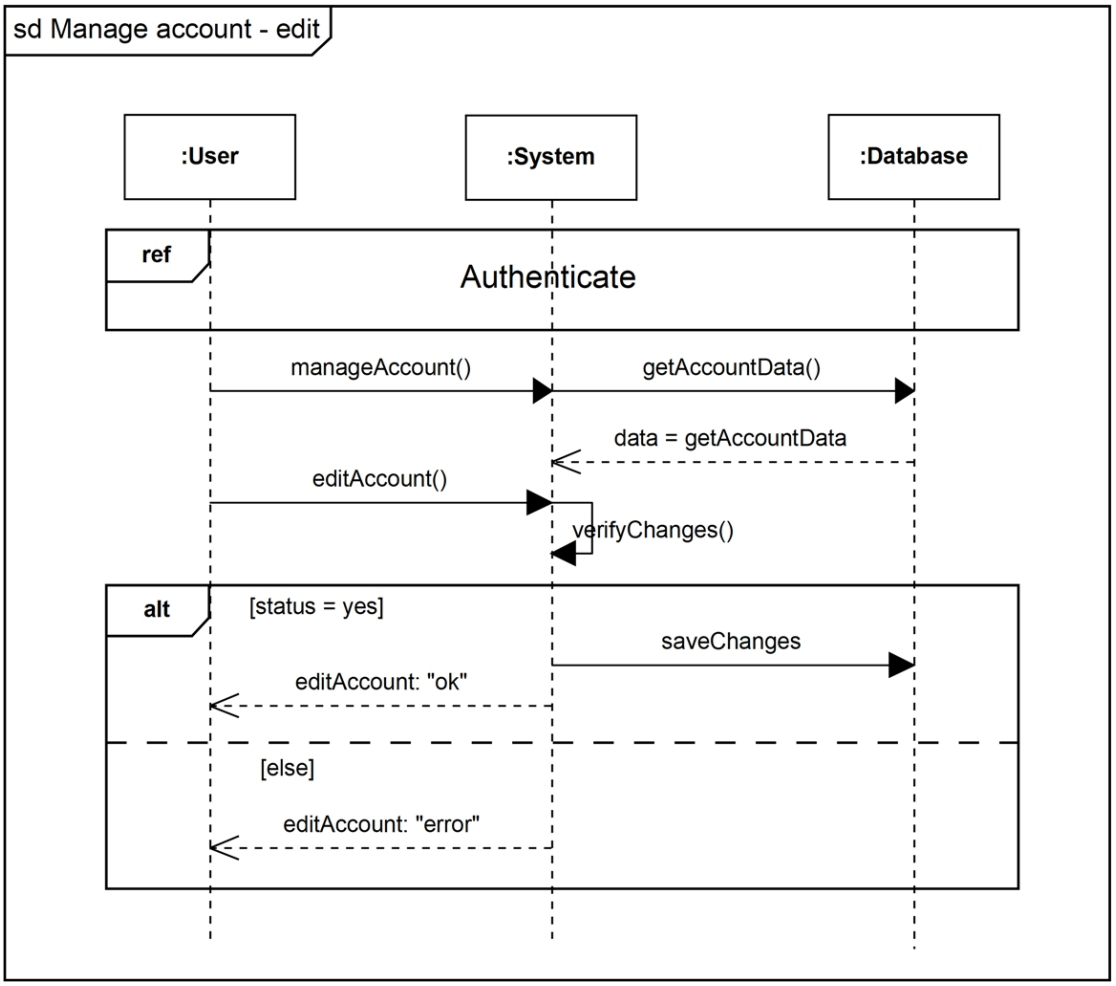


FIGURE 4.6: Sequence diagram for ”Manage account - edit”.

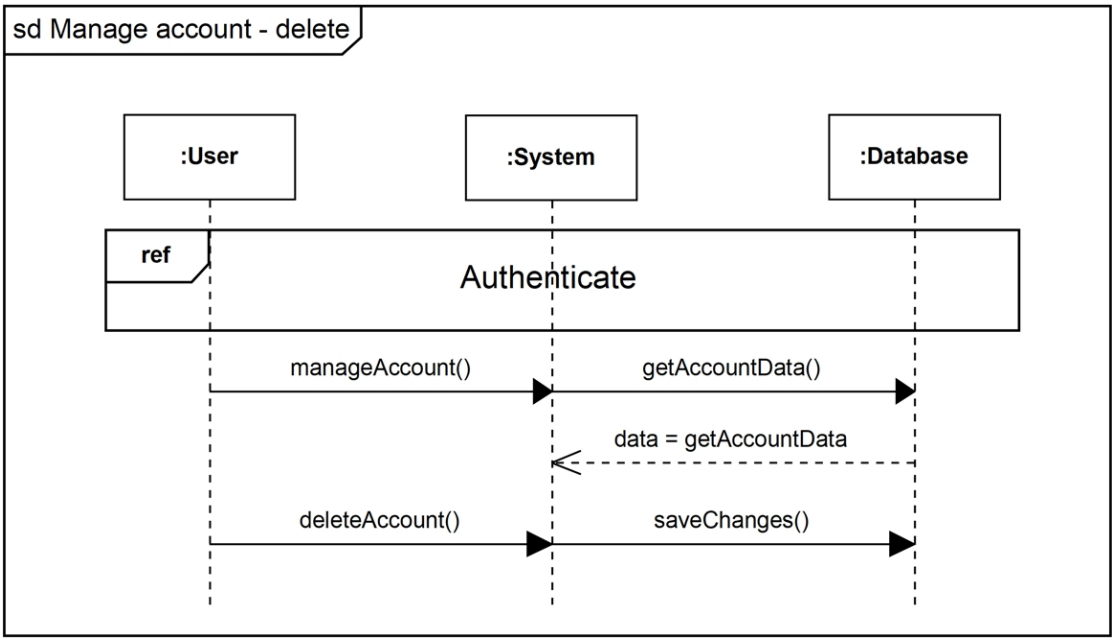


FIGURE 4.7: Sequence diagram for ”Manage account - delete”.

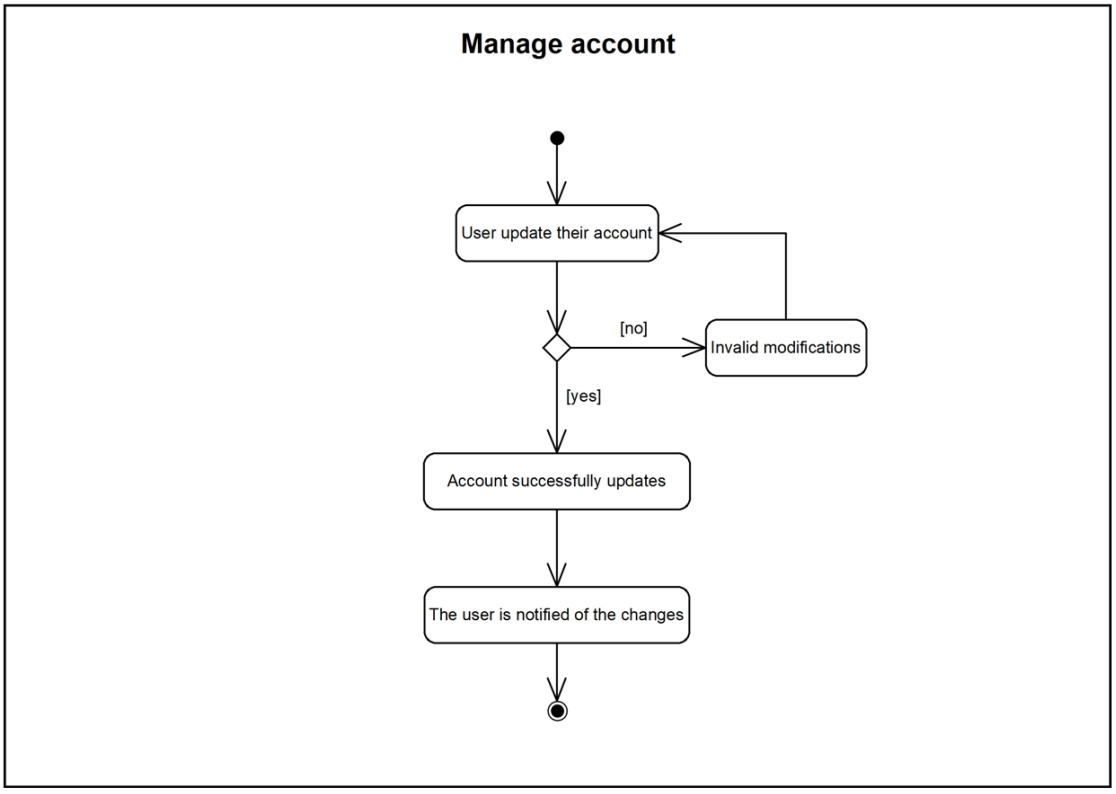


FIGURE 4.8: Activity diagram ”Manage account”.

Name	**Rate**
Brief Description	The ”Rate” use case allows the drivers and passengers to rate each other.
Actors	Passenger, Driver
Preconditions	The user has to be authenticated. The passenger/driver at least had a ride with the other driver/passenger.
Basic Flow	1. The user select another user (driver or passenger) to rate. 2. The system displays the rate panel. 3. The user choose a number of stars and may add a description. 4. The system saves the changes. The system notifies the user about the updates.
Alternate Flows	None
Postconditions	The passenger/driver is rated.

TABLE 4.7: Use case description for ”Rate”.

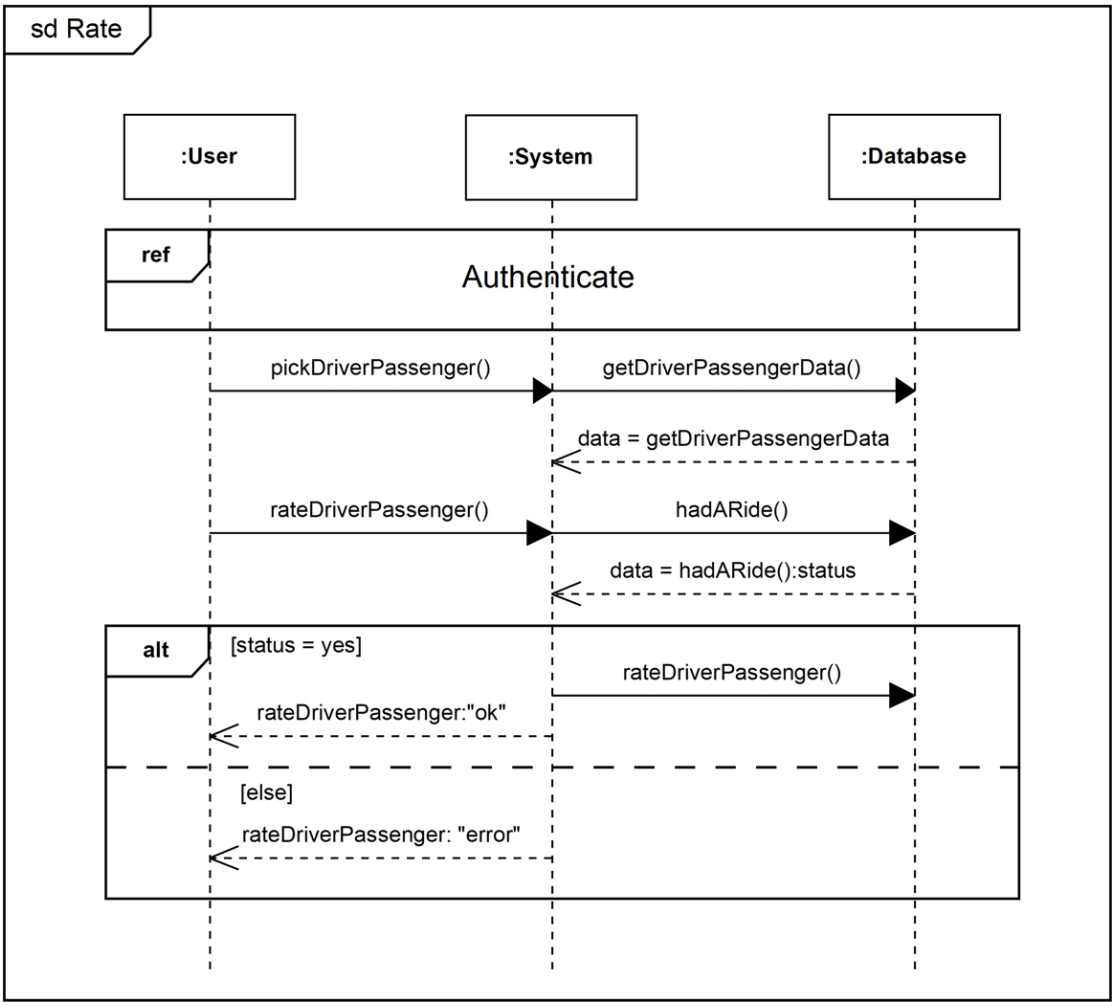


FIGURE 4.9: Sequence diagram for ”Rate”.

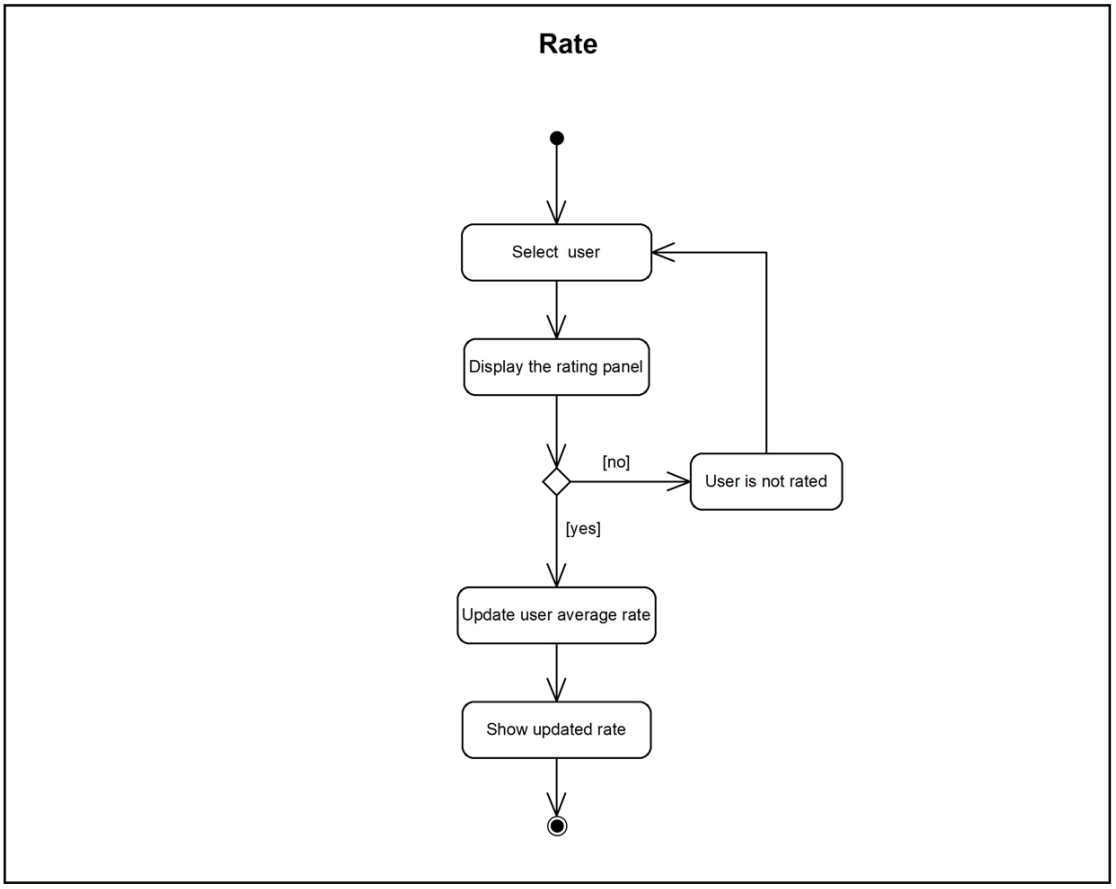


FIGURE 4.10: Activity diagram for ”Rate”.

