الجمهورية الجزائرية الديمقراطية الشعبية
PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
وزارة التعليم العالي و البحث العلمي
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
جامعة عمار ثليجي بالأغواط
UNIVERSITY OF LAGHOUAT
كلية العلوم
FACULTY OF SCIENCES
DEPARTMENT OF COMPUTER SCIENCE

Master degree

Field: Mathematics and Computer Science

Option: Computer Science

Specialization: System and Computer Science Network

Submitted by :
M. Zidelkhir Ali

Theme

Design and implementation of crowd sourcing based delivery system

publicly presented on 08-07-2021 in front of the jury composed of:

M. Allaoui Tahar	Professor	President
M. Ameur Mohamed El Amine	Professor	Examinator
M. Bensaad Mohammed Lahcen	Professor	Supervisor

2020/2021

Acknowledgments

“

First of all, all praise and gratitude are due to Allah, for giving me the will and strength to try finishing this work.

I would like to express my gratitude to my parents and my sisters for their unfailing emotional support.

I also would like to express my sincere gratitude to my supervisor Dr.Bensaad Lahcen, for his support and the amount of time and effort he put into guiding me through the duration of this project.

I would also like to thank all my teachers that have given me the knowledge required to complete this work and achieve this moment. This work is dedicated to all of you.

Thank You.

”

- Ali

Abstract

The demand for transportation is at an all time high and is continuing to increase with time.

We all agree to the importance that transportation holds in maintaining the flow of goods and cash as well as its role in the economy.

Crowd sourced delivery is considered a possible solution for complicated problem with on-demand delivery challenge. The purpose of this thesis is to propose the solution of the crowd sourced delivery with regard to service level and quality.

The aim of this work is to create a platform that organizes the process of crowd transportation model and helps both drivers and possible clients to better interact. Drivers will be able to save the time, minimise the cost of fuel and avoid the traffic jam. Also have the willing to work at any time.

One of the most factors this project will focus on, is the timing and pricing for the clients. using existing method to use in order to solve this problems. The Clients will be able to post their requests that the drivers will handle it in the way of his trip.

Key Words : Optimisation, VRP, Vehicule Routing Problem, Genetic Algorithm, Simulation app, Crowd Sourcing, Crowd Delivery, Delivery Parcel.

ملخص

يشهد قطاع النقل والتوصيل الآن انتعاشا كبيرا في الآونة الأخيرة.

ومع احتياج الأشخاص لخدمات التوصيل في أي وقت، مع توفير السرعة في التوصيل، وتقديم السعر الأمثل لتلبية الطلب، تشهد الكثير من الشركات مشاكل مماثلة لـ : التأخر في التوصيل، السعر المرتفع في بعض الأحيان، سلامة الطرد حين توصيله..وغيرها.

باستعمال طريقة التعهيد الجماعي، وباستخدام قوة المجموعة للمشاركة في حل مشكلة التوصيل في أسرع وقت، مع الأخذ بعين الاعتبار الجانب الربحي.

الهدف من هذه المذكرة هي إنشاء منصة مشتركة للزبائن والسائقين لتنظيم عمليات التوصيل، من خلال التعهيد الجماعي، سيكون بإمكان السائقين العمل بحرية، وتوفير الوقت وتقليل استهلاك الوقود من خلال تقديم أفضل مسار للتوصيل.

نسعى من خلال هذا العمل أن نقدم حلا مقترح لتطوير هذا النموذج في الجزائر ليكون مستقبلا عبارة عن مؤسسة ناشئة، وبدراسة العديد من الطرق والخوارزميات الموجودة حاليا والمطورة، والتي يمكن استغلالها في هذه المنصة لتقدم الحل الأمثل، وللتحسين من جودة التوصيل وتحسين التسعير المعتمد في غالب الأحيان.

الكلمات المفتاحية: التحسين في خدمات التوصيل، الخوارزميات الجينية، مشكلة تحسين مسار السيارات، استخدام الحشود لحل المشكلة، تسليم الطرود

Contents

List of Figures

Introduction

General Presentation

Currently in Algeria we are still missing a real fast service that can handle the efficiency of good delivery. Even with the condition of good pricing. Many companies faced such competition like the huge distance of the national area, in order to hire employees that can cover this diversity. When a customer orders an item by calling the phone number to a seller (or contact him throw the social media) who is located in another state (Willaya), after confirmation the customer pays the seller by bank transfer and that by passing throw some steps such as:

- Passing the Total amount (Cost + Shipping) by being presenting at the nearest Bank.
- Waiting until he pays and get the receive (CCP transfer paper).
- Take a shot for the receive and send it to the seller to confirm.

After all those steps, the seller packs the items and takes it to the central bus station where buses go from there to all cities in the country (in other cases the Taxis Station). He looks for a bus going to the state (Willaya) of the seller and gives the parcel to the bus driver to take it with him. The seller (Sender) gives the phone number of the bus driver to the customer (Receiver).
When the driver is near to the city of the receiver, the bus driver calls the receiver to wait for him in the central bus station of that city. The receiver comes, when the bus arrives and hands the parcel to the receiver.
Within all those costing steps such like:

1. Contacting the Seller.
2. Passing the Total amount throw the bank.
3. Take picture of the receive and send it to the seller to confirm.
4. Preparing the package.
5. Search for the matching city Bus, which may not be available.
6. Negotiate the price of delivery with the driver.
7. Take the phone number.
8. Re-contact the client to give him the information.
9. Inform the client to be present at time when the Bus/Taxi arrive.
10. Handle the package to the client.

In somehow, there is a new companies here in Algeria start doing the buses work, by handling the delivery job of any type parcels, which will offer the clients an easy way to visit their center office to pick the item they ordered.

Problem

The service of delivery needs to be improved and came up with innovative idea to handle the problem with the simplest way, and the most reasonable price that fit the client first, and offering an online solution like a phone application to let the business/client free to choose.
In the last 5 years, many companies in Algeria Start to come up with the service of parcel delivery, such EMS, YALIDINE. Giving the chance for the people and all who have online stores to send their packages easily in the best time, with an average pricing, it could be more expensive than the parcel itself in some cases.
Hiring only few people who have specific type of transporting cars, like (Berlingo, Master, Express, Kongoo... etc.). Also, there is many people who will ask company to deliver parcel directly home.
This model called Business-to-Client (B2C). A few companies offer also the front door delivery, but only for few drivers can work so.

Solution

Client to Client (C2C) delivery model (also known as crowd shipping or crowd logistics) starts with a request from client to deliver a parcel. Other users can choose to deliver selected parcel to had the opportunity to pick it on the way of their destination, this service will give the chance to many people having any type of transportation vehicle.
The drivers can have their own job or side-job while they are traveling through many states or in the local region. For the clients, they will get the best deal with the best pricing and the fastest timing.
Despite the great demand on delivery services, online retailers have not yet had a practical solution to make their business profitable and grow more. This going to make the company we are going to start as the best solution in market, having side profits for its services, and offering the best value.

Structure of Manuscript

The reset of this thesis is organised as follows:

- In the first Chapter we give a general definition of delivery system and why it is important. We will compare between existed companies in Algeria, and their strategy of works. Also introducing some related works and comparing them from various aspects. Also using some important algorithms that will help the project.
- In the second Chapter we provide details about the functionalities of our platform as well as the tools used in the design and development of the project as well as diagrams to further explain how we conceived this work.
- In the third Chapter we describe the tools and platforms used for implementing our system and why we chose them, we also explain the functionality of our system and we include screenshots to further elaborate and clarify the system's functionalities.
- Finally we will finish by giving a general conclusion summarizing the different steps in completing the project and the possible routes that we could take in furthering its development.

Chapter 1

Background & Related Works

In this chapter we are going to talk about the Transportation field and its importance, as well as provide comparison to back up our point, then we are going to see some related works.

1.1 Introduction

Transporting is one of the most important systems in the world. It consists of moving people, goods, parcels, and packages from point A to point B. This system has evolved through the last 20 years in Algeria by helping the Startup and the company to create their own solution to improve the quality of the transportation services. In this thesis, we will focus on crowd shipment, specifically in the case of parcels and packages, which is going to be a standard API can be used for any company in the future. Distribution of goods, products, parcels, and packages is one of the most important processes, because it involves two parties; the one who run his own respective business and the receiver. Sometimes several problems occur during the delivery process.

1.2 The importance of parcels delivery

Parcel's delivery is the dominant form of transport especially at the local and national levels. It is widely available for each one who has a car or motorcycle at least. Especially with in the development of the merchandise of the startup and the small home business, so that the client become in need of those services, which offer the best price and timing as a priority. After that another need show's up, which is the need of side work or full-time work, which offer any one who has a vehicle or a van to offer his service of transportation, and participating with the project.

1.3 Usage of buses and vehicles to deliver the parcels

In the last 10 years, most of the stores in Algeria start to promote their business throw social media (Facebook, Instagram, Web pages…), which make people excited to buy from them, especially if they are not from the same region (Willaya), so the best solution for that was delivering by buses and private taxis, which make the local business reach more and more people. This strategy became the most famous one until it became full of problems, like high pricing, the guaranty of delivery parcel, the timing, and so on.

1.4 Vehicle routing problem

The vehicle routing problem (VRP) is a combinatorial optimization and integer programming problem which asks "What is the optimal set of routes for a fleet of vehicles to traverse in order to deliver to a given set of customers?". It generalizes the well-known travelling salesman problem (TSP). see Figure 1.1

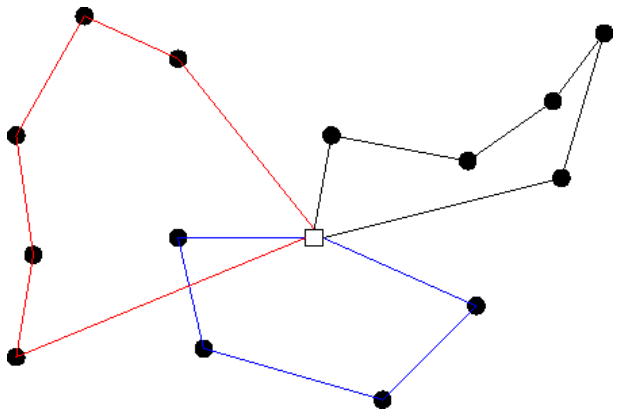

Figure 1.1: illustrating the vehicle routing problem

In the Vehicle Routing Problem, the goal is to find optimal routes for multiple vehicles visiting a set of locations. the context is delivering parcels stored in the office point to the customers. Determining the optimal solution to VRP is NP-hard, so the size of problems that can be solved, optimally, using mathematical programming or combinatorial optimization may be limited. Therefore, commercial solvers tend to use heuristics due to the size and frequency of real world VRP they need to solve.[21]
In the VRP problem there is some factors that we need to consider:

- Network type; oriented, non-oriented.
- Type of service: pickup, delivery, or both.
- Depository; one or more.
- Maximum duration of pickup trip: define the time for the driver to start from the depository point to gathering point.
- Cost of the trip; limited, unlimited.
- Serving time for one client.