4.2.2.3 Passenger use cases description

Name	**Find rides**
Brief Description	The ”Find rides” use case allows the user to search for rides based on the chosen origin, destination and date.
Actors	Passenger
Preconditions	The user has to be authenticated.
Basic Flow	1. The user enters the origin, destination and date of the ride. 2. The system shows a list of drivers according to the passenger requirements. 3. The user selects a ride. 4. The system displays information of the ride.
Alternate Flows	None
Postconditions	The user finds the ride they are looking for.

TABLE 4.8: Use case description for ”Find rides”.

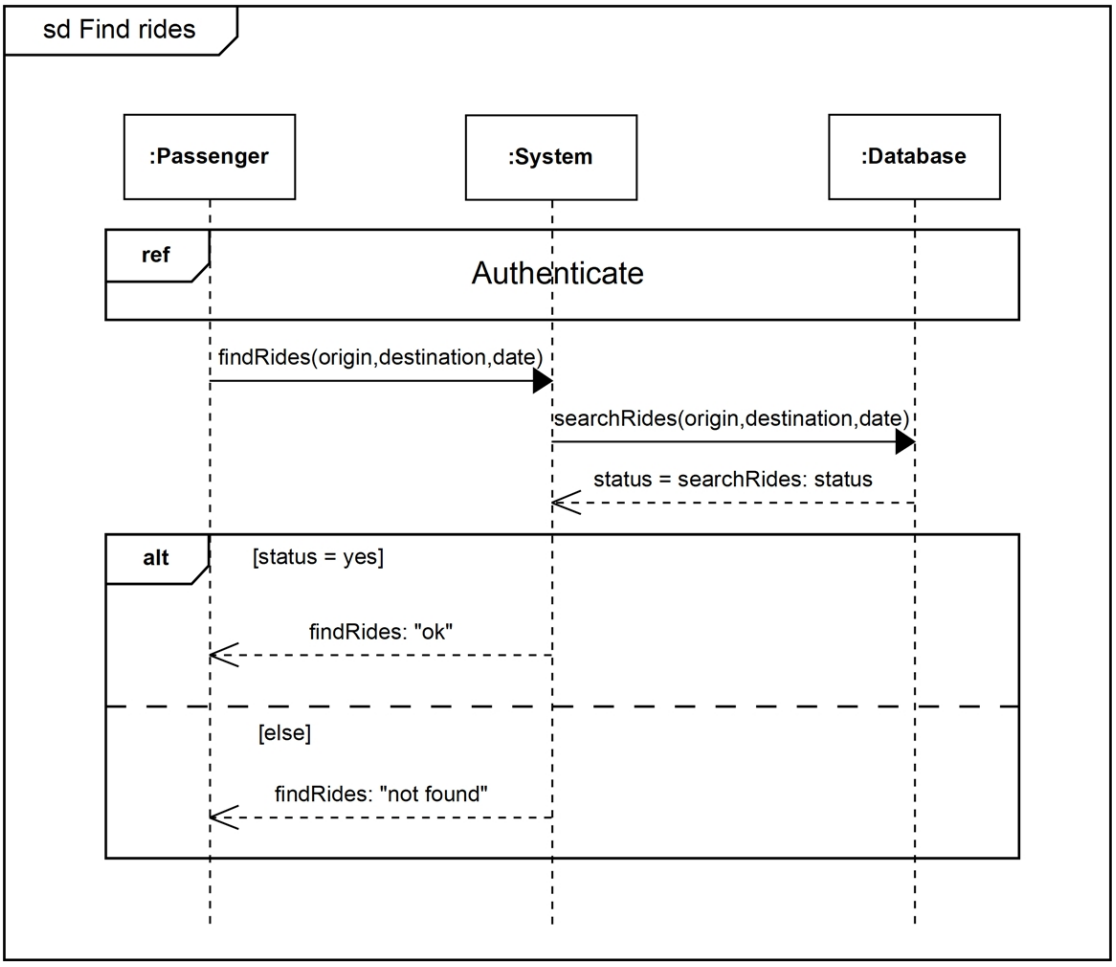


FIGURE 4.11: Sequence diagram for ”Find rides”.

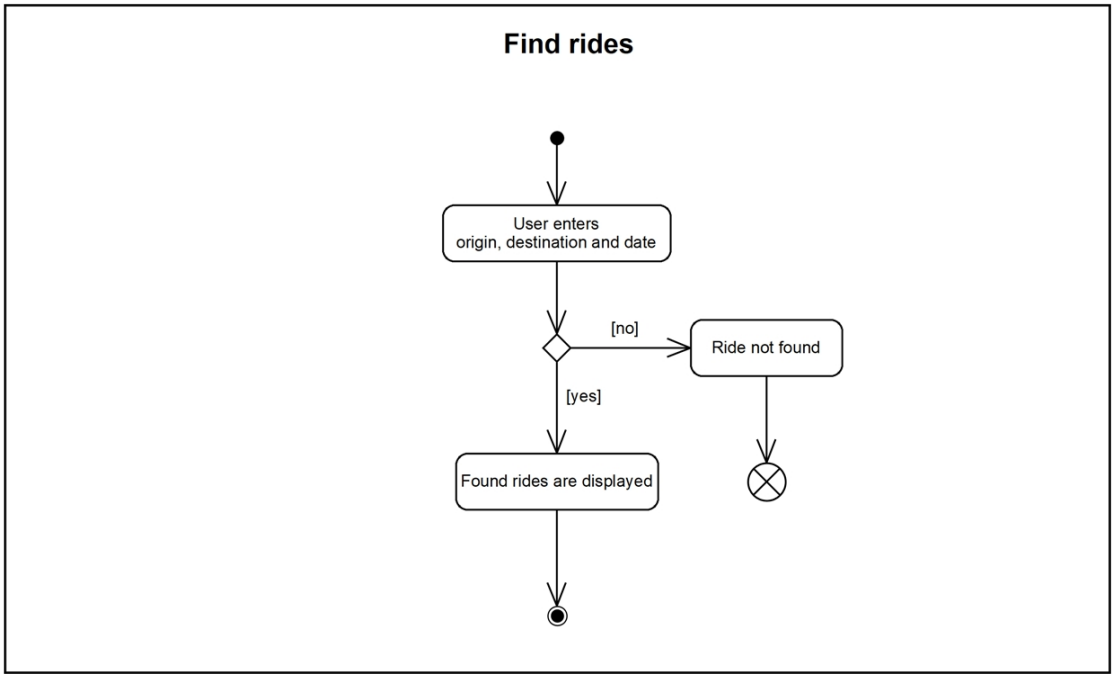


FIGURE 4.12: Activity diagram for ”Find rides”.

Name	**Request a ride**
Brief Description	The ”Request a ride” use case allows the user to request a ride .
Actors	Passenger
Preconditions	The user has to be authenticated.
Basic Flow	1. The passenger selects request a ride from a list of rides displayed by the system. 2. The system asks the passenger to select a number of seats. 3. The passenger selects the number of seats they need. 4. The system sends a request to the corresponding driver of the requested ride.
Alternate Flows	None
Postconditions	A request is sent to the corresponding driver.

TABLE 4.9: Use case description for ”Request a ride”.

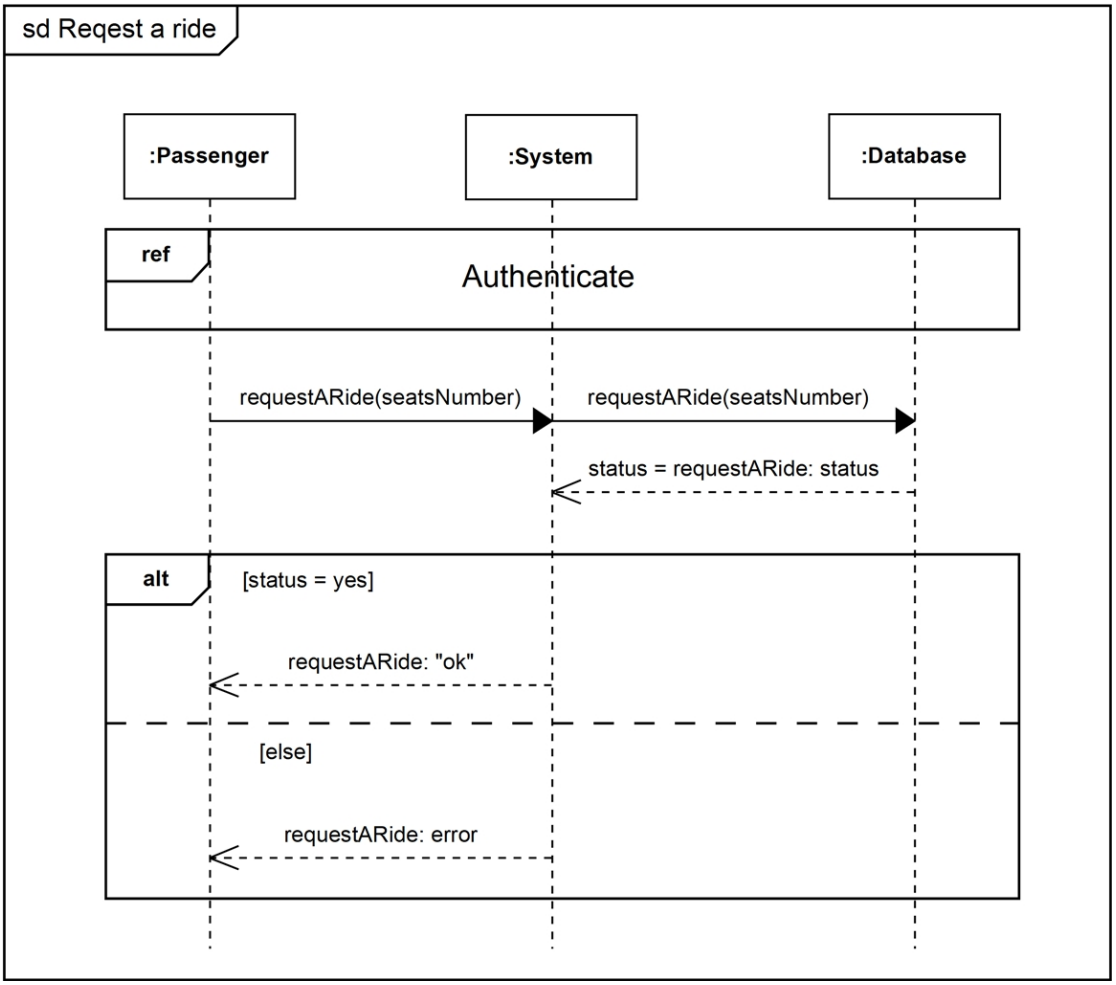


FIGURE 4.13: Sequence diagram for ”Request a ride”.

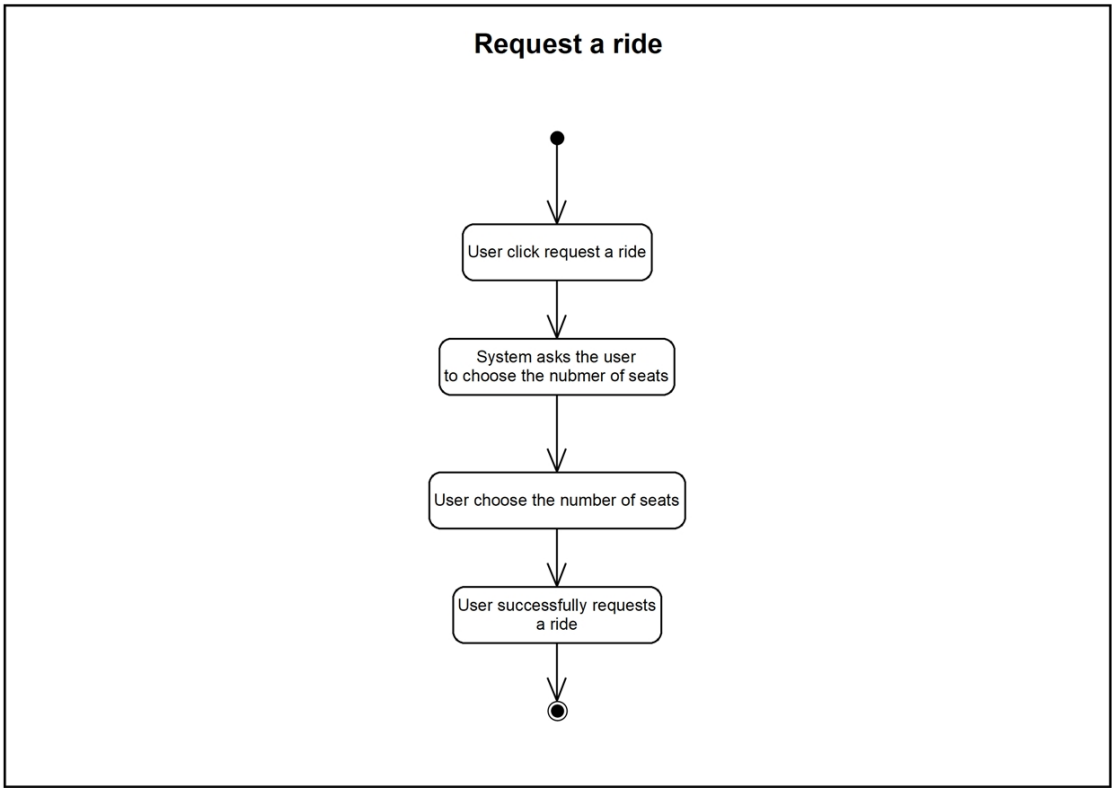


FIGURE 4.14: Activity diagram for ”Request a ride”.

Name	**Manage requests**
Brief Description	The ”Manage requests” use case allows the user to view, edit and cancel requests.
Actors	Passenger
Preconditions	The user has to be authenticated.
Basic Flow	1. The system displays the following options: • View requests. • Edit requests. • Cancel requests.
Alternate Flows	None
Postconditions	The changes made to requests are saved to the user’s account.

TABLE 4.10: Use case description for ”Manage requests”.

Name	**Manage requests - view**
Brief Description	The ”Manage requests - view” use case allows the user to view requests.
Actors	Passenger
Preconditions	The user has to be authenticated.
Basic Flow	1. The system displays all requests and their details. 2. The passenger selects the request to be viewed. 3. The system displays the request.
Alternate Flows	None
Postconditions	The passenger consulted their requests.

TABLE 4.11: Use case description for ”Manage requests - view”.