- Fuel consumption.
- Traffic peak time.
- Construction, and bad roads. e a shot for the receive and send it to the seller to confirm.

1.4.1 Vehicle Routing Problem with Time Windows (VRPTWs)

Customers often demand that their deliveries should be made during a specific period. This limits pick-up and delivery times, as the driver has to show up at a customer's door within the requested time or before.
So, we need to make use some factors in time windows in the most cost-efficient way. [11]

- Soft time windows: Serving outside the time window is allowed, but it comes with penalties.
- Hard time windows: Time violations are not allowed at all. A driver must wait if he or she arrives way ahead of the scheduled time until the time window opens and cannot show up late.
- Disjoint time windows: Showing up between two-time windows, a driver has to wait until the next time window opens.
- Multiple time windows: A set of non-overlapping time windows with different lengths.

The data include a time matrix, which contains the travel times between locations. see Figure 1.2

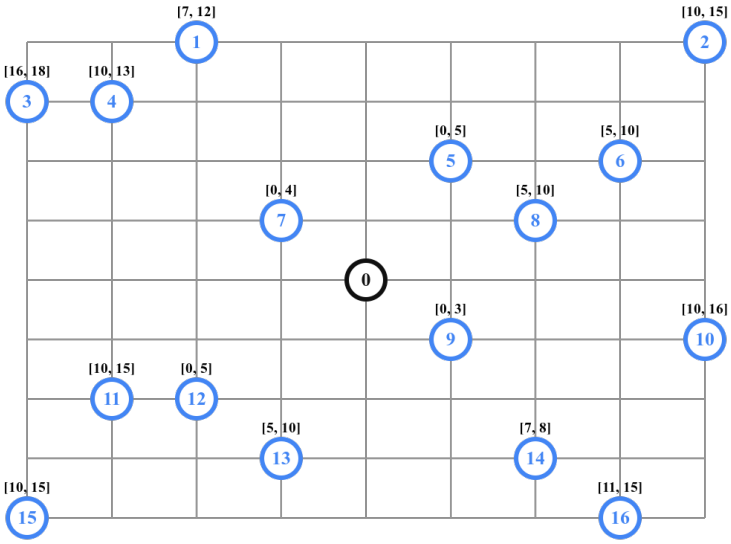

Figure 1.2: VRP with time windows, where the blue circles cited with the location (x,y) above

1.4.2 Pickup and Delivery Vehicle Routing Problem (PDVRP)

On-demand delivery businesses, such as parcels delivery and food delivery companies, need to plan delivery routes multiple times a day, depending on the nature and scale of the business.
Several resource constraints, parameters, and schedules need to be considered. Usually, with PDVRP the challenge is combining delivery and pickup points to help reduce travel time and fuel costs.

1.4.3 Capacity Constraint Vehicle Routing Problem (CVRP)

Each vehicle has a maximum load capacity (weight and volume) which must be considered. So, it is challenging to save costs by loading more items and serving more customers in one trip without exceeding the vehicle's capacity. There might also be additional complications, such:

- The varying sizes of parcels for delivery and pickup,
- The different capacities of all the vehicles.

1.4.4 Methods of solving VRP and its variants

To solve a vehicle routing problem, it is important to take the following two steps, the first is to model the problem, the second step which depends on the mathematical modeling established, consists in choosing a method of resolution and to implement it.
Several resolution methods exist, with which an optimization problem can be solved in an exact or approximate way. We present a classification of some of these methods in the figure below [3]: see Figure 1.3

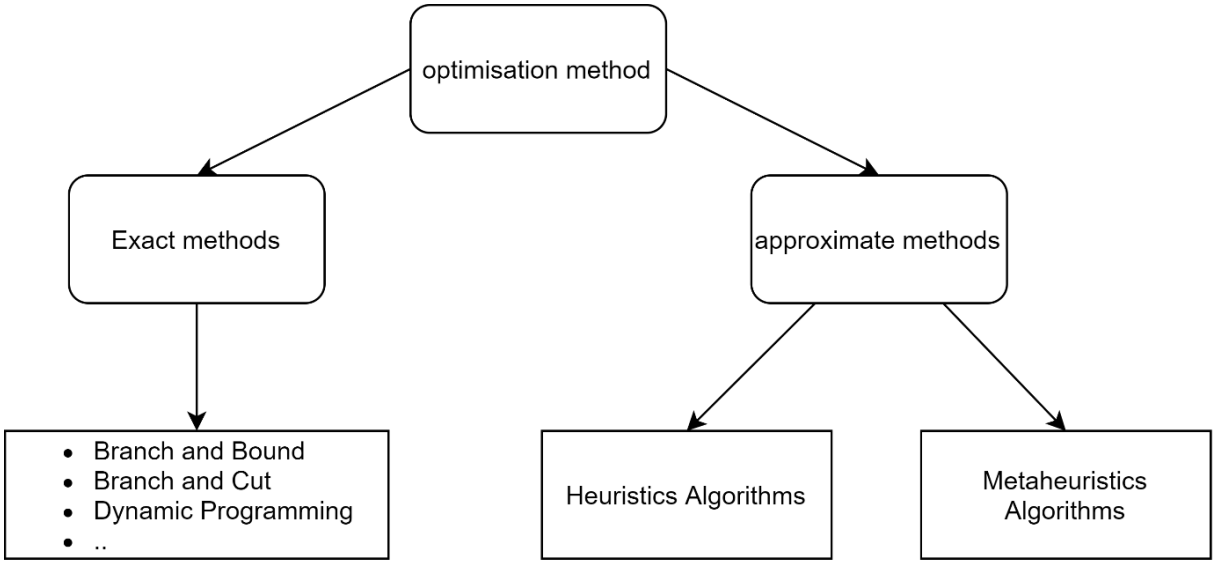

Figure 1.3: Vehicle Routing Problem method classification

1.4.5 Exacts Methods

These methods will make it possible to find the optimal solution

Branch and Bound

Is an algorithm design paradigm which is generally used for solving combinatorial optimization problems. These problems are typically exponential in terms of time complexity and may require exploring all possible permutations in worst case.
The Branch and Bound Algorithm technique solves these problems relatively quickly. This method has been widely used to solve many VRP variants. This method sill represents the state-of-the-art.[12]

Branch and Cut

The combination of an algorithm of cuts and a Branch and Bound makes it possible to obtain the algorithm of Branch and Cut where the nodes of the tree are evaluated using a cutting algorithm.
The cutting algorithm is a continuous relaxation of the problem, this relaxation allows that some variables are continuous. If after resolution, all the variables of the solution obtained are integers, then the problem is solved. Otherwise (and it is usually the case), the search space for solutions is going to be reduced.

Dynamic Programming

Dynamic programming (DP) has been applied to several types of VRPs. we need to solve different parts of the problem (subproblems), then combine the solutions of the subproblems to reach an overall solution. Often when using a naiver method, many of the subproblems are generated and solved many times.
The dynamic programming approach seeks to solve each subproblem only once, thus reducing the number of computations. DP has been used successfully to solve some VRPs to optimality or to obtain very sharp bounds on the value of their optimal solutions.

1.4.6 Approximate Method

When dealing with large-scale problems, the use of approximate methods is necessary in order to produce a workable solution of good quality which is not necessarily optimal but in a reasonable computation time.
We will present the two classes of these methods, heuristics and metaheuristics.

Heuristic Methods

In mathematical optimization and computer science, heuristic is a technique designed for solving a problem more quickly when classic methods are too slow, or for finding an approximate solution when classic methods fail to find any exact solution.
Heuristics aim to develop satisfactory solutions by a reasonable time, many of these methods have been developed for the VRP and its variants.

Tabu Search: The problem of simulated annealing is that after "jump" the algorithm can simply repeat its own track. Tabu search prohibits the repetition of moves that have been made recently. This algorithm will suppose a random value of solution only in the first time, after that it will move to the next state and regenerate new moves, after getting a better solution, the oldest will be ignored. see Figure 1.4

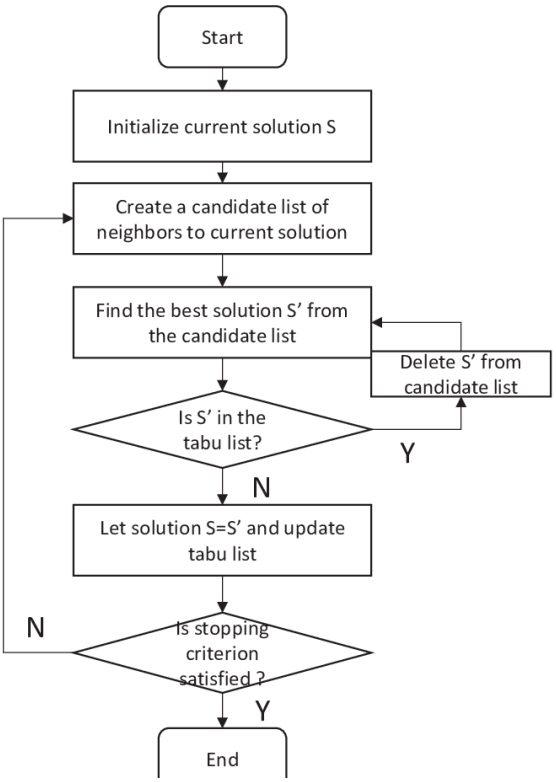

Figure 1.4: Tabu Search in VRP [20]

Metaheuristics Methods

Several metaheuristics have been proposed for the variants of VRP. The objective is to use many components and explore a large solution space even by allowing infeasible solutions. Despite they use more calculation time in general, they hit better solutions. While some algorithms start from an initial solution and try to move iteratively using neighborhoods, others use a population of solutions and generate at every iteration a new solution based on some recombination's so that at the end the best part are kept during generations.

Genetic Algorithm

In computer science and operations research, a genetic algorithm (GA) is a metaheuristic inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms (EA). Genetic algorithms are commonly used to generate high-quality solutions to optimization and search problems by relying on biologically inspired operators such as mutation, crossover and selection.[3]

- Chromosome representation and Initialization
  The individuals of a population in the GA can be seen as an ordered list of artificial chromosomes where every chromosome represents a route a truck is going to take. Each chromosome contains K integers, where K is the number of genes. A gene itself is and integer as well and represents the number of customers. Example of a solution of 4 trucks with 10 customers [13]:
  route1: [2 4 9 10] (served by truck1 to customer 2 until 10)
  route2: [4 6]
  route3: []
  route4: [3 1 6 7 8]

- Selection
  During each successive generation, a portion of the existing population is selected to increase a new generation. Individual solutions are selected through a fitness-based process, where fitter solutions are typically more likely to be selected.

- Crossover
  Crossover is a genetic operator used to vary the programming of a one or more chromosomes from one generation to the next. It is an analogy to reproduction and biological crossover. it takes information from one individual and insert it in the other to create a new child.

- Mutation
  Mutation is a genetic operator used to maintain genetic diversity from one generation of a population of chromosomes to the next. It is analogous to biological mutation.

- Fitness
  To choose which children from the newly created generation will be favored to increase, the fitness of every individual has to be computed. see Figure 1.5

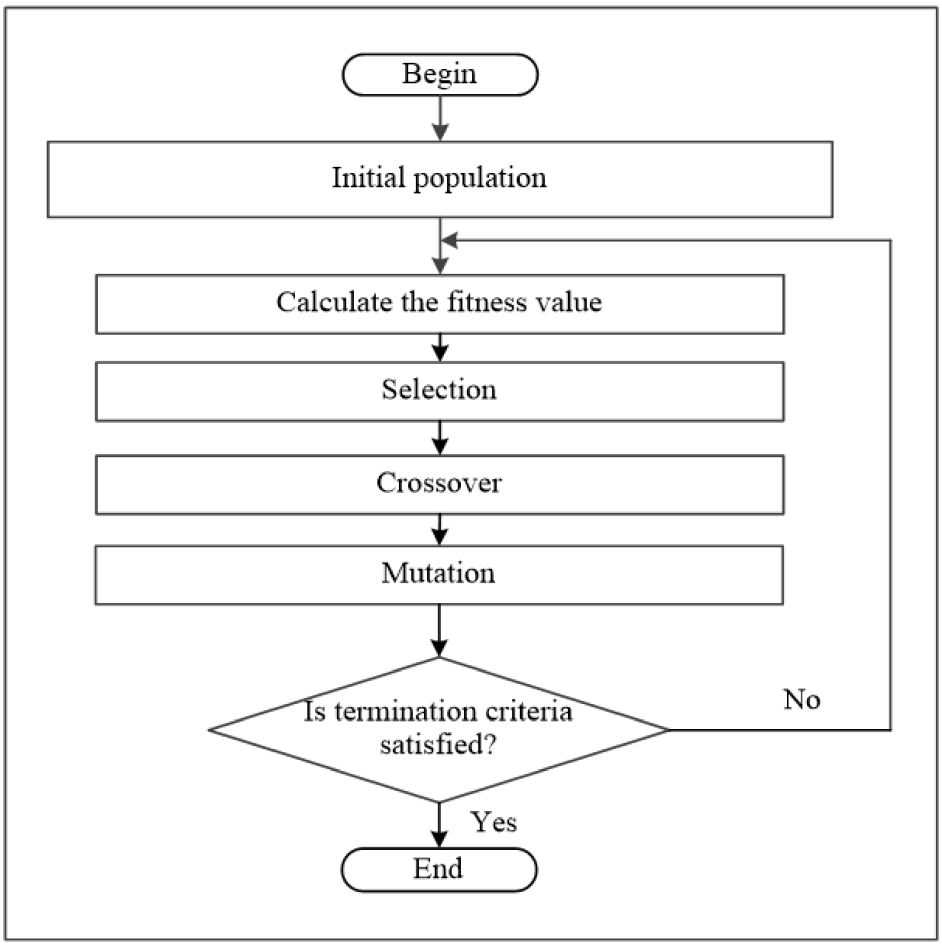

Figure 1.5: Genetic Algorithm Representation

1.4.7 Basic VRP example

Here is an example of a VRP in which the goal is to minimize the longest single route visiting the client only once. With predefined data will be given to the algorithm:

- Distance matrix for each point to other one.
- Location coordinates (x,y) so that we can calculate distance.
- Number of vehicles.
- Number of locations need to be visited.
- The index of the depot/office where to start and go back to.

Suppose 4 driver in our company needs to pick parcels of set of customers in a city. A diagram of the city is shown below, with the company office location marked in red (0) and the locations to visit in blue. see Figure 1.6, Figure 1.7

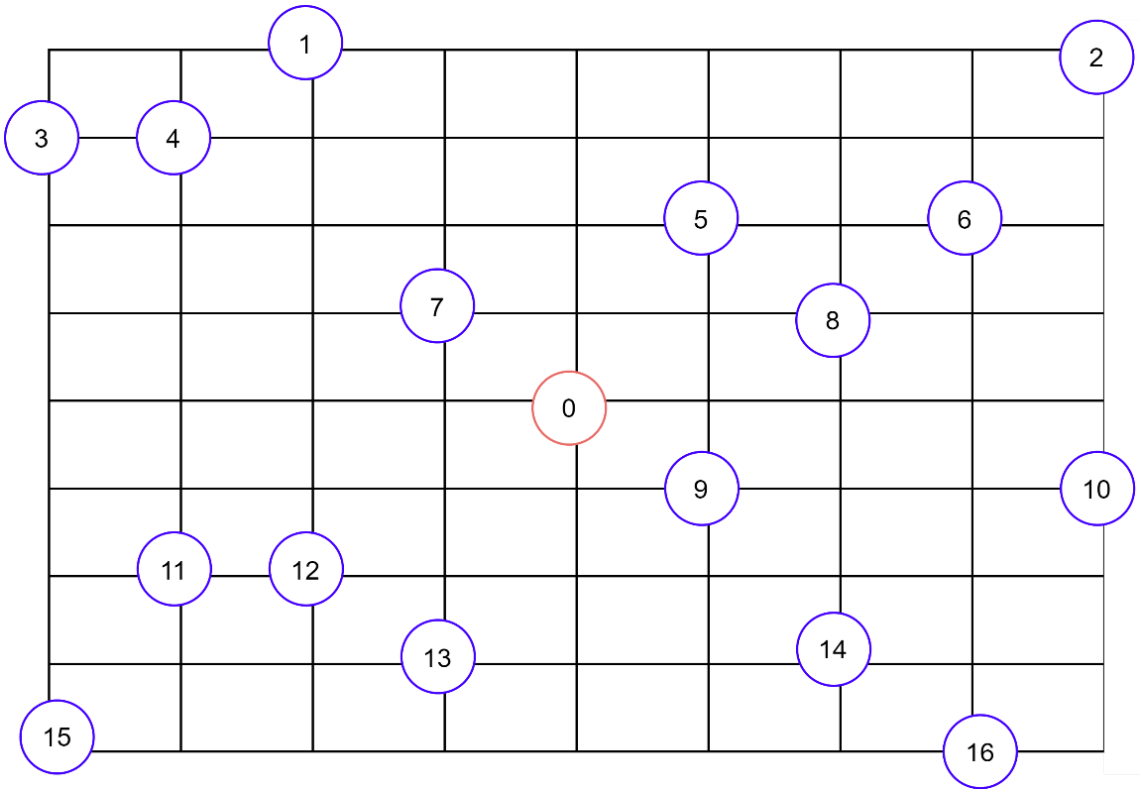

Figure 1.6: example of VRP

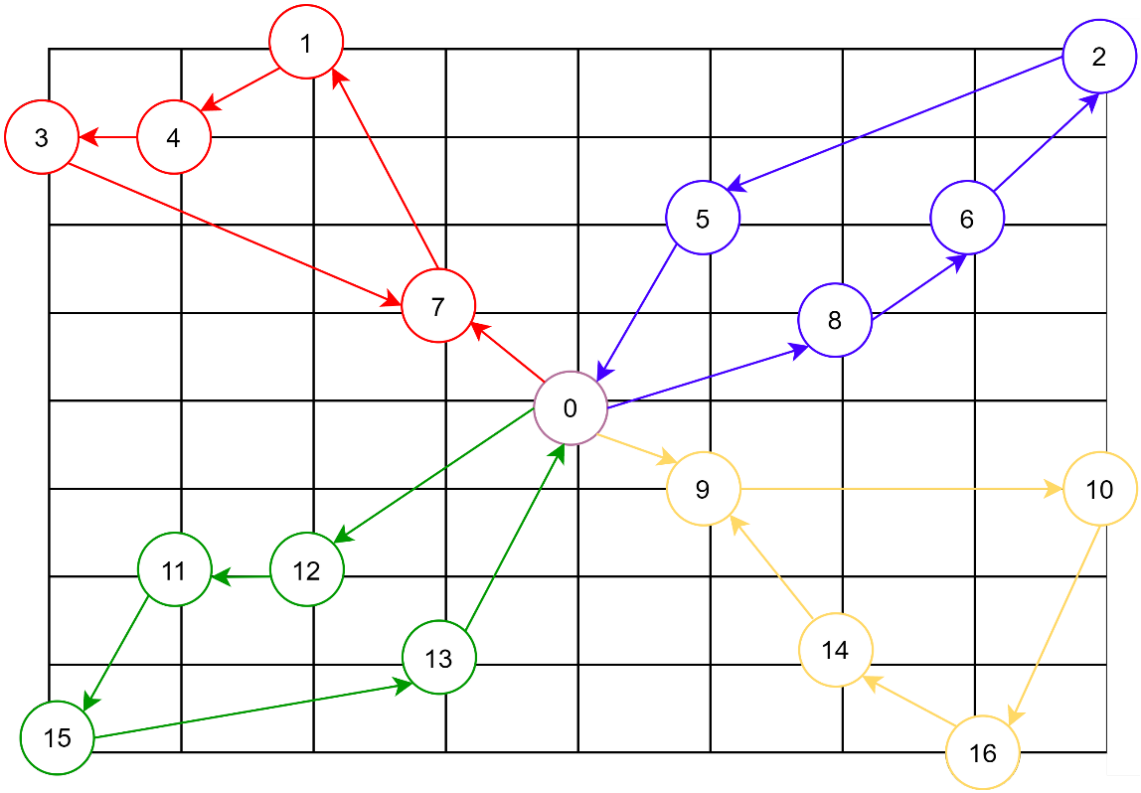

Figure 1.7: Basic Solution of VRP with 4 vehicles

1.5 Crowd Shipment for Parcels Delivery

Crowd sourcing is the practice of using the wisdom of a group for a common goal, it is best applied when attempting to solve complex problems in an innovative way or streamline intricate processes[24]. Crowd sourced delivery or Crowd Shipment is one of the most efficient method to increase the service of delivery, which handle the non-professional couriers and packages to deliver it to customers doors, while it is most common in food and grocery delivery, the crowd model is springing up everywhere like persons pickup, shared travels car, shared rooms, and taxi rides.[5] see Figure 1.8

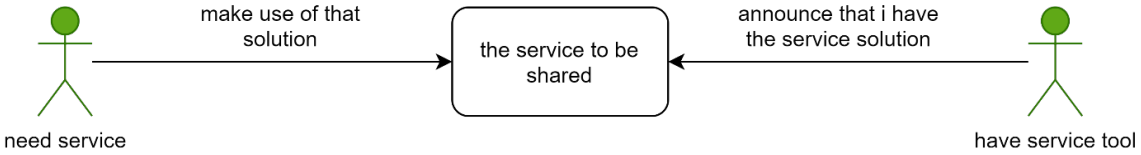

Figure 1.8: General Crowd Delivery Schema

Crowd Shipping presents an innovative shipping alternative that is expected to improve shipping efficiency, increasing the service quality, and decreasing the costs for the customer, and such shipping promises to enhance the sustainability of the transportation system.[14]
In this thesis we are studying the efficiency of the crowd delivery service for parcels, by applying this model in a form of application that can be used by anyone have the service, and anyone need the service solution.
In our case, drivers' value of free time, and willingness to work (WTW) are major advantages. In Crowdsourced shipping system, drivers have a variety of settings (time, ability, satisfaction…).