Name	**Manage requests - edit**
Brief Description	The ”Manage requests - edit” use case allows the user to edit their requests.
Actors	Passenger
Preconditions	The user has to be authenticated.
Basic Flow	1. The system displays all requests and their details. 2. The passenger selects the request to be edited. 3. The passenger makes the needed changes and asks the system to save them. 4. The system saves the changes and notifies the passenger and the driver about the updates.
Alternate Flows	None
Postconditions	The changes made to requests are saved to the user’s account.

TABLE 4.12: Use case description for ”Manage requests - edit”.

Name	**Manage requests - cancel**
Brief Description	The ”Manage requests - cancel” use case allows the user to cancel requests.
Actors	Passenger
Preconditions	The user has to be authenticated.
Basic Flow	1. The system displays all requests and their details. 2. The passenger selects the request to be canceled. 3. The passenger cancel a request and asks the system to save it. 4. The system saves the changes and notifies the passenger and the driver about the updates.
Alternate Flows	None
Postconditions	The changes made to requests are saved to the user’s account.

TABLE 4.13: Use case description for ”Manage requests - cancel”.

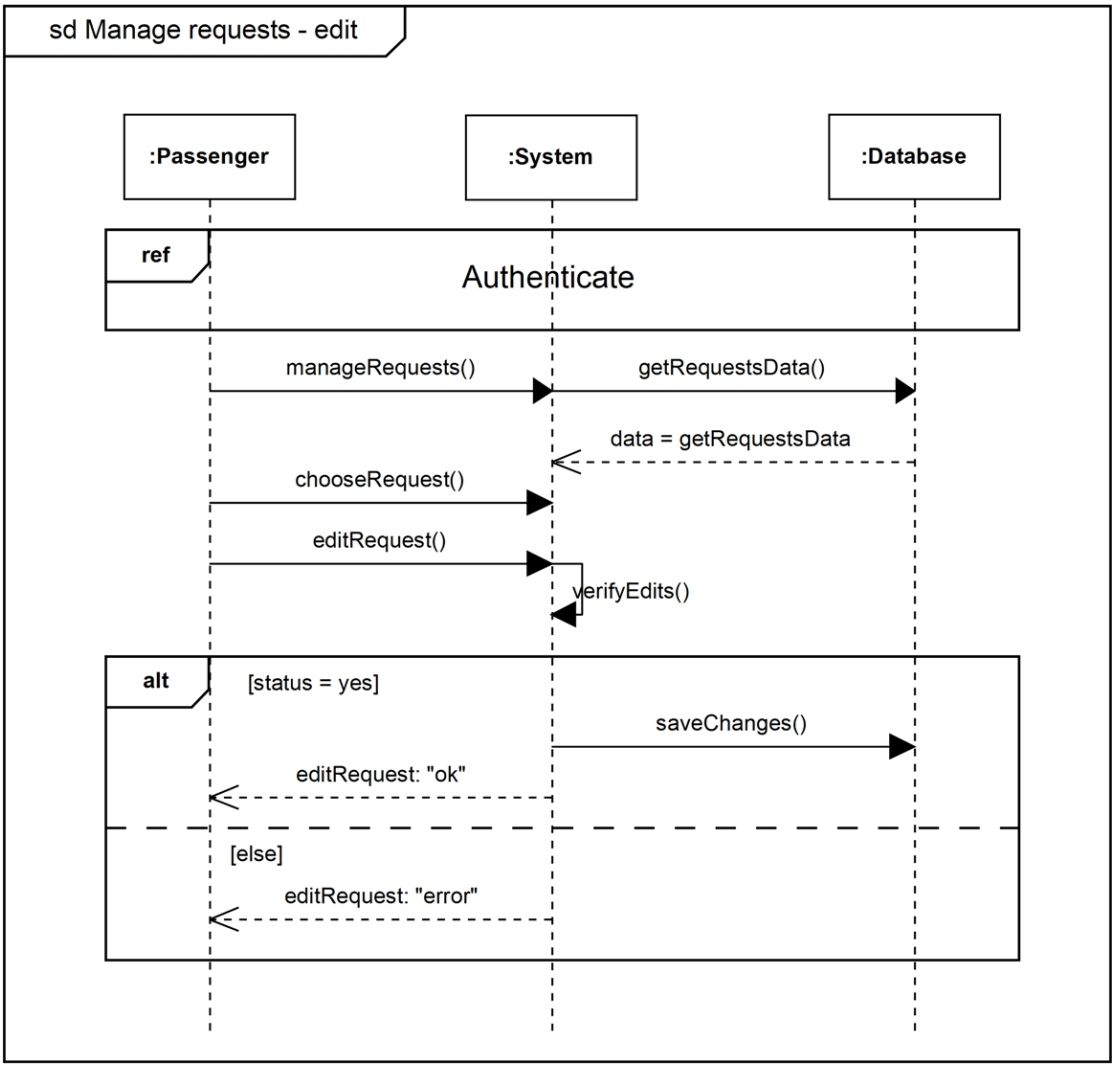


FIGURE 4.15: Sequence diagram for ”Manage requests - edit”.

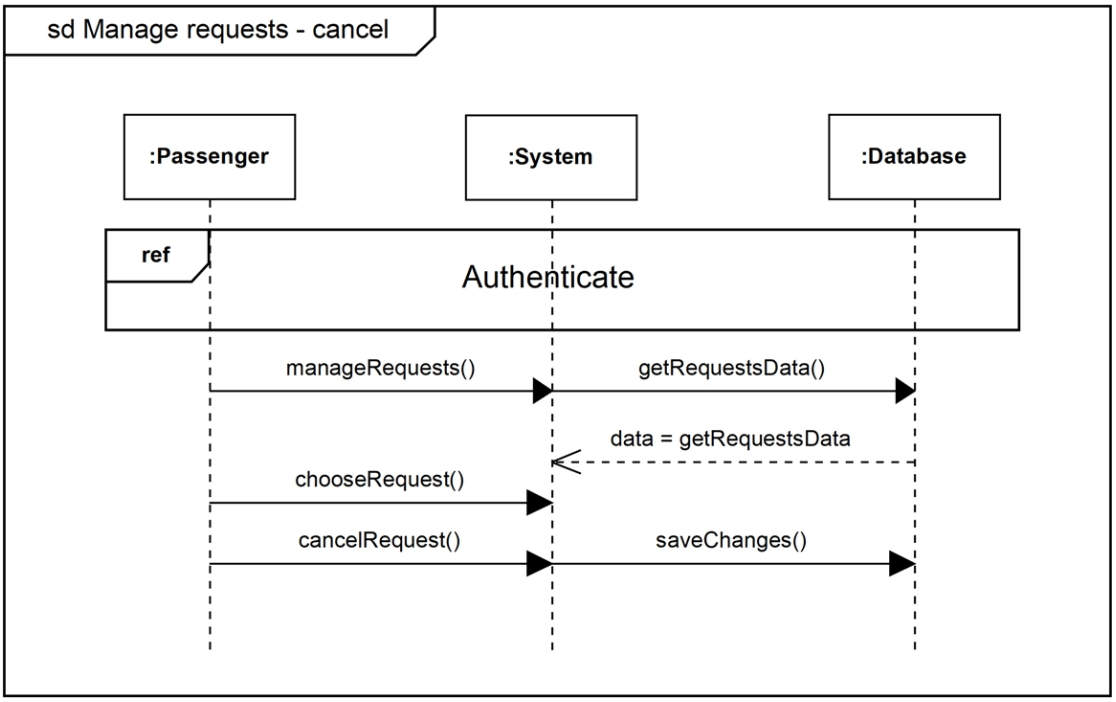


FIGURE 4.16: Sequence diagram for ”Manage requests - cancel”.

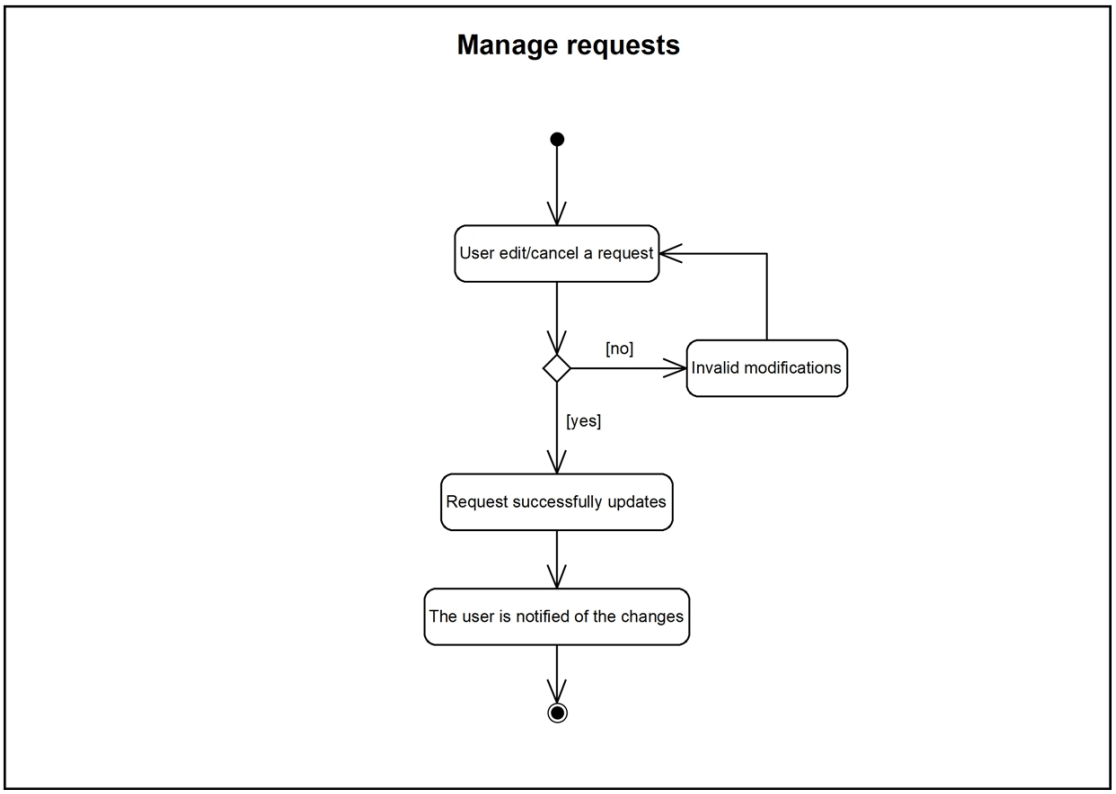


FIGURE 4.17: Activity diagram for ”Manage requests”.

4.2.2.4 Driver use cases description

Name	**Post a ride**
Brief Description	The ”Post a ride” use case allows the user to offer a ride.
Actors	Driver
Preconditions	The user has to be authenticated.
Basic Flow	1. The driver enters the origin, destination and the date and other information concerning the ride. 2. The system saves the ride and notifies the driver that the ride is posted.
Alternate Flows	None
Postconditions	a ride is posted.

TABLE 4.14: Use case description for ”Post a ride”.

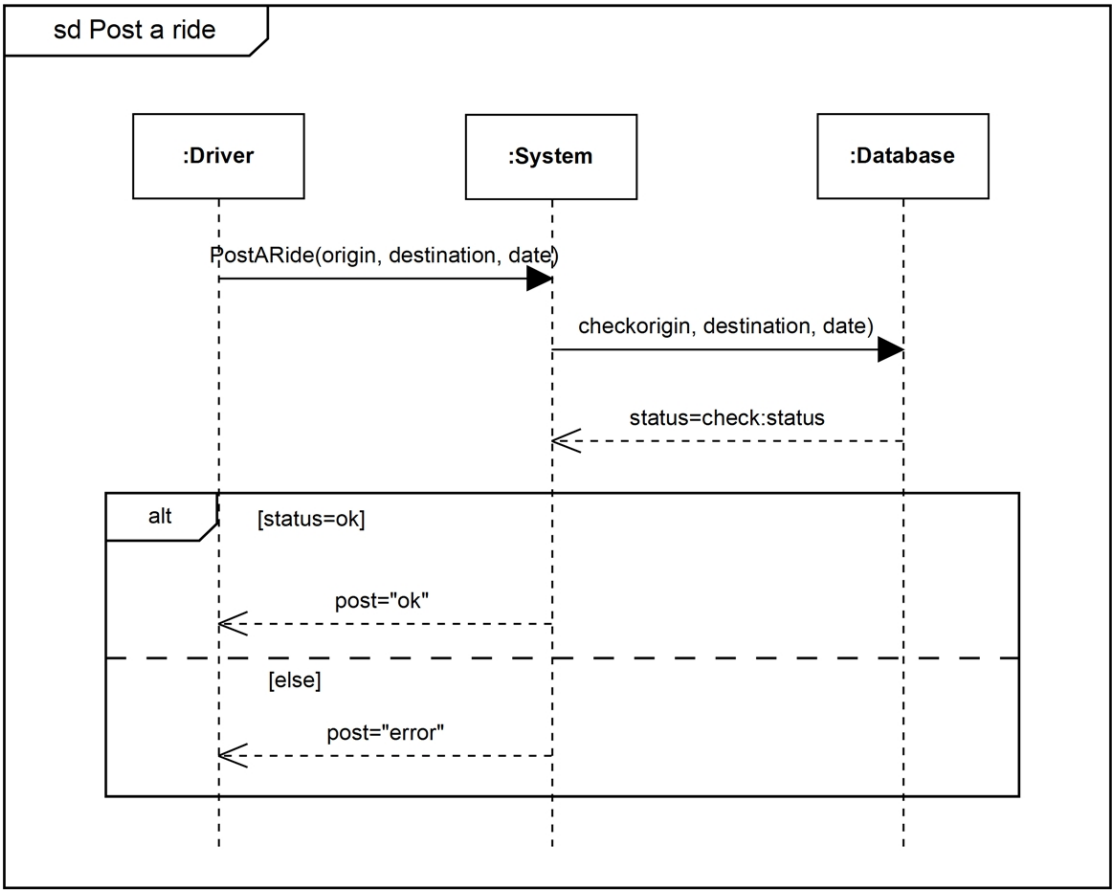


FIGURE 4.18: Sequence diagram for ”Post a ride”.

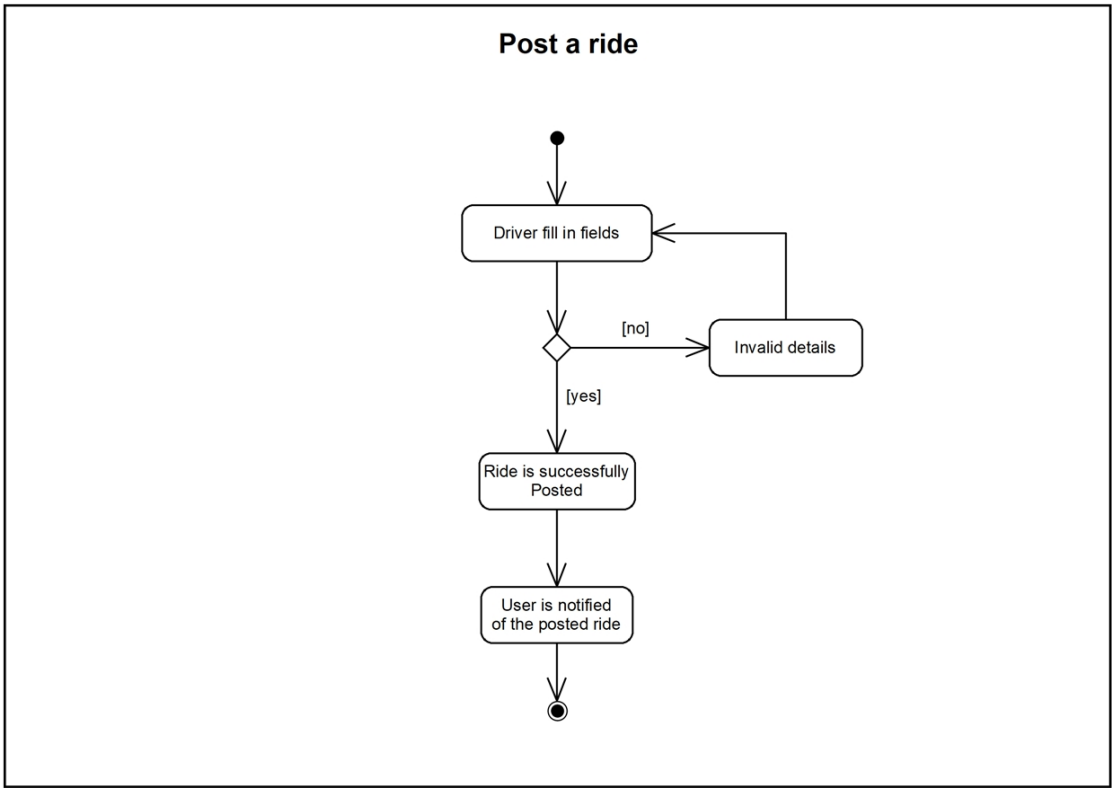


FIGURE 4.19: Activity diagram for ”Post a ride”.

Name	**Confirm request**
Brief Description	The ”Confirm request” use case allows the driver to confirm a request.
Actors	Driver
Preconditions	The user has to be authenticated.
Basic Flow	1. The driver checks the notification window. 2. The system displays all the requested rides. 3. The driver choose to accept or reject requests. 4. The system saves the changes, and notifies the passenger about the driver’s approval.
Alternate Flows	None
Postconditions	The changes made are saved to the users’ accounts.

TABLE 4.15: Use case description for ”Confirm request”.

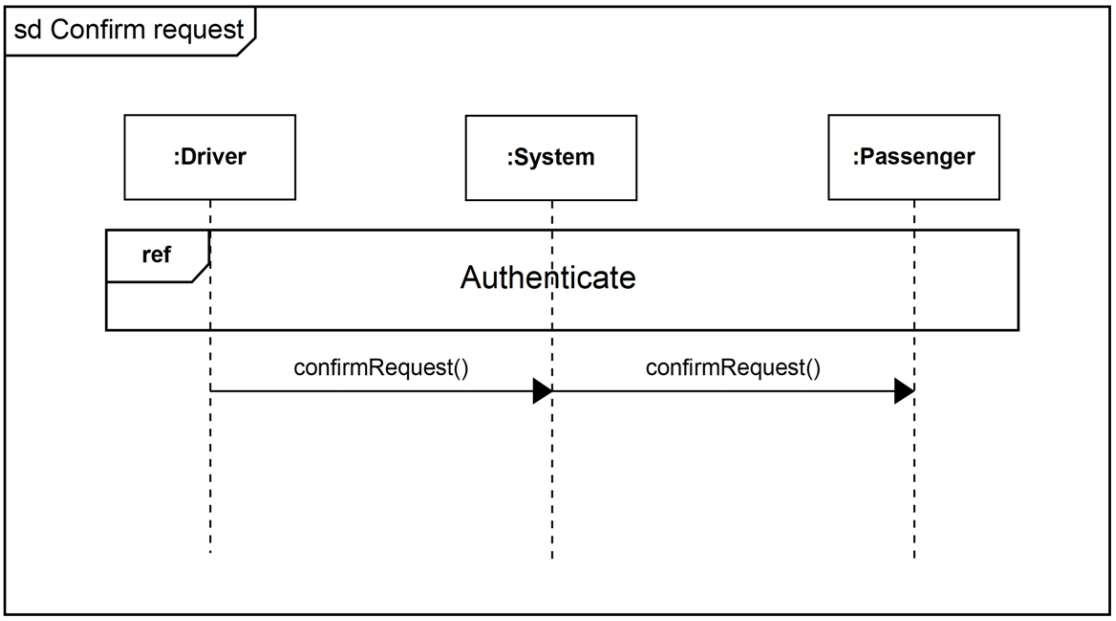


FIGURE 4.20: Sequence diagram for ”Confirm request”.

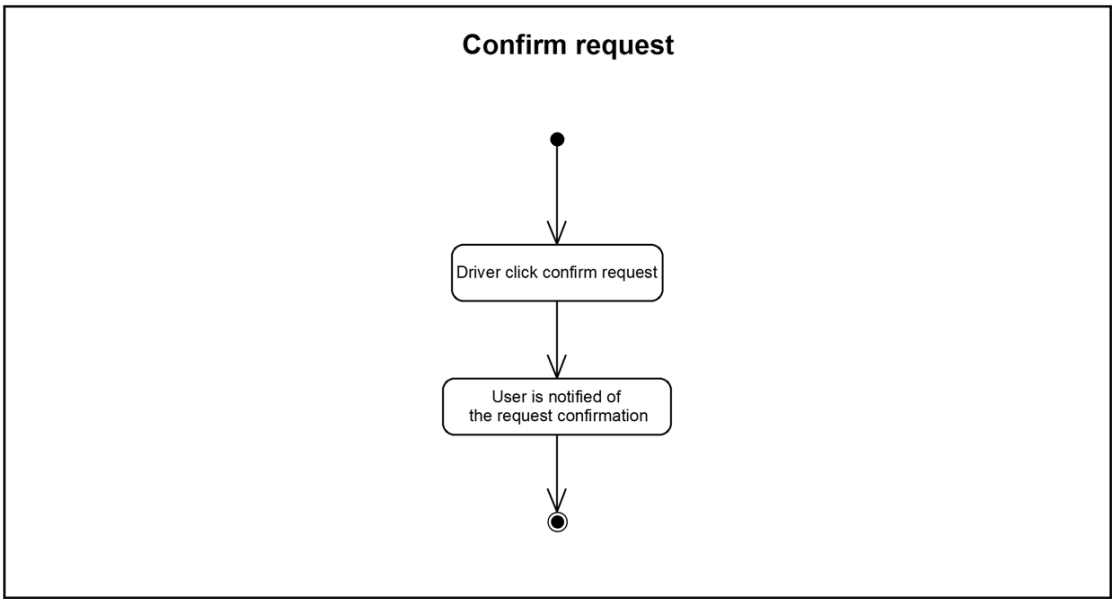


FIGURE 4.21: Activity diagram for ”Confirm request”.

Name	**Manage rides**
Brief Description	The ”Manage rides” use case allows the user to view, edit and cancel posted rides.
Actors	Driver
Preconditions	The user has to be authenticated.
Basic Flow	1. The system displays the following options: • View rides. • Edit rides. • Cancel rides.
Alternate Flows	None
Postconditions	The changes made to rides are saved to the user’s account.

TABLE 4.16: Use case description for ”Manage rides”.

Name	**Manage rides - view**
Brief Description	The ”Manage rides - view” use case allows the user to view the posted rides.
Actors	Driver
Preconditions	The user has to be authenticated.
Basic Flow	1. The system displays all rides and their details. 2. The driver selects the ride to be viewed. 3. The system displays the ride.
Alternate Flows	None
Postconditions	The driver consulted their rides.

TABLE 4.17: Use case description for ”Manage rides - view”.

Name	**Manage rides - edit**
Brief Description	The ”Manage rides - edit” use case allows the user to edit their rides.
Actors	Driver
Preconditions	The user has to be authenticated.
Basic Flow	1. The system displays all rides and their details. 2. The driver selects the ride to be edited. 3. The driver makes the needed changes and asks the system to save them. 4. The system saves the changes and notifies the driver and, the passengers who requested the ride about the updates.
Alternate Flows	None
Postconditions	The changes made to rides are saved to the user’s account.

TABLE 4.18: Use case description for ”Manage rides - edit”.

Name	**Manage rides - cancel**
Brief Description	The ”Manage rides - cancel” use case allows the user to cancel rides.
Actors	Driver
Preconditions	The user has to be authenticated.
Basic Flow	1. The system displays all rides and their details. 2. The driver selects the ride to be canceled. 3. The driver cancel a ride and asks the system to save it. 4. The system saves the changes and notifies the driver, and the passengers who requested the ride about the updates.
Alternate Flows	None
Postconditions	The changes made to rides are saved to the user’s account.

TABLE 4.19: Use case description for ”Manage rides - cancel”.

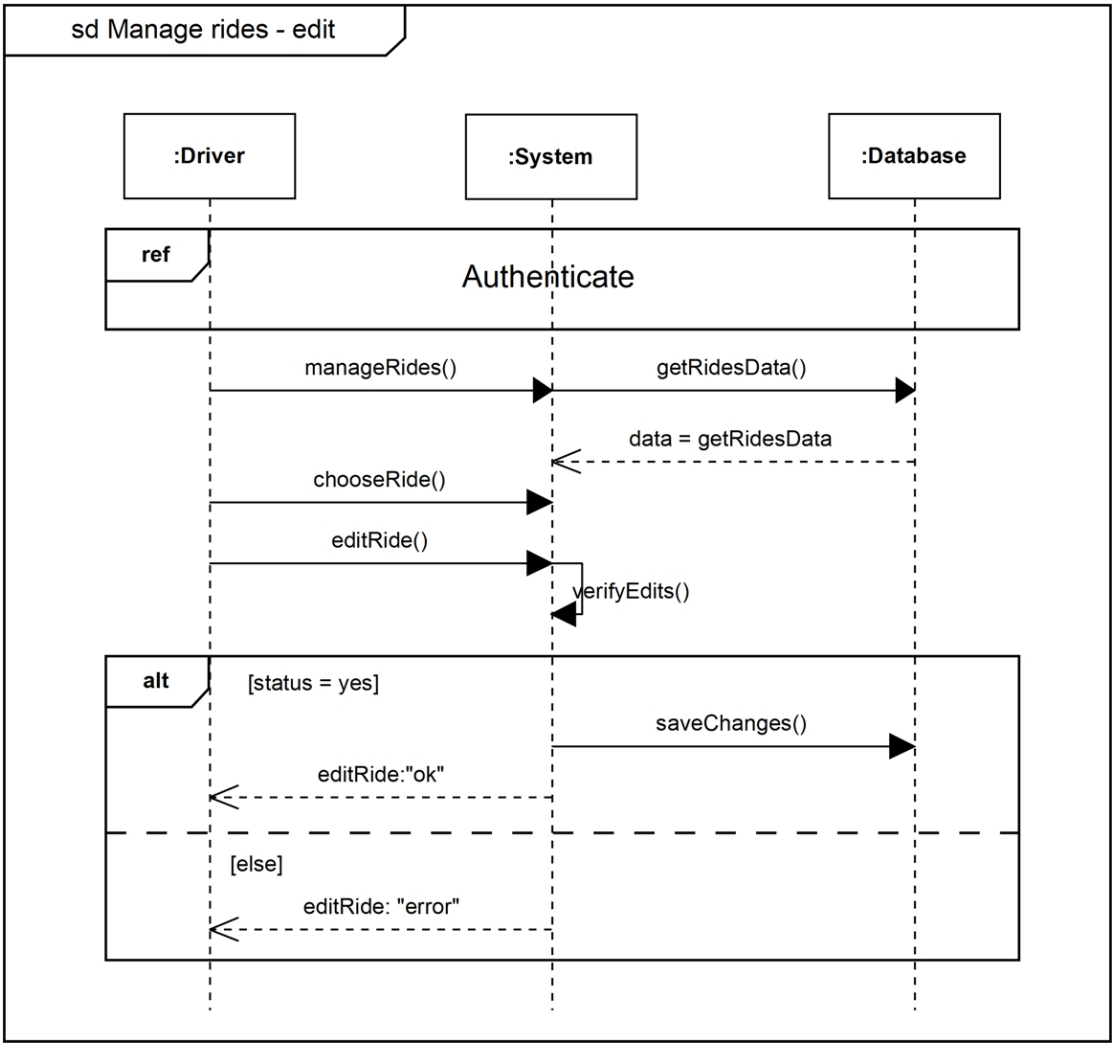


FIGURE 4.22: Sequence diagram for ”Manage rides - edit”.

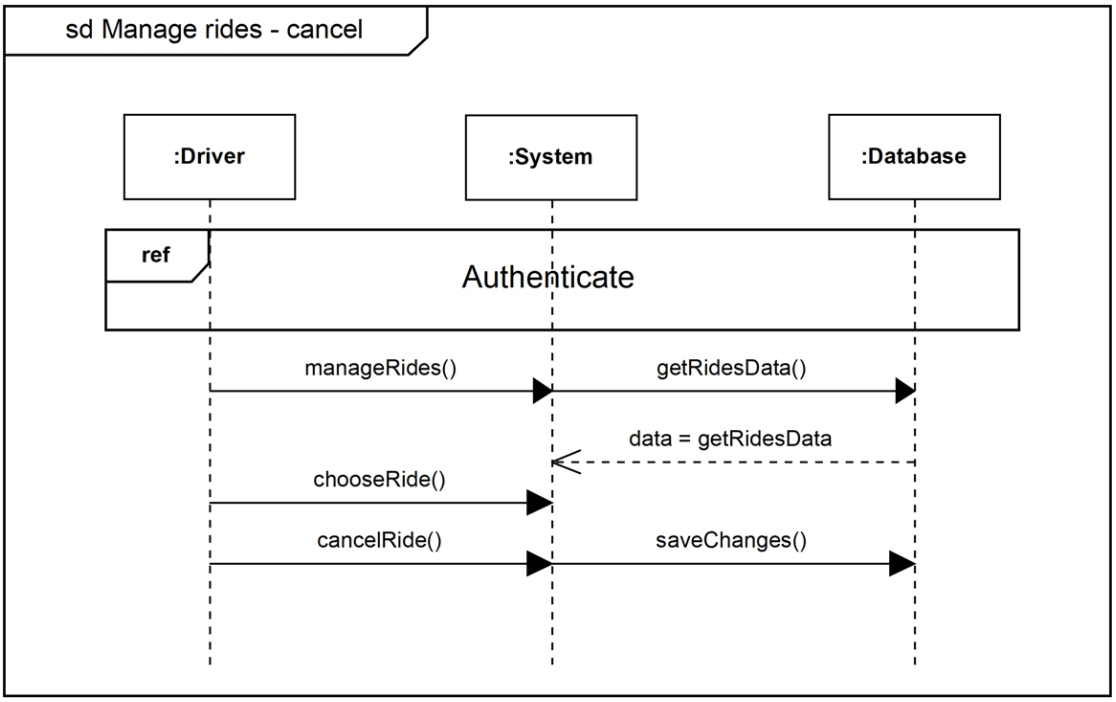


FIGURE 4.23: Sequence diagram for ”Manage rides - cancel”.