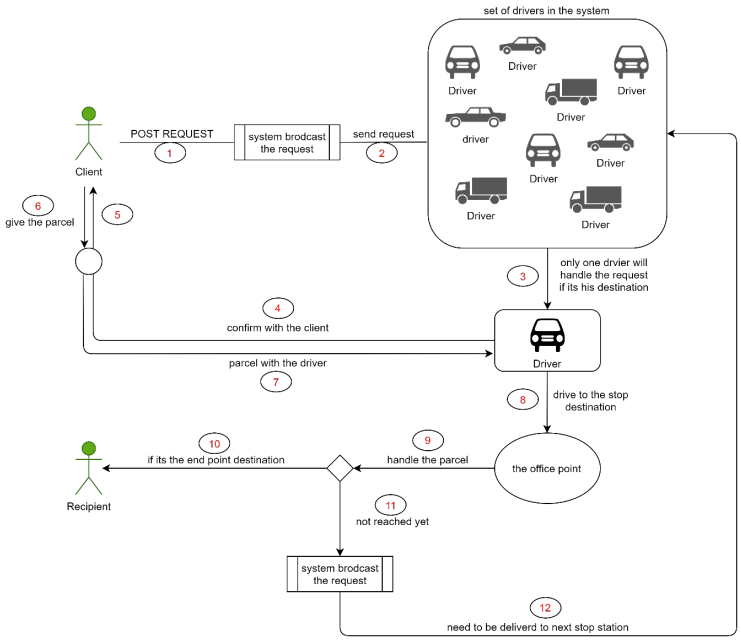

Figure 1.9: how crowd delivery model work

1.6 Existing Delivery Company and their factors

1.6.1 EMS

EMS is an international Postal Express Mail Service, for documents and goods, offered by postal operators of the Universal Postal Union (UPU).[15] The EMS Champion Post Algeria, a 100% subsidiary of Enterprise Algeria Poste, has initiated a process of modernization and development which allows it to offer its customers a quality service at competitive prices. see Figure 1.10

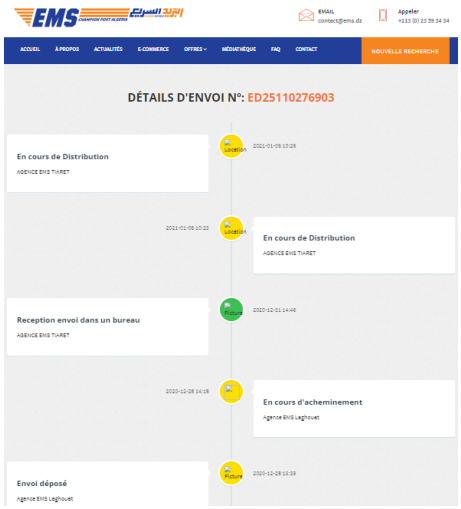

Figure 1.10: screenshot of tracking option of EMS

The EMS services comply with the International Norms and Standards set by the EMS-Cooperative of the Universal Postal Union (UPU), which our company is affiliated as a permanent member. As a service intended to meet defined needs, the Champion Post service has the following advantages:

- Speed, reliability and security, electronic shipment tracking.
- Customizable services according to needs, a traceability system and reliable feedback (IPS Track and Trace).

EMS, Work with 48-58 Willaya's, and having 34 Relay Point in Algeria, and 301 office.
EMS now became the first rewarded company in Africa, for her speed and guaranty services, in the national and international area, offering also tracking option, which was the first option that have been activated in Algeria.
The tracking option start from where the client gives his own package to the office, and then with unique code, he can follow all steps of the package, starting from the van until the client receive it.

How EMS works?

see Figure 1.11

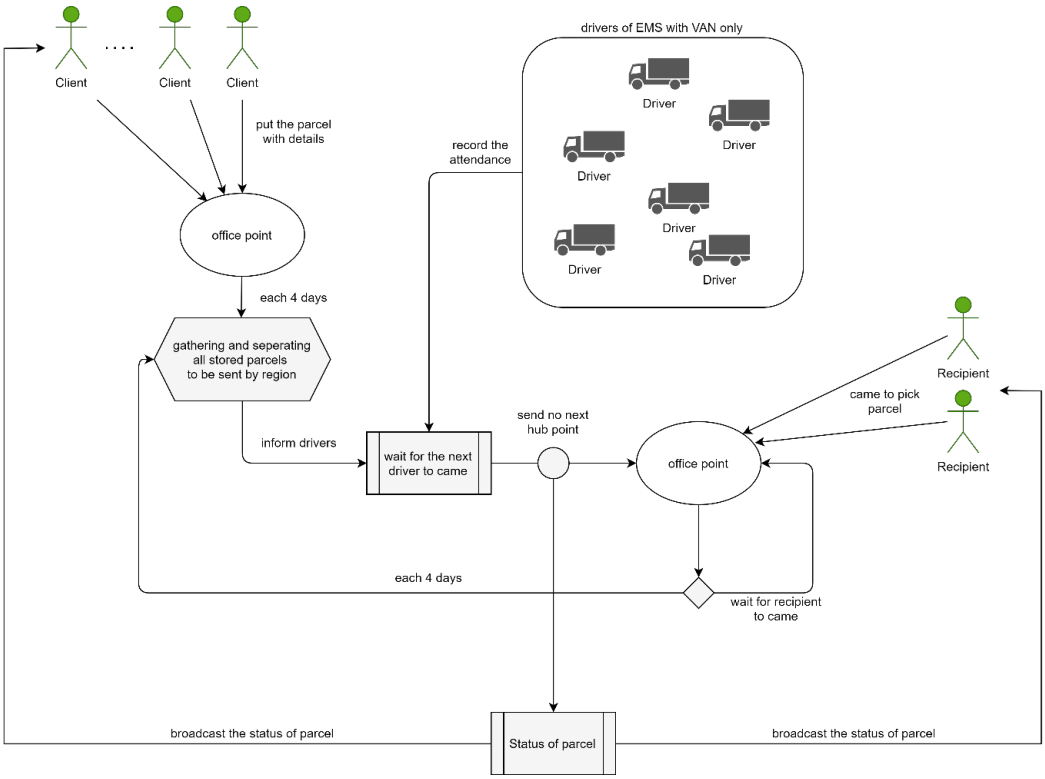

Figure 1.11: Strategy work of EMS

1.6.2 YALIDINE

Using B2B from different sectors of activity: pharmaceutical and Para pharmaceutical (medicaments and instruments for general and specialized surgery), optics (medical glassware and lenses), biological samples, industries (transport of samples), specifications, submission of tenders and various documents... etc.[16]
This company is type of Start-up Company since 2013, having only 1 sorting center in Algiers, and 19 sub-centers in the regional area, and almost 48 office in each Willaya.
This company based on specific type of delivery, which is choosing specific van to work with them only, and very few agent vans in each region.
Yalidine now offering the tracking option from the deposit until the client will receive it.

How YALIDINE works?

see Figure 1.12

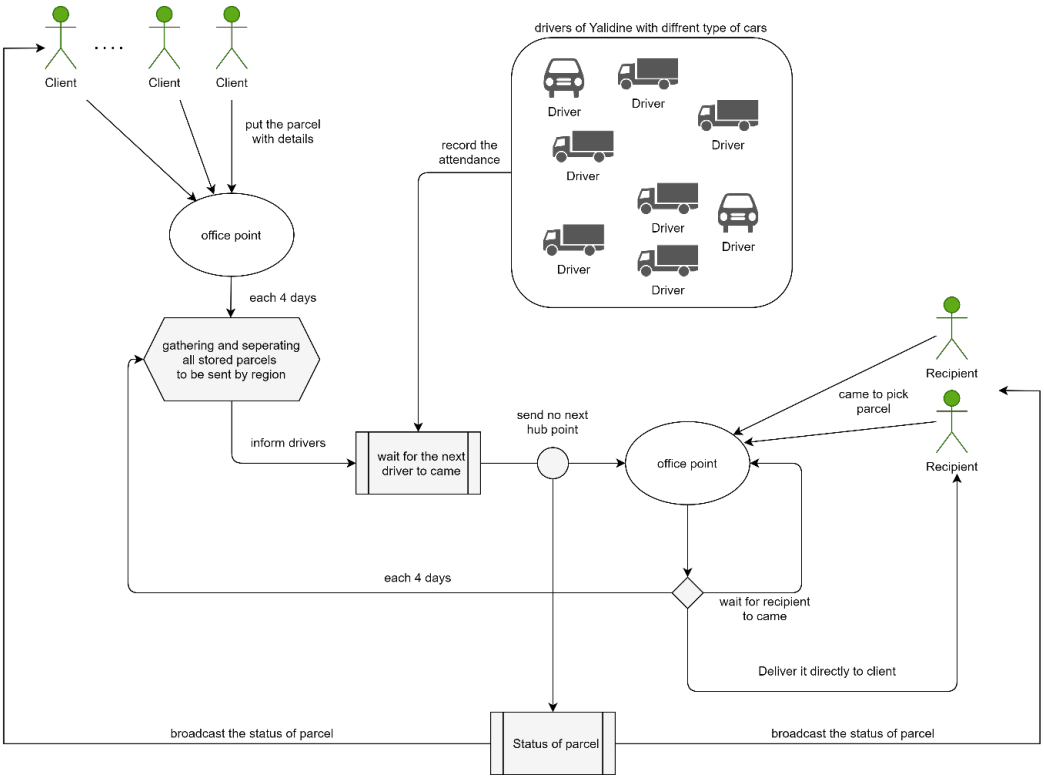

Figure 1.12: Strategy work of YALIDINE

1.6.3 ZIOUANE EXPRESS

Ziouane express Specialist in tailor-made delivery knows perfectly how to meet the needs of brands, stores, businesses and e-commerce, by offering a quality, flexible and transparent service to its partners, offers these end-customers a unique experience, and taking also the urgent deliveries, whether local, regional or national.[17]

How ZIOUANE-EXPRESS works?