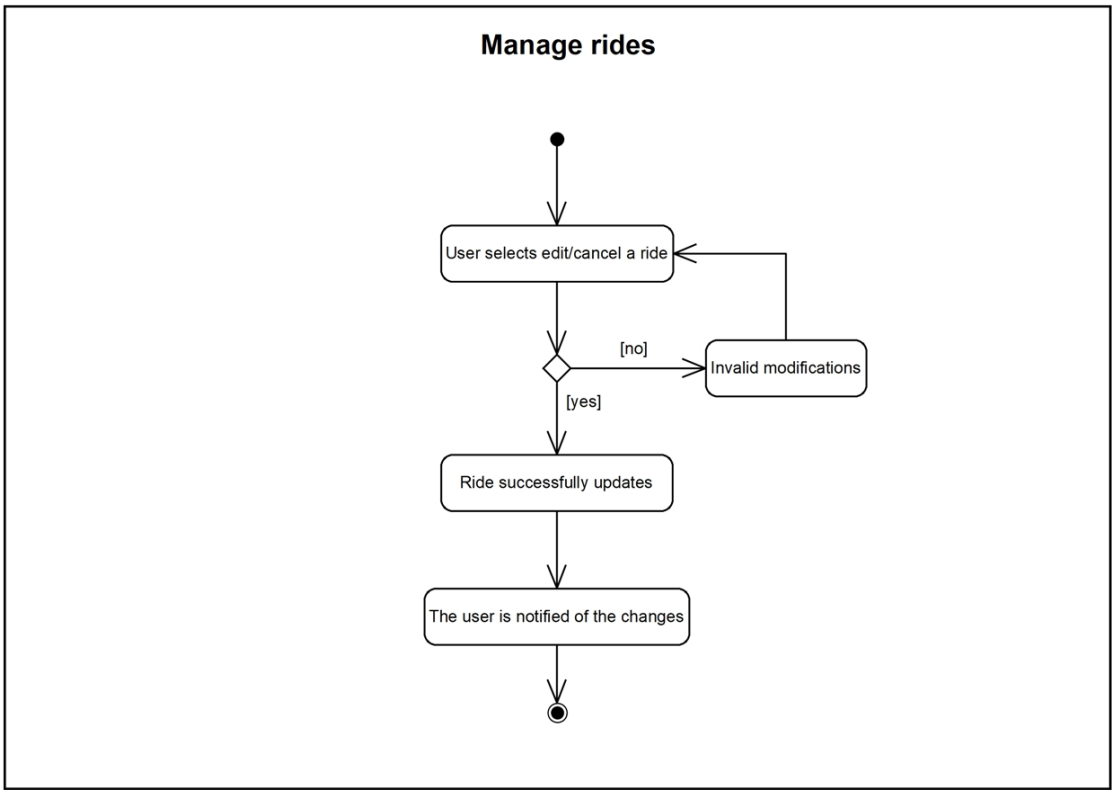


FIGURE 4.24: Activity diagram for ”Manage rides”.

4.2.2.5 Administrator use cases description

Name	**Manage users accounts**
Brief Description	The ”Manage users accounts” use case allows the administrator to view, search, activate, deactivate and delete users accounts.
Actors	Administrator
Preconditions	The user has to be authenticated.
Basic Flow	1. The system displays the following options: • View accounts. • Search for users accounts. • Activate users accounts. • deactivate users accounts. • Delete users accounts.
Alternate Flows	None
Postconditions	The changes made to accounts are saved.

TABLE 4.20: Use case description for ”Manage users accounts”.

Name	**Manage users accounts - view**
Brief Description	The ”Manage users accounts - view” use case allows the administrator to view the users account.
Actors	Administrator
Preconditions	The user has to be authenticated.
Basic Flow	1. The administrator asks to view the users accounts. 2. The system displays the account.
Alternate Flows	None
Postconditions	The administrator viewed users accounts.

TABLE 4.21: Use case description for ”Manage users accounts - view”.

Name	**Manage users accounts - search**
Brief Description	The ”Manage users accounts - search” use case allows the administrator to search for users.
Actors	Administrator
Preconditions	The user has to be authenticated.
Basic Flow	1. The administrator searches for users. 2. The system displays users.
Alternate Flows	None
Postconditions	The administrator consulted users accounts.

TABLE 4.22: Use case description for ”Manage users accounts - search”.

Name	**Manage users accounts - delete**
Brief Description	The ”Manage users accounts - delete” use case allows the administrator to delete users accounts.
Actors	Administrator
Preconditions	The user has to be authenticated.
Basic Flow	1. The administrator asks to delete users accounts. 2. The system deletes the account (accounts) and notifies the administrator about the account removal.
Alternate Flows	None
Postconditions	users accounts are removed.

TABLE 4.23: Use case description for ”Manage users accounts - delete”.

Name	**Manage users accounts - activate**
Brief Description	The ”Manage users accounts - activate” use case allows the administrator to activate users accounts.
Actors	Administrator
Preconditions	The user has to be authenticated.
Basic Flow	1. The administrator asks to activate users accounts. 2. The system activate the account (accounts) and notifies the administrator about the updates.
Alternate Flows	None
Postconditions	users accounts are activated.

TABLE 4.24: Use case description for ”Manage users accounts - activate”.

Name	**Manage users accounts - deactivate**
Brief Description	The ”Manage users accounts - deactivate” use case allows the administrator to deactivate users accounts.
Actors	Administrator
Preconditions	The user has to be authenticated.
Basic Flow	1. The administrator asks to deactivate users accounts. 2. The system deactivate the account (accounts) and notifies the administrator about the updates.
Alternate Flows	None
Postconditions	users accounts are activated.

TABLE 4.25: Use case description for ”Manage users accounts - deactivate”.

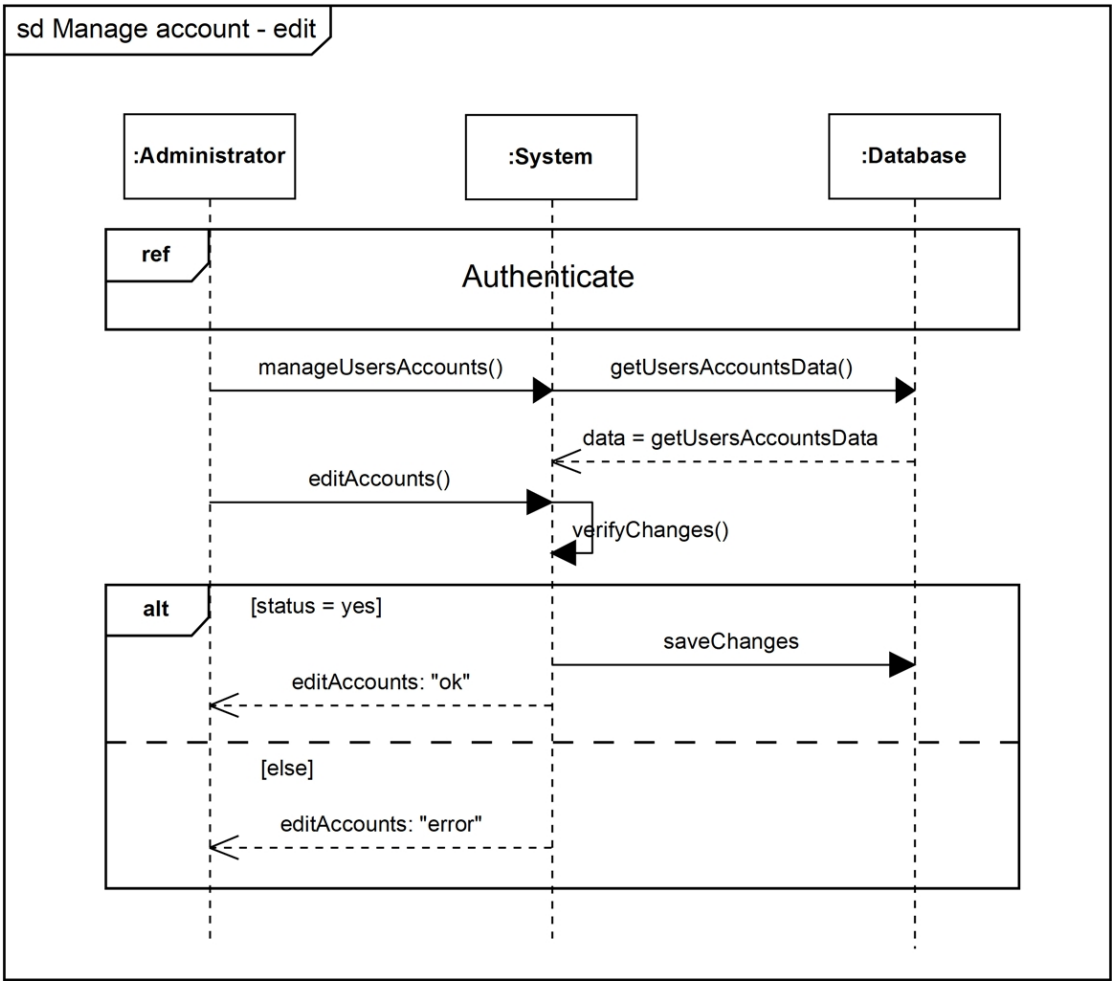


FIGURE 4.25: Sequence diagram for ”Manage users accounts - edit”.

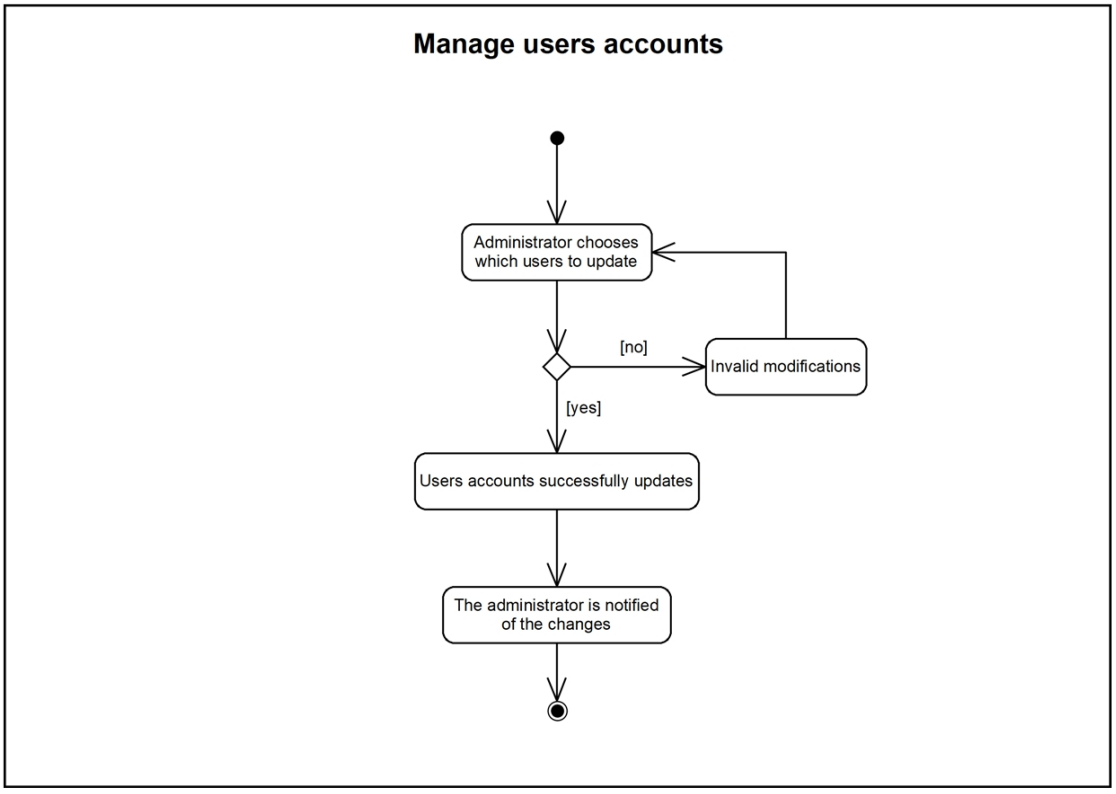


FIGURE 4.26: Activity diagram for ”Manage users accounts”.

Name	**Manage users rides**
Brief Description	The ”Manage users rides” use case allows the administrator to view, edit and cancel posted rides.
Actors	Administrator
Preconditions	The user has to be authenticated.
Basic Flow	1. The system displays the following options: • View rides. • Search for rides. • Delete rides.
Alternate Flows	None
Postconditions	The changes made to rides are saved to the user’s account.

TABLE 4.26: Use case description for ”Manage users rides”.

Name	**Manage users rides - view**
Brief Description	The ”Manage users rides - view” use case allows the user to view rides.
Actors	Administrator
Preconditions	The user has to be authenticated.
Basic Flow	1. The system displays all rides and their details. 2. The administrator selects the ride to be viewed. 3. The system displays the ride.
Alternate Flows	None
Postconditions	The administrator consulted rides.

TABLE 4.27: Use case description for ”Manage users rides - view”.

Name	**Manage users rides - search**
Brief Description	The ”Manage users rides - search” use case allows the administrator to search for rides.
Actors	Administrator
Preconditions	The user has to be authenticated.
Basic Flow	1. The administrator search for rides. 2. The system displays rides.
Alternate Flows	None
Postconditions	The administrator searched for rides.

TABLE 4.28: Use case description for ”Manage users rides - edit”.

Name	**Manage users rides - delete**
Brief Description	The ”Manage users rides - delete” use case allows the user to delete rides.
Actors	Administrator
Preconditions	The user has to be authenticated.
Basic Flow	1. The system displays all rides and their details. 2. The administrator selects the ride to be deleted. 3. The administrator delete a ride (rides) and asks the system to save the changes. 4. The system saves the changes and notifies the administrator about the updates.
Alternate Flows	None
Postconditions	The changes made to rides are saved to the user’s account.

TABLE 4.29: Use case description for ”Manage users rides - delete”.

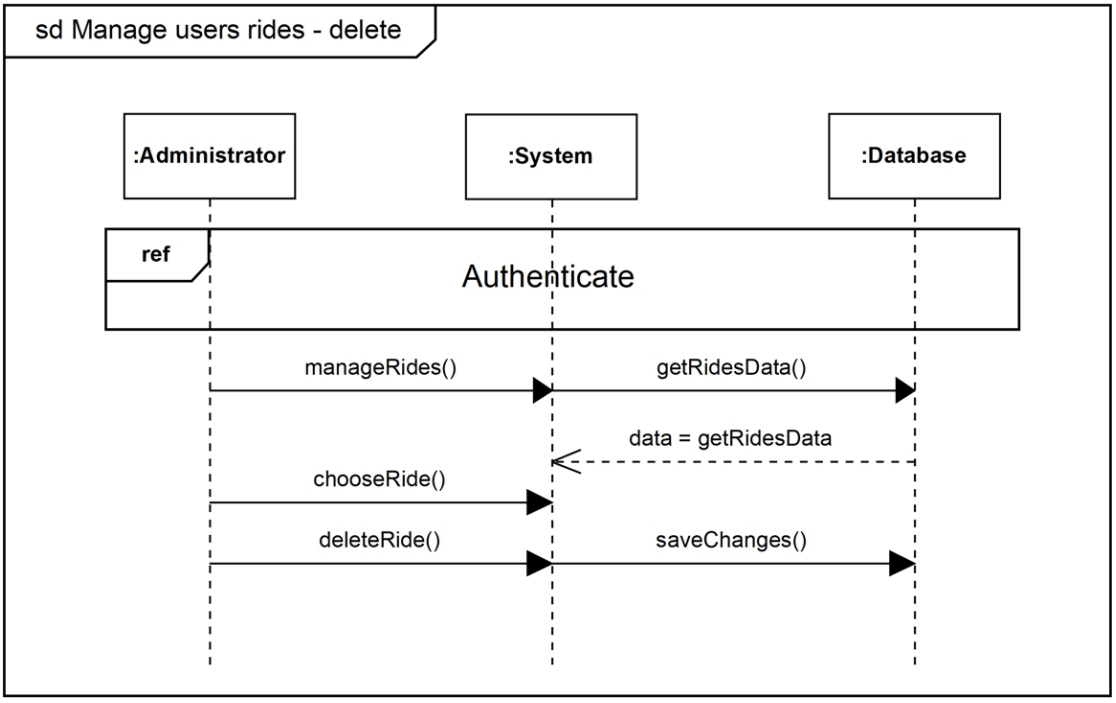


FIGURE 4.27: Sequence diagram for ”Manage users rides - delete”.

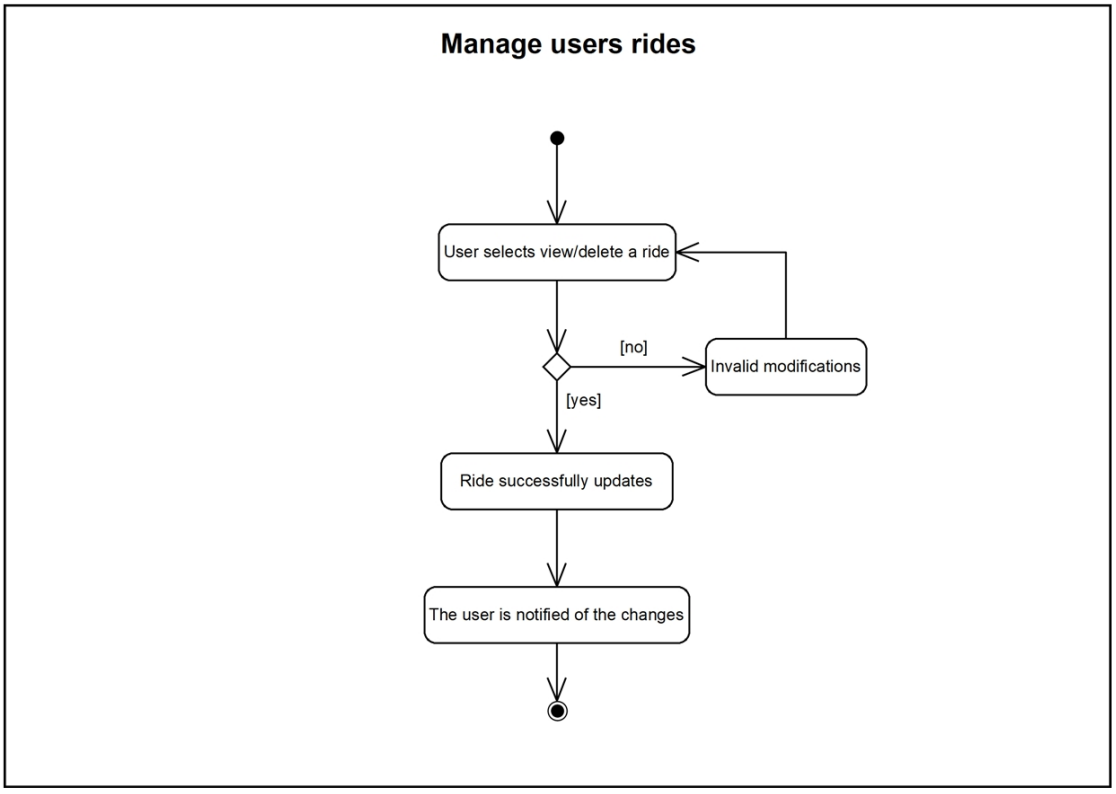


FIGURE 4.28: Activity diagram for ”Manage users rides”.

Name	**Statistics**
Brief Description	The ”Statistics” use case allows the administrator to view different statistics about the system users and the daily rides.
Actors	Administrator
Preconditions	The user has to be authenticated.
Basic Flow	1. The administrator asks the system for statistics. 2. The system displays statistics about the users of the system and the different rides.
Alternate Flows	None
Postconditions	The administrator can see the statistics.

TABLE 4.30: Use case description for ”Statistics”.

4.3 Class conception diagram

In this section, we give the class diagram of our system resulting from a detailed conception of its functioning, and let before give the catalog of different classes and their attributes, while providing a description for each one.

4.3.1 Class catalog

Class	Attributes	Description
Request	id	The request identifier.
	owner	The identifier of the passenger sending the request.
	seatsNumber	The number of seats requested by the passenger.
	confirmed	A boolean to indicate if the request is confirmed by the driver or not.
Notification	id	The notification identifier
	owner	The identifier of the notification owner.
	text	The text of the notification.
	seen	A boolean to indicate if the notification is read or not.
	date	To indicate the date and time the notification is created.
Rate	id	The rate identifier.
	rater	The identifier of the user who rated.
	ratee	The identifier of the user who has been rated.
	rating	The number of stars given by the rater.
	description	The rate description text.
User	id	The user identifier.
	email	The user’s email.
	password	The user’s password.
	firstName	The user’s first name.
	lastName	The user’s last name.
	dob	The user’s date of birth.
	phone	The user’s phone number.
	gender	The gender of the user.
	role	To specify if the system user is a driver, passenger or both.

	home	The home address.
	specialAddresses	An array of strings for other addresses of the user.
	cars	An array of cars' identifiers.
	rides	An array of rides' identifiers.
	requests	An array of both requests' identifiers and the requested rides identifiers.
	notifications	An array of notifications' identifiers.
	averageRating	The calculated average rating of the user.
	avatar	A string contains the link to the user's avatar.
	smoker	A boolean to indicate of the user is a smokes or not.
	animals	A boolean to indicate of the user carries animals with him.
	foodDrinks	A boolean to indicate of the user carries foods or drinks with him.
	bigBags	A boolean to indicate of the user carries big bags with him.
	talkative	A boolean to indicate of the user like to talk a lot.
Car	id	The car identifier.
	model	The car's model.
	brand	The car's brand.
	manufacturingYear	The year in which the car is manufactured.
	seats	The number of seats of the car.
	imageUrls	An array of strings for the car images.
Address	id	The address identifier.
	wilaya	The address's wilaya.
	street	The address's street.
	city	The address's city.
	lat	The address's latitude.
	lon	The address's longitude.
Ride	id	The ride identifier.
	owner	The owner of the ride.
	origin	The origin address of the ride.
	destination	The destination address of the ride.
	departureTime	The departure time of the ride.

	seatsNumber	The chosen seats number for the ride.
	flexibilityInMinutes	The additional time, which the driver can stay before he/she starts the trip.
	price	The price of a single seat.
	car	The chosen car for the ride.
	requests	An array for the requests sent to that ride.

TABLE 4.31: Classes and attributes description

4.3.2 Class diagram

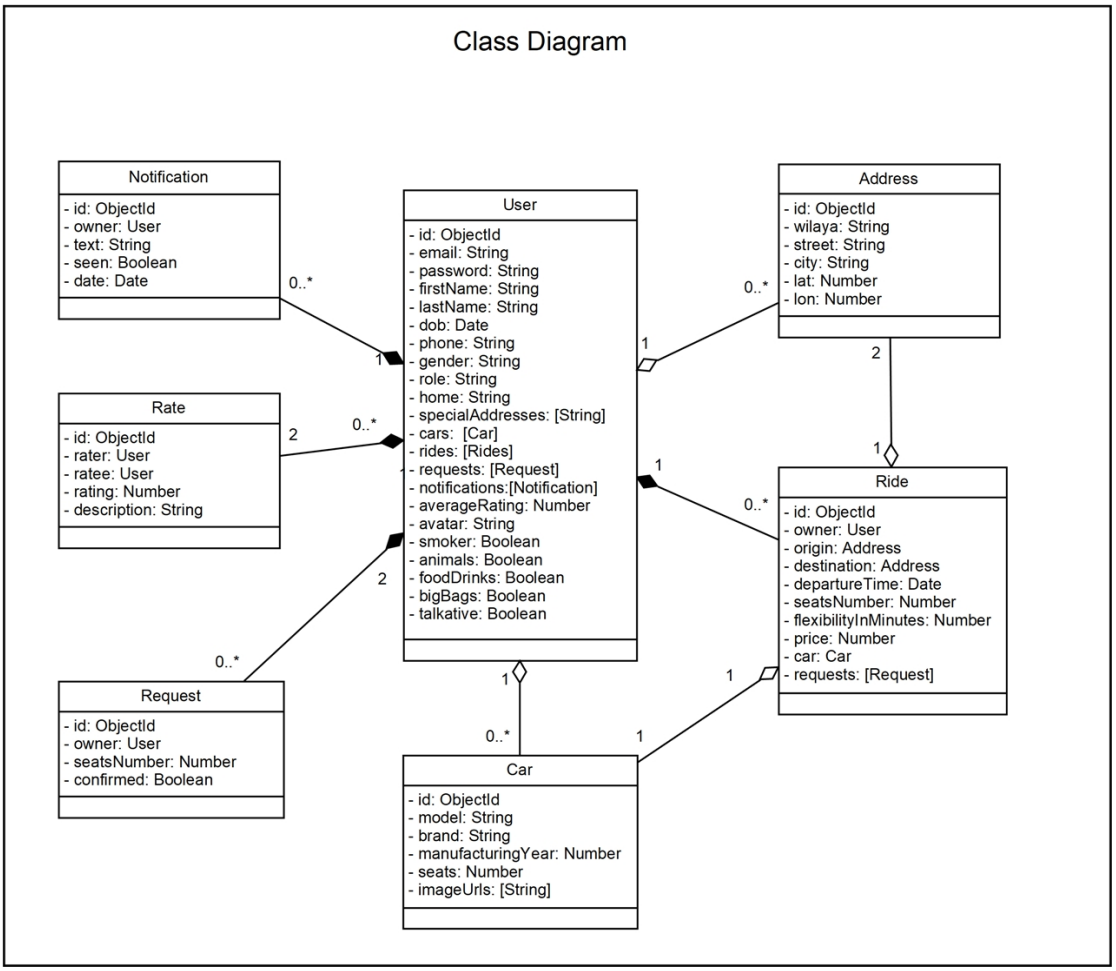


FIGURE 4.29: Class diagram.

4.4 Technical needs

In this section, we present the logical architecture we adopted for our system, and through which we describe the overall functioning of the system. Also, we present the database model used for developing the application.

4.4.1 System architecture

As we can understand from the detailed design presented in the previous sections, we need a two levels architecture, namely a Client/Server architecture, for our system, presented in Figure 5.1. This conclusion is based on the following findings:

- With the growing of the application use over the time, we will get huge data to be managed by the system (users accounts, rides, etc.), and hence a server to store and manage (deal with user requests and provide needed information) is needed.
- Interactions with data are from distributed entities represented by different users of the system. Also, sources of these data are users themselves. These users represent the clients of the system.

We have by definition a Client/Server architecture is an network system that divides functions into client subsystem, which usually requests services, and server subsystem, which provides the requested services, with communication procedures to sharing data between them. This kind of architecture is known to be efficient for sharing database information in a multiuser environment [49]. Therefore, the convenient architecture for our system is the Client/server architecture. Note that there are several kind of server, here our server is a databse server.

In our system, both the visitor and the driver operate on the client side according to the adopted architecture.

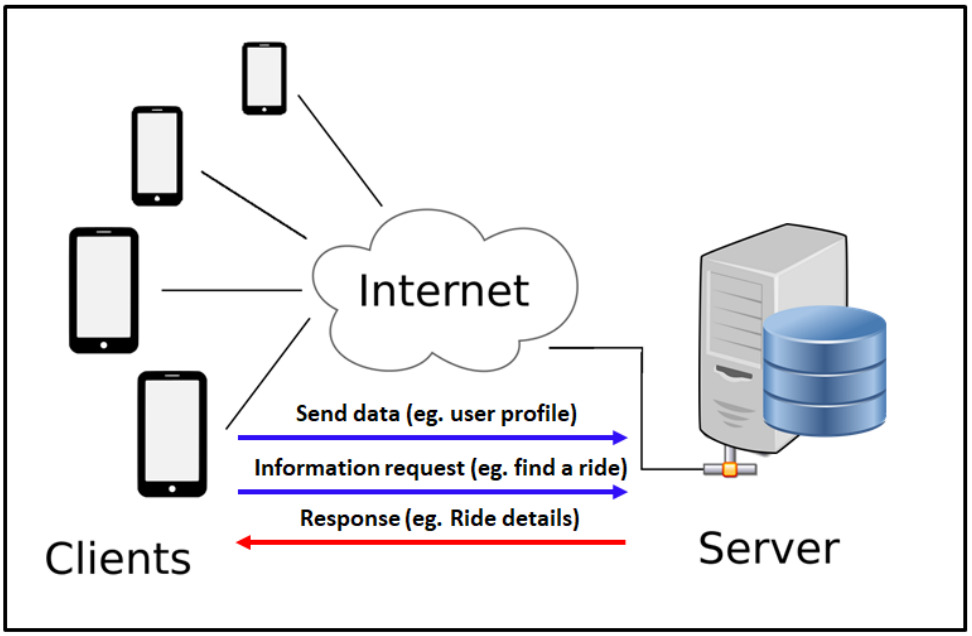


FIGURE 4.30: The system architecture.

4.4.2 Database model

We decided to choose a NoSQL database for developing this application, which is going to be MongoDB. This decision was based on many factors. Indeed, in SQL databases a predefined schema is used to determine the structure of data, and the to define and manipulate data the structured query language is used. However these databases are not effective in dealing with large amounts of data. On the other hand, MongoDB is a document database which is fast in performing simple queries. Also, it allows dynamic schema, making it more flexible, and the most important caracteristics that made us decide to use it is that it scales very well and can hold large data.

4.4.2.1 The NoSQL database system

There are four types of NoSQL database management systems [50]:

- **Key-value stores.** are the simplest NoSQL databases. Every single item in the database is stored as an attribute name (or ”key”) together with its value. Examples include Riak, Voldemort, and Redis.
- **Wide-column stores.** store columns of data together instead of rows and are optimized for queries over large datasets. Cassandra and HBase are wide-column databases.
- **Document databases.** pair each key with a complex data structure known as a document. Documents can contain many different key-value pairs, or key-array pairs, or even nested documents. MongoDB is a document database.

- **Graph databases** are used to store information about networks, such as social connections. Examples include Neo4J and HyperGraphDB.

The MongoDB database stores data in a binary representation called BSON (Binary JSON). The MongoDB documents tend to have all data for a given record in a single document, whereas in a relational database information for a given record is usually spread across many tables. For example, considering the data model for a blogging application, in a relational database, the data model would comprise multiple tables such as Categories, Tags, Users, Comments and Articles, while in MongoDB the data could be modeled as two collections, one for users, and the other for articles as shown in Figure 4.31. As a result of the document model, data in MongoDB is more localized, which dramatically reduces the need to JOIN separate tables [51].