see Figure 1.13

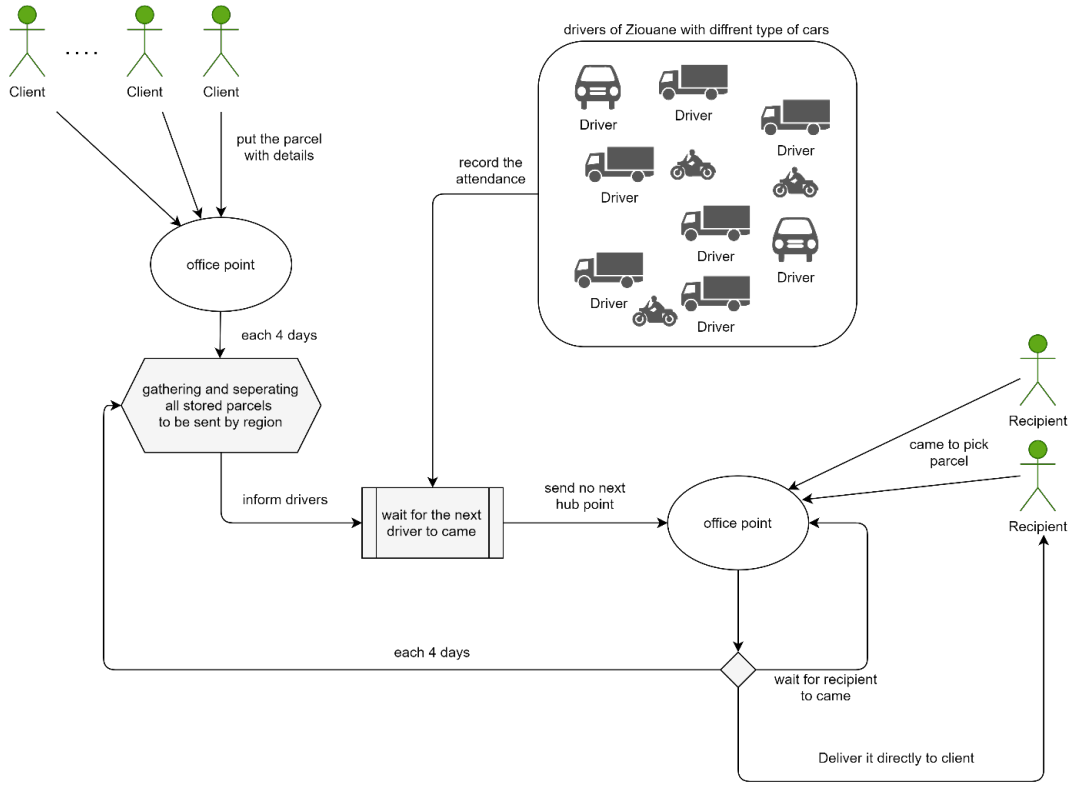

Figure 1.13: Strategy work of ZIOUANE Express

1.6.4 Comparison between the delivery's company

The important factors that most of companies focus on, defining the business model of the company, and its advantages of work. Here are some of those factors:

- Efficiency of delivering the parcels.
- Best timing to deliver the parcel.
- Distance between the destination and pickup point.
- Minimum risks of losing parcels.
- Packaging.
- Tracking option.
- Covering most of the national states.
- Option of grab the parcel from the office, and home delivery.

From those factors, each company can cover most of its clients, by offering best timing, pricing, and tracking. see Figure 1.14

	min price	price base	deliver in	max time	Willaya's cover	condition	track	service area	option delivery	phone app
EMS	840	weight + distance	4	21	48	free	available	national + international	office	none
Yalidine	300	weight + distance	2	19	48	free	available	national	delivery + office	none
Ziouane express	400	weight	5	19	48	free	none	national	delivery + office	none
Nord et ouest express	none	contract	2	9	48	commerce register	none	national	delivery	none
Kazi Tour	200	weight + distance	1	5	20	free	available	national	office	none

Figure 1.14: Table represent the most famous company of delivery in Algeria and their strategy of work

In this table we notice that the average price of delivery starting from 300da up to 800da. Until this moment, most of these companies have only a web site for visitors. They miss a smartphone application to present their work and facilitate the client role. One of the most important conditions to offer a work for someone who would like to deliver with these companies, having special type of van. Having their own Sticker on it with the Logo of the company, which is very hard for most of the people, preventing them to have a willing to work with their own car to offer their services.

1.6.5 Similar Crowd Sourcing Solutions

CARGOMATIC

CARGOMATIC, is a private company in USA, created in 2013, the platform made to solve the inefficiency and fragmentation of the local trucking industry, simply by connect shippers with qualified drivers.see Figure 1.15
(The solution based only on trucks delivering to any type of clients)
The company based on the model of On-Demand solution to reach more people having the best service, and drivers to having their own job.[4]
The focus of company is about:

- Capacity utilization; Maximizing carrier truck capacity to reduce empty runs.
- Open chance; Allowing truckers to do additional jobs, or owner operators to be their own bosses.
- Drayage; Minimizing wait times to relieve congestion.
- Less than truckload shipping; Assisting shippers without having trucks, or the shipper company that are at over capacity by adding supplemental drivers.

This company now having:

- +1500 kg loaded every day.
- +10.000 deliveries per month up to front door.
- +22.000 driver over the USA.

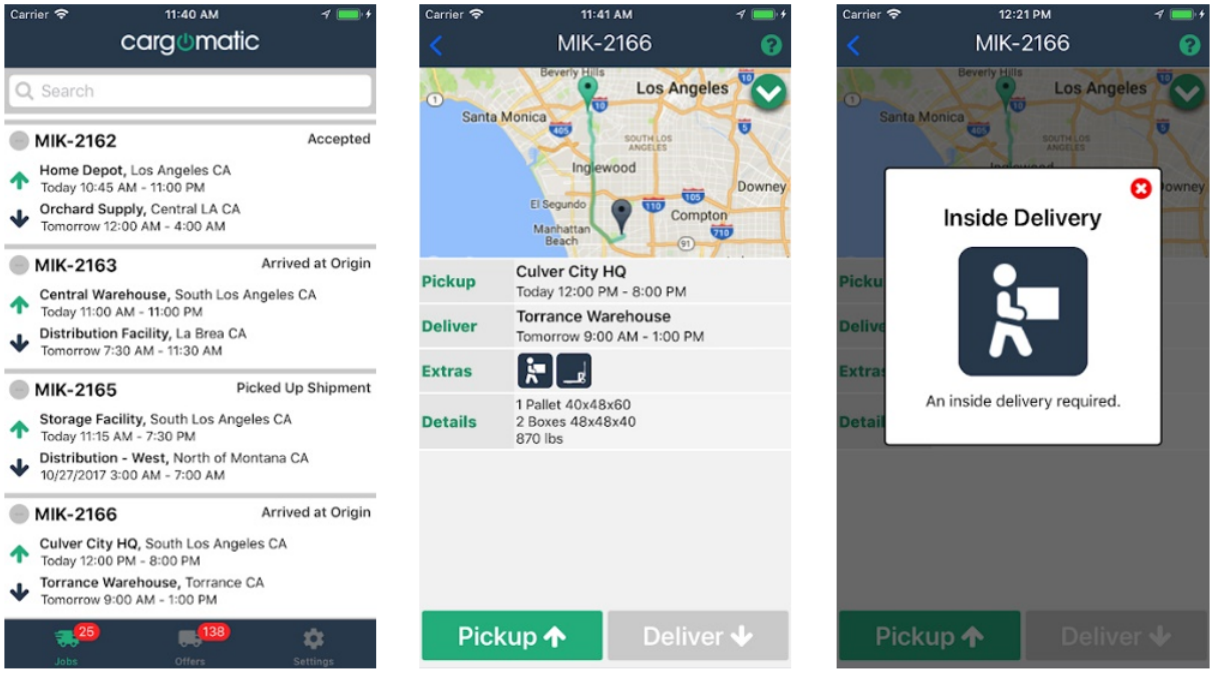

Figure 1.15: screenshot of similar solution

Uber

Uber is an android application focused on urban road transport. Uber is active in several countries. It offers several services to it's passengers and drivers. As it has separate apps that provide different interfaces for both sides: see Figure 1.17

- After registering, passengers can choose a meeting point and request a ride to their selected destination.They can also see the nearest drivers to their location.
- The application then gives the rider the estimated time of arrival for the driver.
- The application provides a communication interface between the driver and the rider.
- Payment is handled in-app, meaning no cash required.
- After ending the trip, the rider is prompted to rate the driver and vise versa, this two way rating system helps to give a fair assessment of both sides' experience.

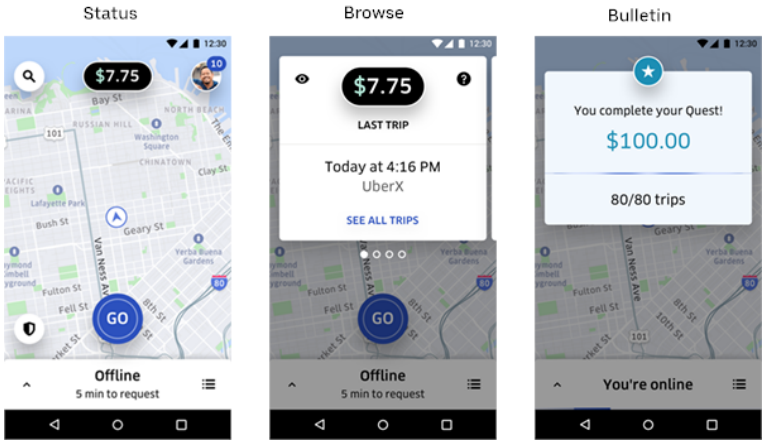

Figure 1.16: screenshot of Uber Driver Functions

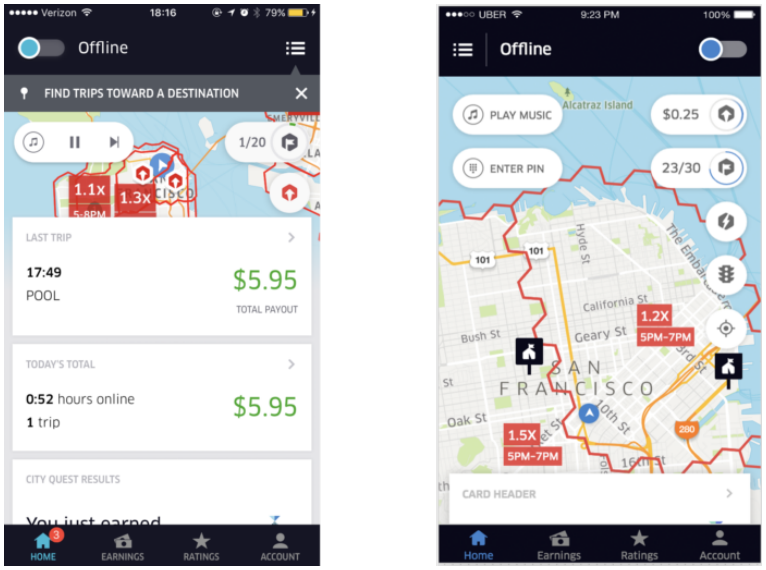

Figure 1.17: screenshot of Uber Driver Functions

Uber Eats

Uber Eats is a part of Uber's app suite and is specialized for food delivery. Uber has separate applications for drivers and customers. it functions like this: see Figure 1.18

- After logging in, customers are greeted with a pre-selected list of restaurants.
- Customers can choose a restaurant from that list or search for the restaurant of their choice.
- After clicking on the restaurant ,the customers are then invited to choose from the restaurant's menu.Then they place their order.
- The driver nearest to the passenger then gets a notification for the order and can choose whether to take it or not.