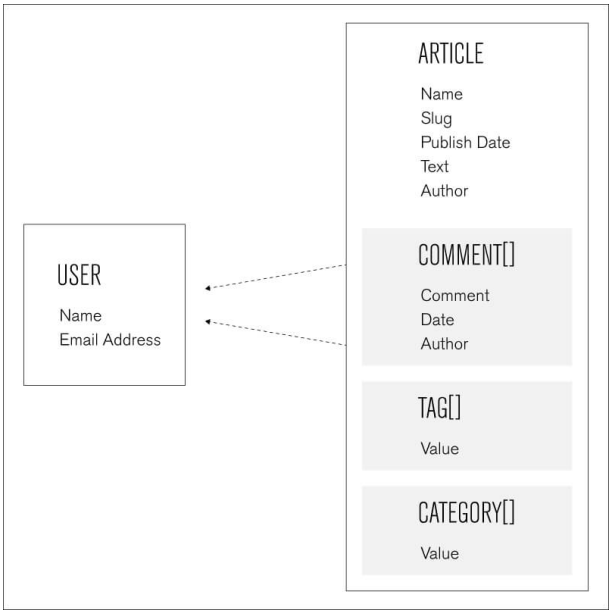


FIGURE 4.31: Example of multiple tables modeled in a single MongoDB collection.

4.5 Conclusion

In this chapter, we presented a detailed design of our system through a set of essential UML diagrams, including the use case diagram, activity diagrams, sequence diagrams and the class diagram. Also, we described the two main technical needs for developing our system, namely the choice of the system architecture as well as the database model. We showed that our system needs a Client/Server architecture, while for the database, we adopted the NoSQL model for different convenient arguments. All that will allow us to embark on the next step, which is implementing our carpool application named ClickRide.

Chapter 5

Implementation

5.1 Introduction

This chapter deals with the whole implementation of the application basing on the analysis and conceptual design of the previous chapter. We will present the system architecture. Then, we will give an overview of the different tools and programming languages used to build both of the mobile application and the web application. Lastly, we will present the system main windows.

5.2 System software architecture

Our system consists of two main parts:

The front-end part, which by itself consists of two different front-end applications: one for general application users, which is a mobile app developed in Java using Android Studio Integrated Development Environment (IDE), and another for the administrator, which is a web based application developed in HTML and CSS with Bootstrap library and Vue.js framework using Visual Studio Code editor.

The back-end part, which consists of a NoSQL database (MongoDB) and an API in Node.js (a JavaScript run-time environment) using express framework and mongoose Object Document Mapper (ODM). The back-end part was developed in Visual Studio Code, and Postman (an API development Environment) for testing our API. The database is hosted in mlab.com and the API in www.heroku.com.

Figure 5.1 shows the basic architecture of our app:

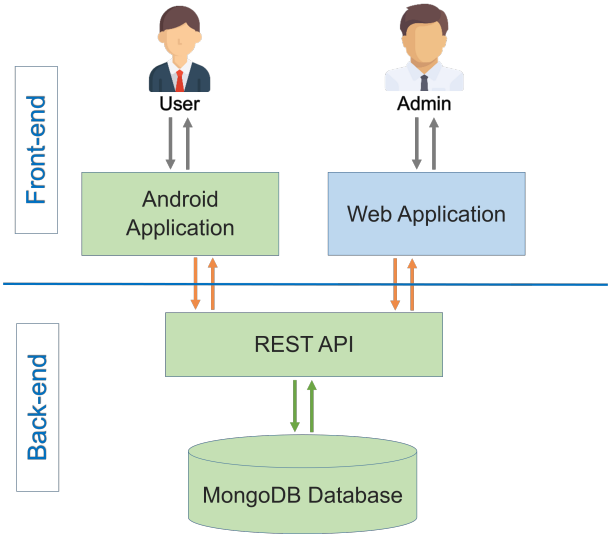


FIGURE 5.1: The system architecture.

5.3 Tools

In this section, we present different tools we used for implementing the application.

5.3.1 Android Studio

Android Studio is the official IDE for Google's Android operating system, based on IntelliJ IDEA and designed specifically for android development. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance the productivity when building Android apps, such as [52]:

- A flexible Gradle-based build system.
- A fast and feature-rich emulator.
- A unified environment where you can develop for all Android devices.
- Instant Run to push changes to your running app without building a new APK.
- Code templates and GitHub integration to help you build common app features and import sample code.
- Extensive testing tools and frameworks.
- Lint tools to catch performance, usability, version compatibility, and other problems.
- C++ and NDK support.
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine.

FIGURE 5.2: Android Studio logo.

5.3.2 Visual Studio Code

Visual Studio Code is a lightweight but powerful source code editor built by Microsoft, which runs on desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages such as: C++, C#, Java, Python, PHP, Go, and runtimes such as: .NET and Unity [53]. We used Visual Studio Code for writing our API.

FIGURE 5.3: Visual Studio Code logo.

5.3.3 Postman

Postman is a complete API development environment: the only end-to-end solution with collaboration, documentation, mocks, and monitoring. It is the essential toolchain for API developers to share, test, document and monitor APIs [54]. We used Postman for testing the API's different routes.

FIGURE 5.4: Postman logo.

5.3.4 www.heroku.com

Heroku is a cloud platform that lets companies build, deliver, monitor and scale apps. It offers a free plan with 512 MB of RAM, which is ideal for experimenting with cloud applications in a limited sandbox [55]. We used Heroku for hosting our API.

FIGURE 5.5: Heroku logo.

5.3.5 mlab.com

mLab is the largest cloud MongoDB service in the world, hosting over a half million deployments on AWS, Azure, and Google. It offers a free Sandbox plan, which provides a single database with 0.5 GB of storage on a shared database server process running on a shared virtual machine (VM). This plan is best for development and prototyping [56]. We used mLab for hosting our database.

FIGURE 5.6: mLab logo.

5.4 Programming languages and technologies

In this section, we present different technologies as well as programming languages used for implementing our application.

5.4.1 Java

Java is a statically-typed object-oriented programming language that produces software for multiple platforms. The compiled code also know as bytecode runs on most operating systems (OS), including Windows, Linux and MacOS. Java is derived from C++. It was originally developed by James Gosling at Sun Microsystems, and initially released in the mid-1990s. We used Java for developing the main part of our system, which is the mobile application.

5.4.2 JavaScript

JavaScript is a scripting language that typically runs in the browser It is primarly used to makes web pages dynamic and interactive by displaying timely content updates, interactive maps, animated 2D/3D graphics, scrolling video jukeboxes, etc. Today JavaScript is also quickly growing as a server-side technology since the release of Node.js in 2009 [57]. We used JavaScript for writing all of the back-end code.

5.4.3 HTML

HTML stands for Hypertext Markup Language. It is the standard markup language for creating web pages and web applications, for instance defining paragraphs, headings, and data tables, or embedding images and videos in the page. We used HTML for creating the administrator application [57].

5.4.4 CSS

CSS stands for Cascading Style Sheet. It is a language of style rules that we use to apply styling to our HTML content, for example setting background colors and fonts, and laying out our content in multiple columns. We used CSS to style the administrator application [57].

5.4.5 Node.js

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world. It is used for developing server-side and desktop applications [58]. We used Node.js to build the back-end part of the system.

FIGURE 5.7: Node.js logo.

5.4.6 Express.js

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications [59]. We used Express to help organize our API code, since it's the standard server framework for Node.js.

5.4.7 MongoDB

MongoDB is an open source database that uses a document-oriented data model. It is one of several database types to arise in the mid-2000s under the NoSQL banner. Instead of using tables and rows as in relational databases, MongoDB is built on an architecture of collections and documents. Documents comprise sets of key-value pairs and are the basic unit of data in MongoDB. Collections contain sets of documents and function as the equivalent of relational database tables [60]. We created the ClickRide database using MongoDB.

FIGURE 5.8: MongoDB logo.

5.4.8 Mongoose

Mongoose is a JavaScript framework that is commonly used in a Node.js application with a MongoDB database. It is an Object Document Mapper (ODM). This means that Mongoose allows you to define objects with a strongly-typed schema that is mapped to a MongoDB document [61]. In our project we used Mongoose particularly for modeling the MongoDB database.

5.4.9 Bootstrap

Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins [62]. Bootstrap is used in our project especially for making the administrator application responsive.

5.4.10 Vue.js

Vue is a progressive framework for building user interfaces. Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable. The core library is focused on the view

FIGURE 5.9: Bootstrap logo.

layer only, and is easy to pick up and integrate with other libraries or existing projects. On the other hand, Vue is also perfectly capable of powering sophisticated Single-Page Applications when used in combination with modern tooling and supporting libraries [63]. Vue helps us to build reactive interfaces for the administrator application.

FIGURE 5.10: Vue.js logo.

5.5 System presentation

In this section, we present the main windows for both the users and the administrator applications.

5.5.1 User application

The user application allows users whether they are passengers or drivers to to post rides, search for rides, request rides, choose to accept or reject requests for a ride, etc.

5.5.1.1 Authentication

Figure 5.11 shows the login and sign up windows.

5.5.1.2 The account main window

Figure 5.12 shows the main window once the user log into the application. The main window allows the user to see the last posted rides. The toolbar consists of:

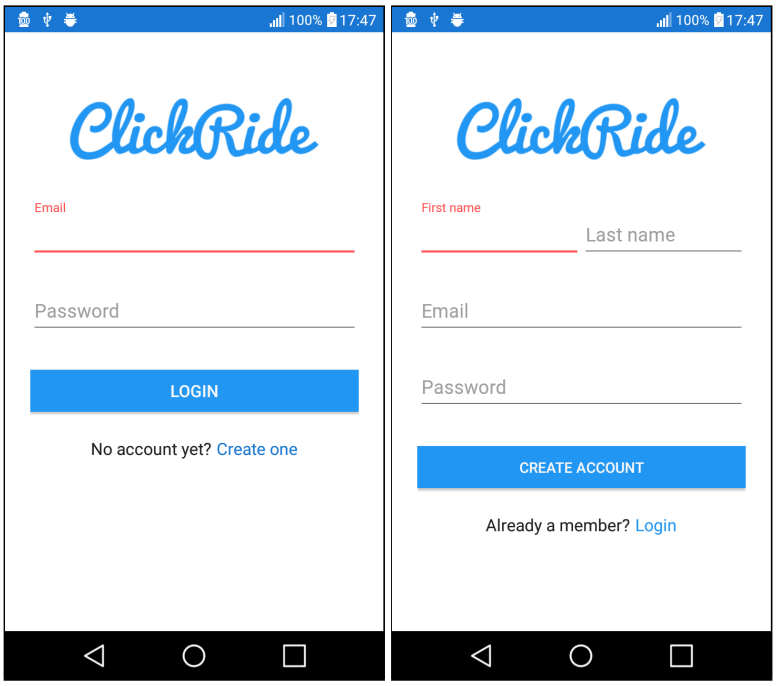


FIGURE 5.11: Login and sign up windows

- A main menu for settings and log out.
- A notifications icon for viewing the latest notifications, such as requests for rides, confirmed requests, etc.
- A search icon to search for rides.
- A navigation drawer, which contains Dashboard, Create a ride, My rides, My requests and settings.

5.5.1.3 Searching for a ride

Figure 5.13 shows find rides window along side with found rides window. Basically, the user enters the origin, and use the spinners to pick a day and the time for the ride, then tap find rides button. The rides are displayed in another window.

5.5.1.4 Adding a ride

Figure 5.14 shows post a ride window. Basically, the user enters the origin, and uses the spinners to pick a day and the time for the ride, and enters the price and the number of seats, then tap post ride button.

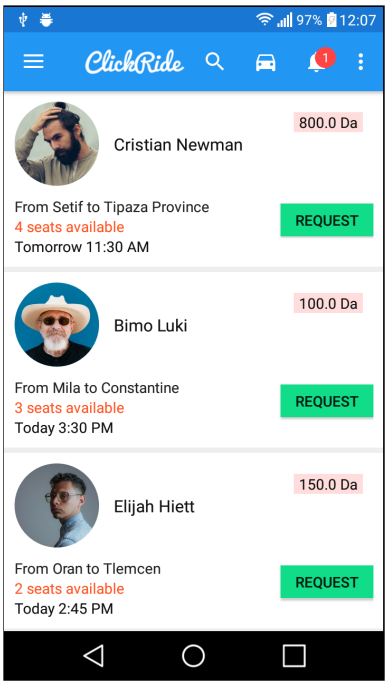


FIGURE 5.12: The main window

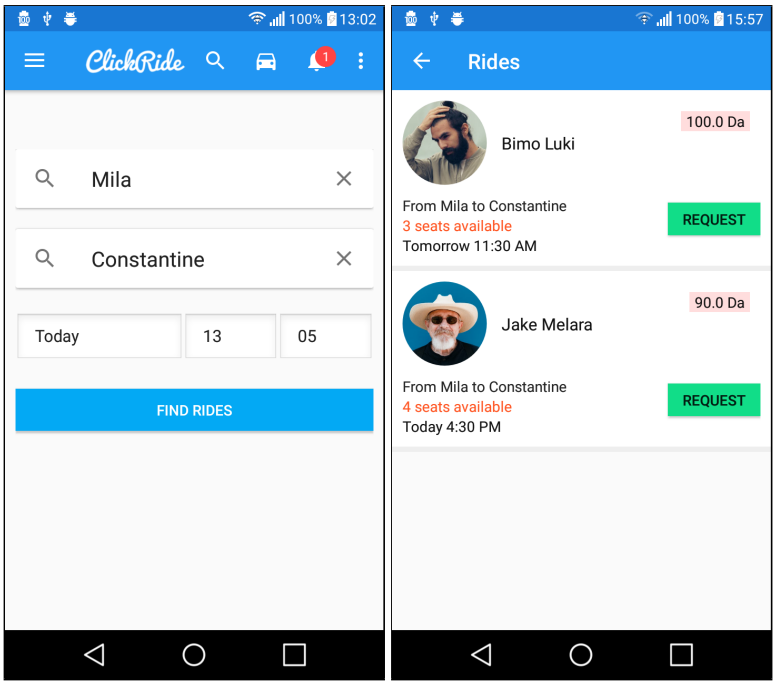


FIGURE 5.13: Find rides and found rides window.

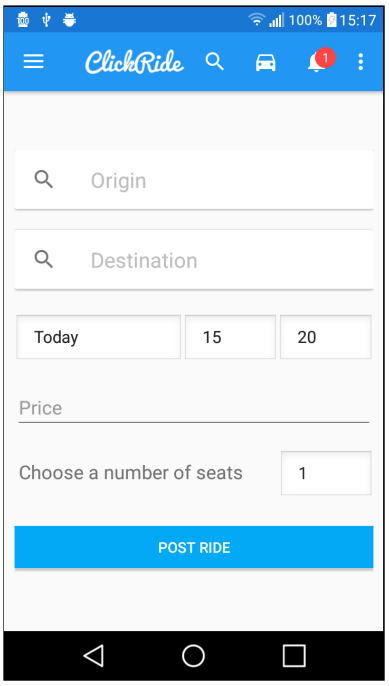


FIGURE 5.14: Post a ride window.

5.5.1.5 Sending a request for a ride

Figure 5.15 shows request a ride window along side with found a notifications window. Basically, the user tap request button, an alert dialog shows up asking the user to set the number of seats they want to request. When the user tap set button a request will be send to the driver. The driver can see requests in the notification window.

5.5.2 Administrator application

The administrator application allows the administrator to manage users and rides. The dashboard shows the last registered users and the last posted rides. The administrator click View to view details; delete to delete a user or a ride; or activated/deactivated to activate or deactivate a user.

5.6 Conclusion

In this last chapter, we have presented the implementation of our system, different tools, languages and technologies used for developing our application. Also, we presented some of the main use cases for both the users and administrator application by means of some screen-shots.