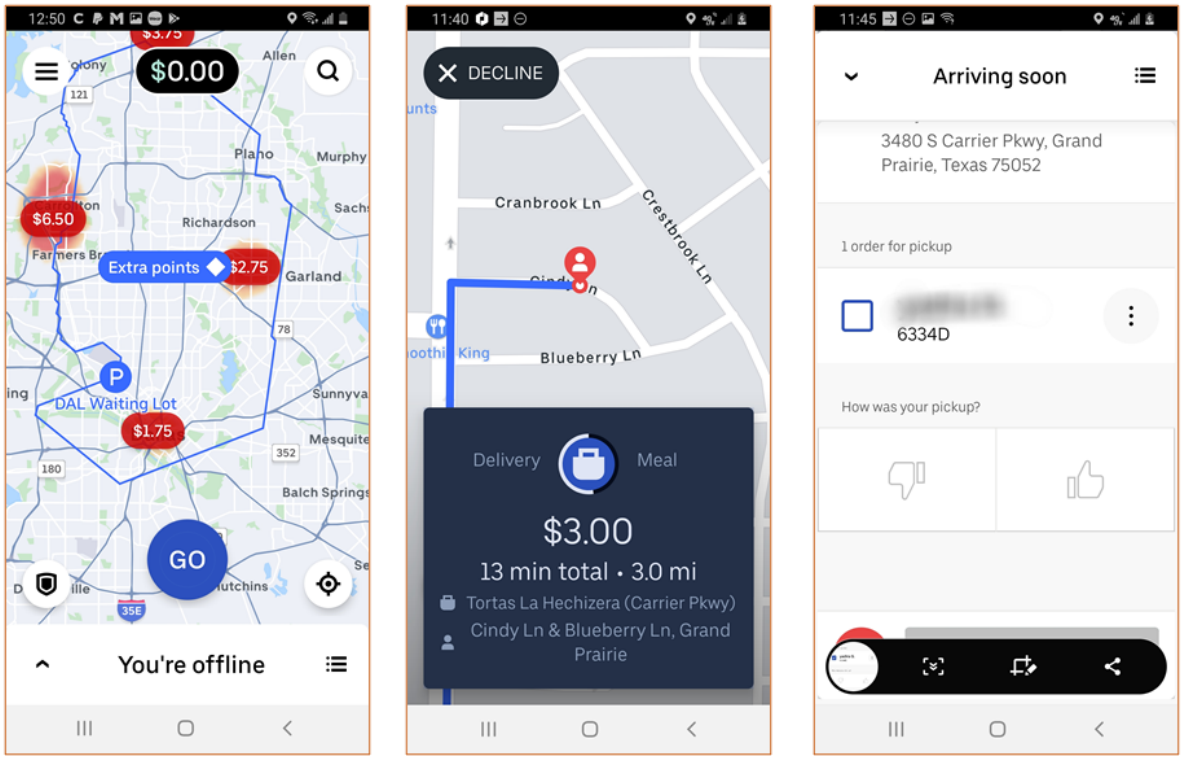

Figure 1.18: screenshot of Uber Eats Functions

Amazon Flex

Amazon Flex provides delivery services by using a fleet of gig worker drivers who use their own vehicles. Amazon Flex driver deliveries include: see Figure 1.19

- Packages: Picked up at a delivery station and delivered to customers. Your work must be completed within a time slot, which is usually in 3- to 6-hour blocks.
- Prime Now and Amazon Fresh: Groceries and household items are picked up from an Amazon delivery station, usually in 2- to 4-hour blocks.
- Store orders: Pickups from local stores, in blocks of 2 to 4 hours.
- Instant offers: Only available in limited areas, these deliveries start near your current location and last from 15 to 45 minutes.

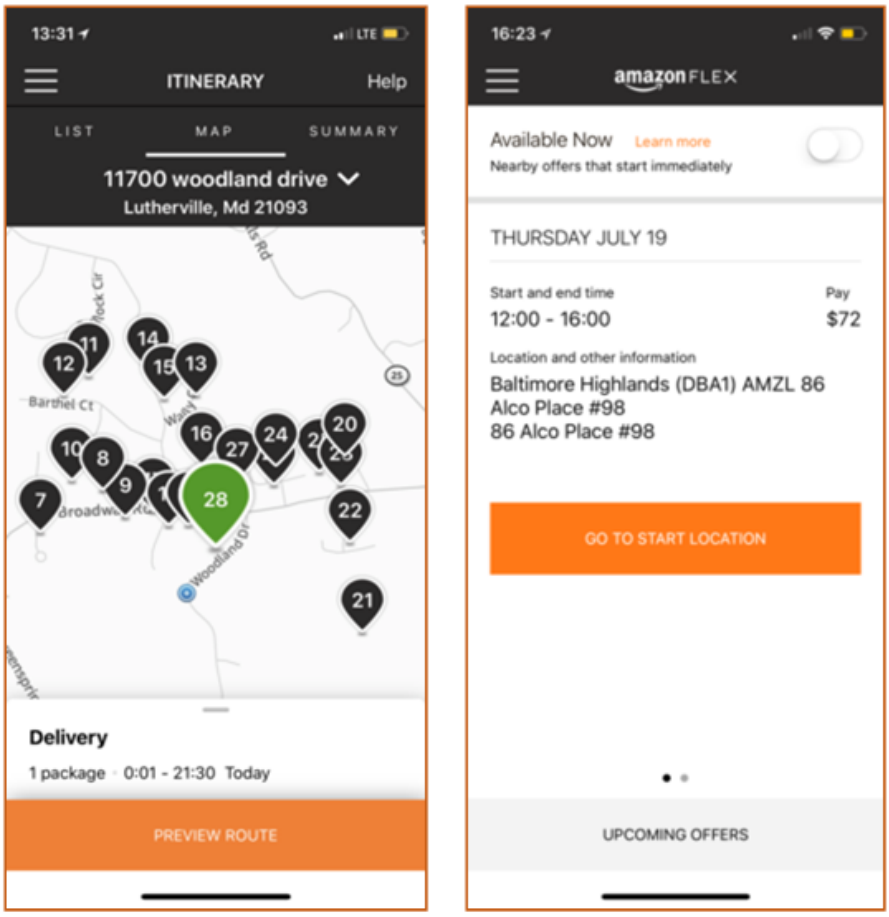

Figure 1.19: screenshot of Amazon Flex Functions

1.7 Efficiency of the project

The crowd delivery became the most efficient service in Europe and America, started with UBER, UBER Eats, Lift, Dropoff and YASSIR in Algeria.
From those company they came up with the idea of minimizing the rescue and charge of the cars, like maintains, oil change, insurance of the car and so many costs, easing the ability of companies to cope better with fluctuating demand, to leverage resources for different uses (e.g., parcel delivery and taxi services), and to get access to cheaper, less regulated labor, since almost anyone can drive a cab or deliver parcels. Many start-ups complain about quality and reliability issues.[18]
So, from here it will be easy to manipulate the project with the minimum charges, and focus only on the service itself, and the security of parcels itself.
In our case we need to focus on the security and guaranty of the parcel taken by the driver, by gathering all the legal papers from the driver before get involved in the company to work, and tracking his movement until he delivers the package safely. see Figure 1.20

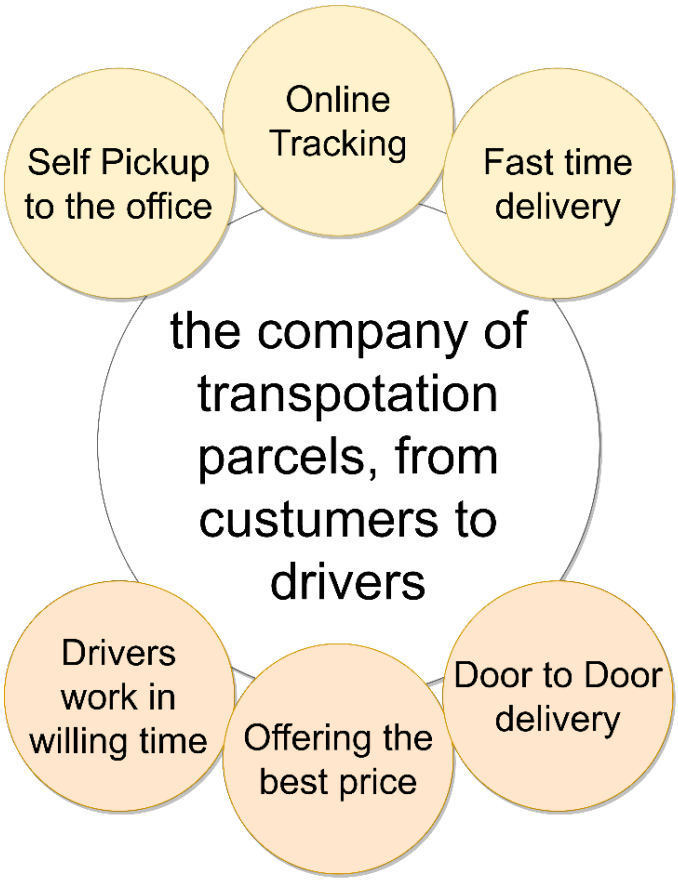

Figure 1.20: efficiency of this project

1.8 Pricing factors

In this project, we will focus to give the client the option to choose the best pricing, first by defining the estimated price based on the Weight, and the Dimension of the parcel, and here is the factor that will affect the price:

- Price based on best timing to deliver.
- Price based on the lowest price.
- Price based on guaranty and safety (if it's frangible).

Here is an example we can use in the project: For the guaranty of the parcel, we will increase the price with a value that will define the new price, multiplied by the distance and duration, and here the dimension will affect the driver capability, especially if only one parcel will take the whole space, so we need to increase the price based on that:
getEstimatedPrice()
estimatedPrice=weight*X + Height*Y + Width*Z + Large*W;
//estimatedPrice=3590(g)*0.3+29(cm)*0.2+40(cm)*0.2+30(cm)*0.1;

EMS use the factor of distance and weight.
Yalidine use the factor of distance and weight and dimension.

1.8.1 Is Frangible

This option could increase the price of delivery, especially when it comes to handle highly sensitive objects, like glass, and handmade stuff.
If (Frangible option demanded)
price = estimatedPrice + distance*Value1+ duration*Value2;
//price = 300(Da) + 406(km)*0.33;

This factor used only by few companies, like EMS.