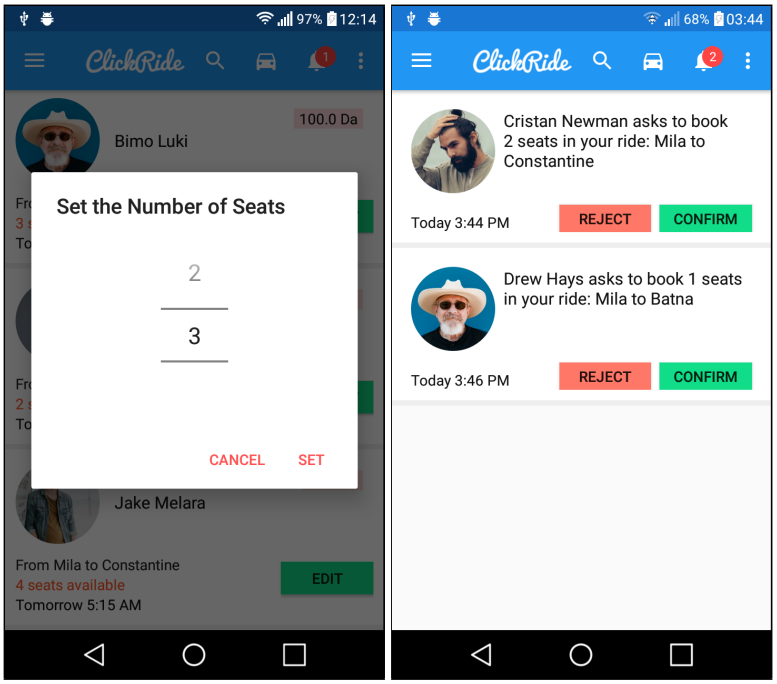


FIGURE 5.15: Request a ride and notifications windows.

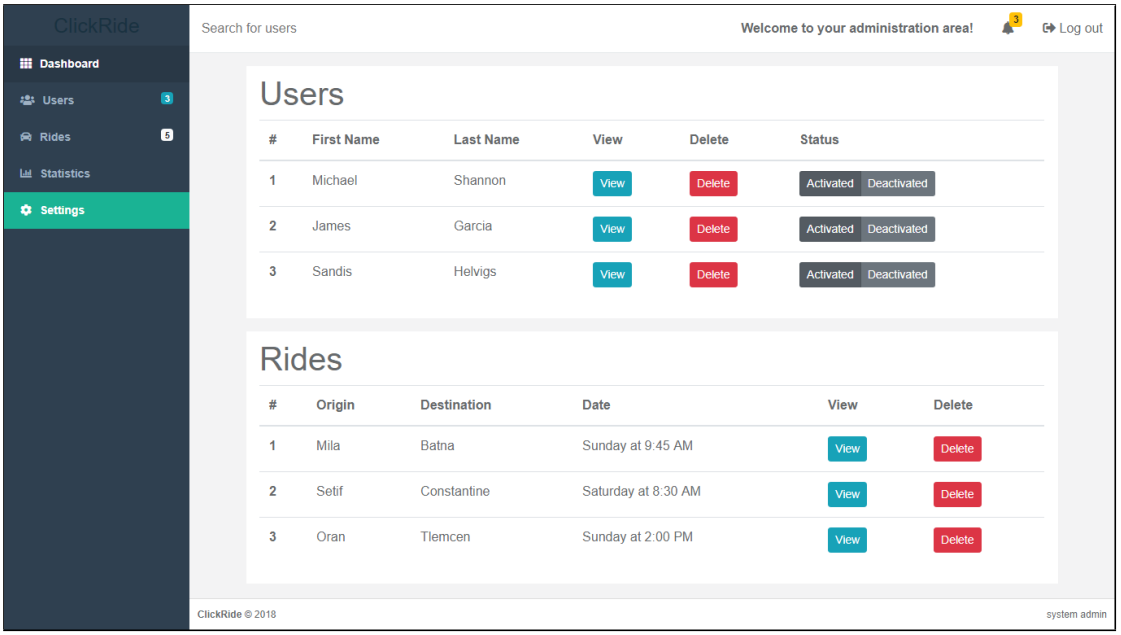


FIGURE 5.16: The administrator dashboard.

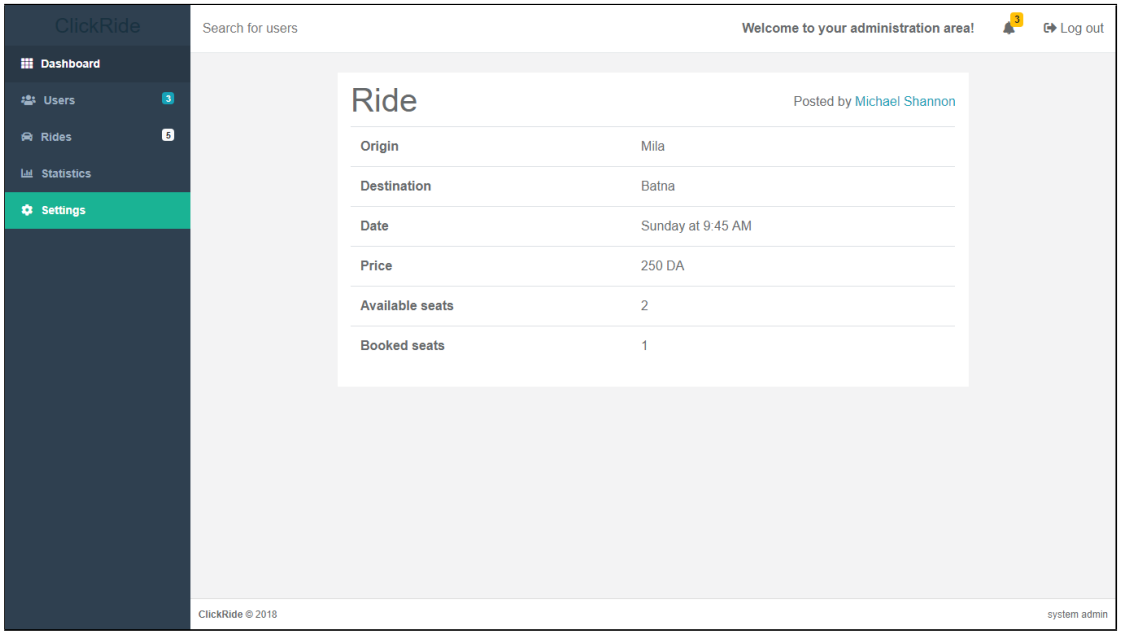


FIGURE 5.17: View ride window.

Chapter 6

Conclusion

The recent advances in Information and Communication Technology (ICT) have tremendously affected the way people travel by improving accessibility, passengers experience, capacity, road safety and reducing environmental impacts. Several transport platforms, websites and applications were specifically built to improve the overall travel experience, and made the access to these instruments within our fingertips.

The main goal of this work was to develop a platform for carpooling as a way to provide an additional and alternative mode of transportation to assist the already existing transport systems, which helps to match between passengers and drivers with common origin, destination, and time schedule. Carpooling can be of great benefits to people as well as the environment if implemented in the right manner and in a large scale. We developed a web application to be used by the system administrator and a mobile application (ClickRide) for android devices, which can be used by both passengers and drivers, eliminating the need for an additional application which can consume more resources of the user's device and can be frustrating: having to switch between two applications when the user is utilizing it for both cases (passenger and driver). ClickRide allows the users to post and/or request a ride.

There are several functionalities that we wanted to add to ClickRide, but we could not implement due to time constraints, and hence they constitute the future perspectives of our project. Among these functionalities we mention:

1. Adding the messaging functionality, by allowing users to send and receive messages
2. To implement third party authentication through Facebook and use Facebook friends as a way to establish more trust among passengers and drivers.
3. Adding the friendship functionality, as it allows people to make new friends as well as to enhance trust among the application's users
4. By developing an API for the back-end, the development of application for IOS and web application is much easier since we will be concerned about the front-end only.
5. Integrating electronic payment.

Regarding good software writing practices/Improvements we mention the following tasks for future consideration:

1. Implementing Model–View–ViewModel (MVVM) software architecture pattern, which provides a clean separation of concern and makes the application easier to test and maintain.
2. Enhancements in the overall design and user experience of the application as they play a major role to persuade users to use the application.

Bibliography

[1] Alessandro Lue and Alberto Colorni. A software tool for commute carpooling: a case study on university students in milan. *International Journal of Services Sciences*, 2(3-4):222–241, 2009.

[2] Vinícius Cezar Monteiro de Lira Lira. Exploring human activity behavior and mobility data in carpooling. 2014.

[3] Hamid Ould Ahmed. Algerian lawmakers approve budget calling for hikes in fuel prices, taxes, November 2017. URL https://af.reuters.com/article/investingNews/idAFKBN1DQ0KK-OZABS. Accessed 9 Mars 2018.

[4] GHAR Correia. *Carpooling and Carpool Clubs: Clarifying Concepts and Assessing Value Enhancement Possibilities*. PhD in Transportation, Department of Civil Engineering, Instituto Superior Técnico, 2009.

[5] World Bank. The world bank annual report 2017. Technical report, 2017.

[6] Jean-Paul Rodrigue, Claude Comtois, and Brian Slack. *The geography of transport systems*. Routledge, 2016.

[7] Nicole Rupersburg. Innovative urban transportation apps for 2018, April 2018. URL http://meetingoftheminds.org/12-innovative-urban-transportation-apps-4708. Accessed 17 April 2018.

[8] Johan Öberg et al. Connected things 2017: Smart public transport, 2017.

[9] Joe Hindy. 10 best transit apps for android, April 2018. URL https://www.androidauthority.com/best-transit-apps-android-782796/. Accessed 25 April 2018.

[10] URL https://citymapper.com/.

[11] URL https://www.uber.com/.

[12] URL https://www.lyft.com/.

[13] URL https://moovit.com/.

[14] URL https://www.easytaxi.com.

[15] URL https://transitapp.com/.

[16] Valerie. 5 best parking apps for your city, April 2018. URL https://thedroidguy.com/2018/04/5-best-parking-apps-city-1077301. Accessed 2 May 2018.

[17] URL https://spothero.com/.

[18] URL https://www.forbes.com/.

[19] . URL https://www.parkme.com/.

[20] . URL https://www.parkwhiz.com/.

[21] Clark Williams-Derry. Planes, trains, and automobiles: How we estimated emissions from vehicle travel., February 2008. URL http://www.sightline.org/2008/02/08/planes-trains-and-automobiles/. Accessed 11 April 2018.

[22] Catherine Morency. The ambivalence of ridesharing. *Transportation*, 34(2):239–253, 2007.

[23] URL https://www.blablacar.fr/.

[24] URL https://www.carpooling.com/.

[25] Ross D Eckert and George W Hilton. The jitneys. *The Journal of Law and Economics*, 15(2):293–325, 1972.

[26] Adam Hodges. 'roping the wild jitney': the jitney bus craze and the rise of urban autobus systems. *Planning Perspectives*, 21(3):253–276, 2006.

[27] Andrew M Amey. *Real-time ridesharing: exploring the opportunities and challenges of designing a technology-based rideshare trial for the MIT community*. PhD thesis, Massachusetts Institute of Technology, 2010.

[28] Nelson D Chan and Susan A Shaheen. Ridesharing in north america: Past, present, and future. *Transport Reviews*, 32(1):93–112, 2012.

[29] Maurizio Bruglieri, Diego Ciccarelli, Alberto Colorni, and Alessandro Luè. Poliunipool: a carpooling system for universities. *Procedia-Social and Behavioral Sciences*, 20:558–567, 2011.

[30] Jesús Friginal, Sébastien Gambs, Jérémie Guiochet, and Marc-Olivier Killijian. Towards privacy-driven design of a dynamic carpooling system. *Pervasive and mobile computing*, 14:71–82, 2014.

[31] Christopher DeGruyter. Investigating a cbd-wide carpooling scheme for melbourne. *Qualitative research*, 28:30, 2006.

[32] URL https://www.statista.com/outlook/368/100/ride-sharing/worldwide.

[33] Erin Carson. 10 rideshare apps to crowdsource your commute, April 2014. URL https://www.techrepublic.com/article/10-rideshare-apps-to-crowdsource-your-commute/. Accessed 29 April 2018.

[34] Kyle Wiggers. Go ahead, have another! the best ridesharing apps help get you home safely, December 2017. URL https://www.digitaltrends.com/mobile/best-ride-sharing-apps/. Accessed 29 April 2018.

[35] Julia Eddington. Ride-sharing vs. ride-hailing: What's the difference?, April 2016. URL https://www.thezebra.com/insurance-news/2811/ride-sharing-vs-ride-hailing/. Accessed 30 April 2018.

[36] . URL https://gett.com/juno/.

[37] Niels Agatz, Alan Erera, Martin Savelsbergh, and Xing Wang. Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, 223(2):295–303, 2012.

[38] Ministère des Travaux Publics et des Transports. Le réseau routier, December 2013. URL http://www.mtp.gov.dz/fr/permalink/3937.html. Accessed 15 April 2018.

[39] Office National des Statistiques. Park national automobile. Technical report, Direction des Publications et la Diffusion -8 & 10 rue des moussebiline- Alger 16.000, 2016.

[40] World Health Organization et al. World health statistics 2017: monitoring health for the sdgs, sustainable development goals. 2017.

[41] Centre national de prévention et de sécurité routière. URL http://www.cnpsr.org.dz/document_name/Statistiques_2011.

[42] Aouragh Leila and Bourmada Noureddine. Assessment of the contribution of road traffic to greenhouse emissions: A case of an algerian city. *Journal of Environmental Protection*, 5(13):1364, 2014.

[43] URL http://www.globalcarbonproject.org/.

[44] Office National des Statistiques ALGER. Compendium national sur les statistiques de l'environnement. Technical report, Office National des Statistiques, 2006.

[45] sonelgaz. Bilan energétique national 2016. Technical report, Ministère de l'Energie, 2017.

[46] Internet World Stats. Algeria internet stats and country data, May 2018. URL https://www.internetworldstats.com/africa.htm. Accessed 16 May 2018.

[47] Internet World Stats. Number of mobile cellular subscriptions in algeria from 2000 to 2016 (in millions).

[48] Grady Booch. *The unified modeling language user guide*. Pearson Education India, 2005.

[49] Alex Berson. *Client-server architecture*. Number IEEE-802. McGraw-Hill, 1992.

[50] URL https://www.mongodb.com/scale/types-of-nosql-database-management-systems.

[51] URL https://www.mongodb.com/mongodb-architecture.

[52] URL https://developer.android.com/studio/.

[53] URL https://code.visualstudio.com/docs.

[54] URL https://www.getpostman.com/postman.

[55] URL https://www.heroku.com/.

[56] URL https://mlab.com/.

[57] itsPugle. What is javascript?, April 2018. URL https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript. Accessed 18 May 2018.

[58] URL https://nodejs.org/.

[59] URL https://expressjs.com/.

[60] Margaret Rouse. Mongodb, March 2014. URL https://searchdatamanagement.techtarget.com/definition/MongoDB.

[61] Jamie Munro. An introduction to mongoose for mongodb and node.js, October 2017. URL https://code.tutsplus.com/articles/an-introduction-to-mongoose-for-mongodb-and-nodejs--cms-29527. Accessed 24 May 2018.

[62] . URL https://getbootstrap.com/.

[63] URL https://vuejs.org/v2/guide/.