1.8.2 Is Time Important

This option is also high recommended for those who are really in hurry to send their parcels in most responsive time, so that become an advantage for our project, that will make the driver decide if they are going to pick the parcel in the mentioned time or not.
If (Time Base option demanded)
price = estimatedPrice + distance*Value3 + duration*Value4;
//price = 300(Da) + 406(km)*0.55 + 305(min)*0.1;

1.8.3 Auction system

A reverse auction is a type of auction in which drivers bid for the prices at which they are willing to transport the parcels for a client.

- Regular Auction: In a regular auction, a driver announces his service with the advantages, and client place bids until the close of the auction, at which time the item goes to the highest bidder.
- Reverse Auction: In a reverse auction, the client puts up a request for a required delivery service. The driver then places bids for the amount they are willing to be paid for the service, and at the end of the auction of the driver with the lowest amount wins.[7]
- Fixed price: We can suggest starting with predefined prices, that drivers will post their own offers, for example: a driver will go from Laghouat to Algiers. He will post to deliver a parcel with 500DA.

1.8.4 The profit of the project

Using the precious factors, we are now able to control the market pricing, and giving the client best offers to let him choose the best option that fit his willing to send a parcel.
Since the project is offering a Crowd Shipment. The price is going to be the minimum. Each driver will get a percentage from the total price of shipment (Example YASSIR drivers takes 75%).

1.9 Tracking parcel operation

To make sure the client will be committed to this service, by offering tracking system contains: see Figure 1.21

- Unique code will be created for each parcel.
- The client confirms handing the parcel by scanning the code.
- From driver part, he will confirm the first pickup of parcel.
- When driver arrive to any end point office, the office will scan each parcel, to confirm the arrival.
- When the driver delivers the parcel to the end-client the driver will scan the code, and confirm the success of delivering.
- When the end-client receive the parcel successfully he will confirm the success of operation.

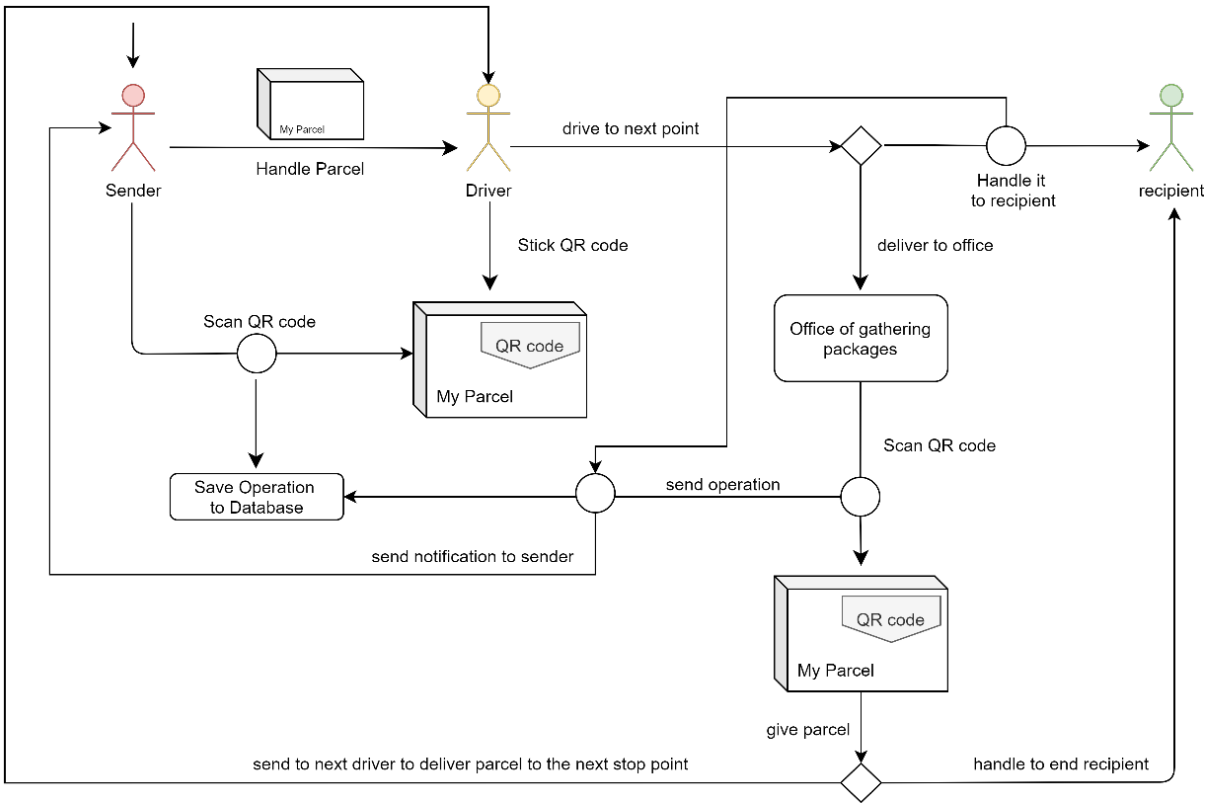

Figure 1.21: the operation of tracking parcel from sender to recipient

1.10 Limitation and outlines

In this project, we will face many problems, starting from the basic idea which is the acceptance of making delivery with unknown drivers, and an unknown client having only few legal information about them, which will cause many problems:

- Security: especially when the driver delivers a parcel from an unknown client and doesn't know what the parcel contains inside, and same thing for the client when he is not sure if the driver deserves to take the parcel especially with priceless stuff.
- Warranty: concern about delivering the parcel in the good condition and healthy state.
- Late deliveries: this can be caused by the driver himself and his own condition of making the late to pick the parcel or deliver it.
- Client canceling the requests: which is major problem, but currently it can't be controlled unless if there is pre-payment.
- Driver ignoring client request: some request of the client could be stayed ignored forever, and no driver will accept it.
- Unfair opportunities: some drivers will be always active in our system, chasing the best request first all the time, letting the lowest price or the faraway distance for who came late, (same problem happening now to Amazon Stores, where the drivers hanging their phones in the trees next to these stores to have the best opportunity's).[1]

1.11 Conclusion

From what we discussed about before, we noticed that the service of Crowd Shipment doesn't exist yet as a real handled service in Algeria. Yet it can offer the opportunity to give the best solution for the customer. Based on advantages of time, price, and guaranty, also for anyone who had a car or van will be integrated automatically in the system.

Chapter 2

Design & Analysis

2.1 Introduction

In this chapter we look at the blueprint of our work in the form of conceptual diagrams. We used UML for our project, because it provides a set of diagrams that are the easiest to implement and translate to actual code in our case, such as the complete tools that we are going to use to improve the best efficiency of the delivery optimization, and see the effect of these tools, to implement the best result.

2.2 UML presentation

UML or Unified Modeling Language is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. It was created by the Object Management Group (OMG) in 1997 and now it is one of their standards. UML is used to model the structure as well as the behavior of systems, such as software or others. UML has a multitude of diagrams to represent various aspects of a system constitution and behavior.

2.2.1 Class Diagram

Here is our class diagram: see Figure 2.1

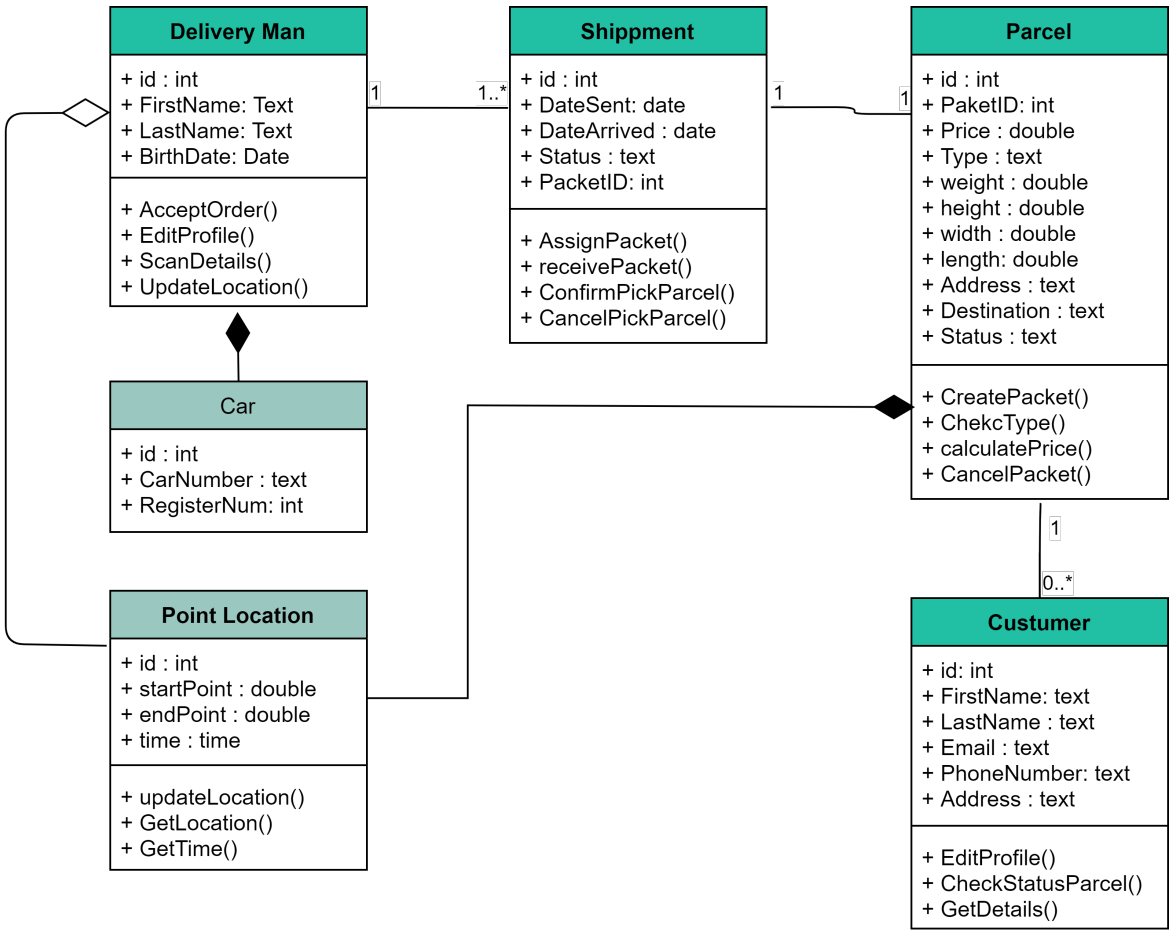

Figure 2.1: Representation of Class Diagram

2.2.2 Use case Diagram

see Figure 2.2 This here is our use case diagram:

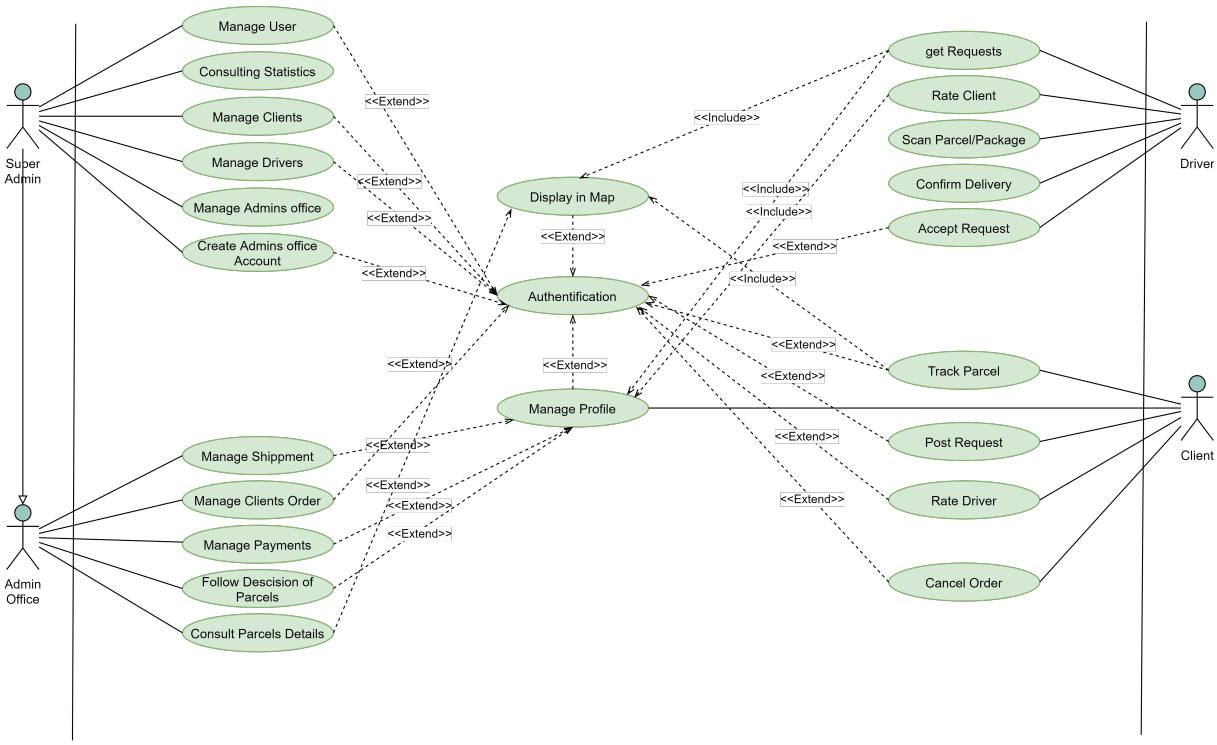

Figure 2.2: Representation of Use Case Diagram

2.2.3 Sequence Diagrams

This section will contain those subsections:

1. User Sequence Diagrams that represent the functionalities available to all users of the application.
2. Driver Sequence Diagrams showing in detail what the Drivers can do when using the transport service of our application
3. Administration service for the client and driver services.

2.2.4 User Sequence Diagrams

Register User

this sequence diagram shows the process of registration of a new user: see Figure 2.3

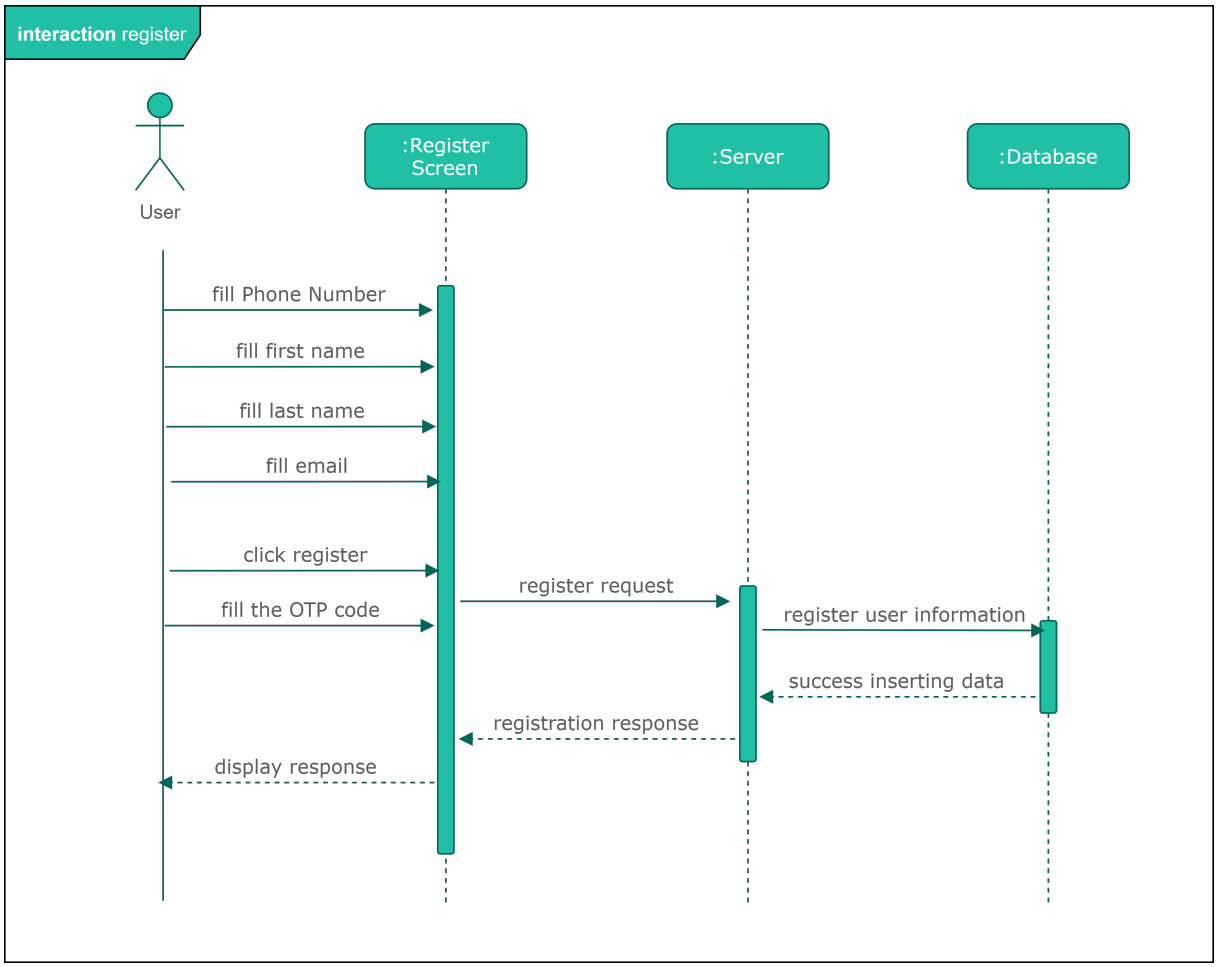

Figure 2.3: User Registration Sequence Diagram

Login

this sequence diagram describes the process of signing in to the platform with an existing account: see Figure 2.4

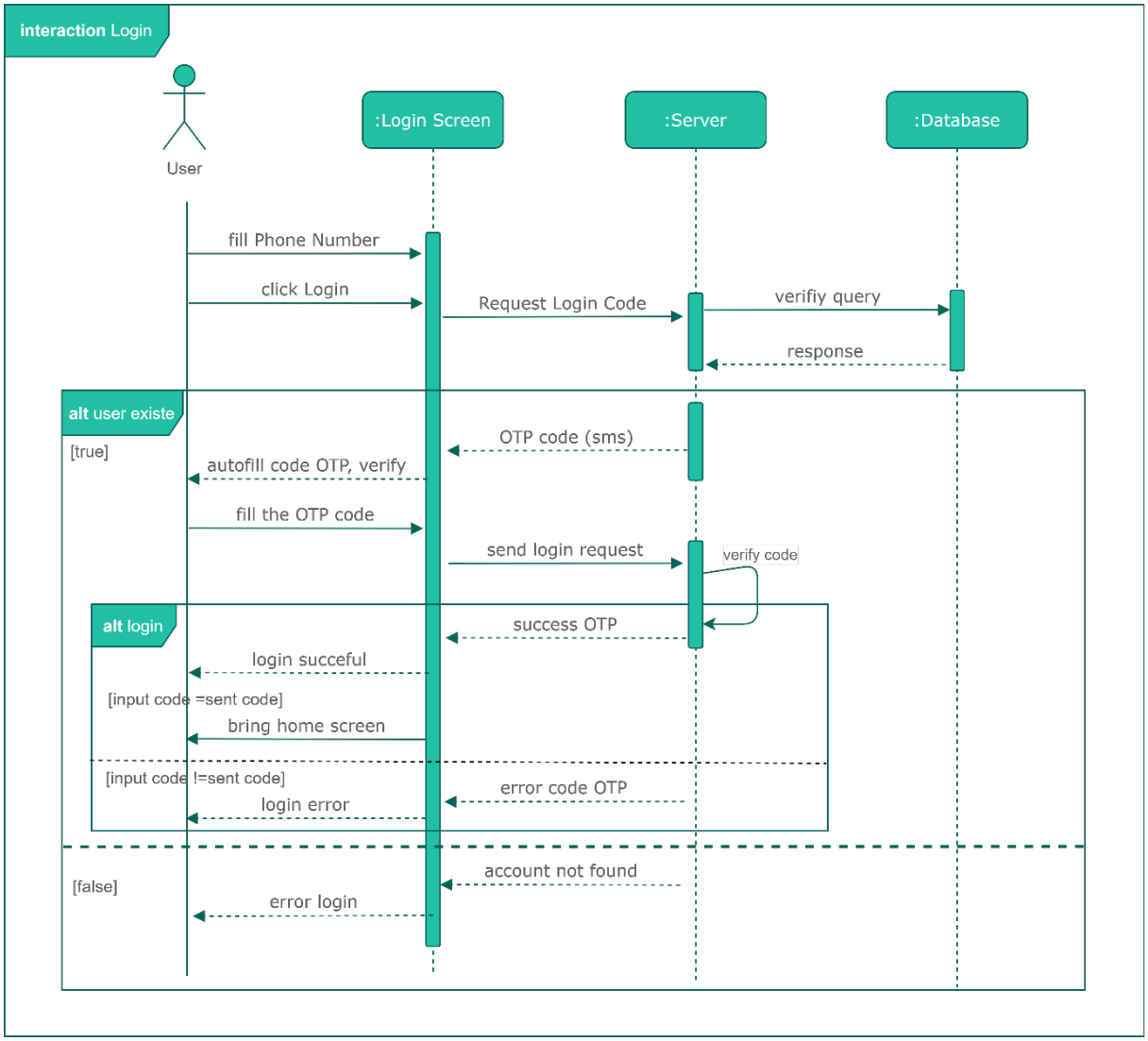

Figure 2.4: User Login Sequence Diagram

Edit profile

this sequence diagram describes the process of editing user information (profile): see Figure 2.5

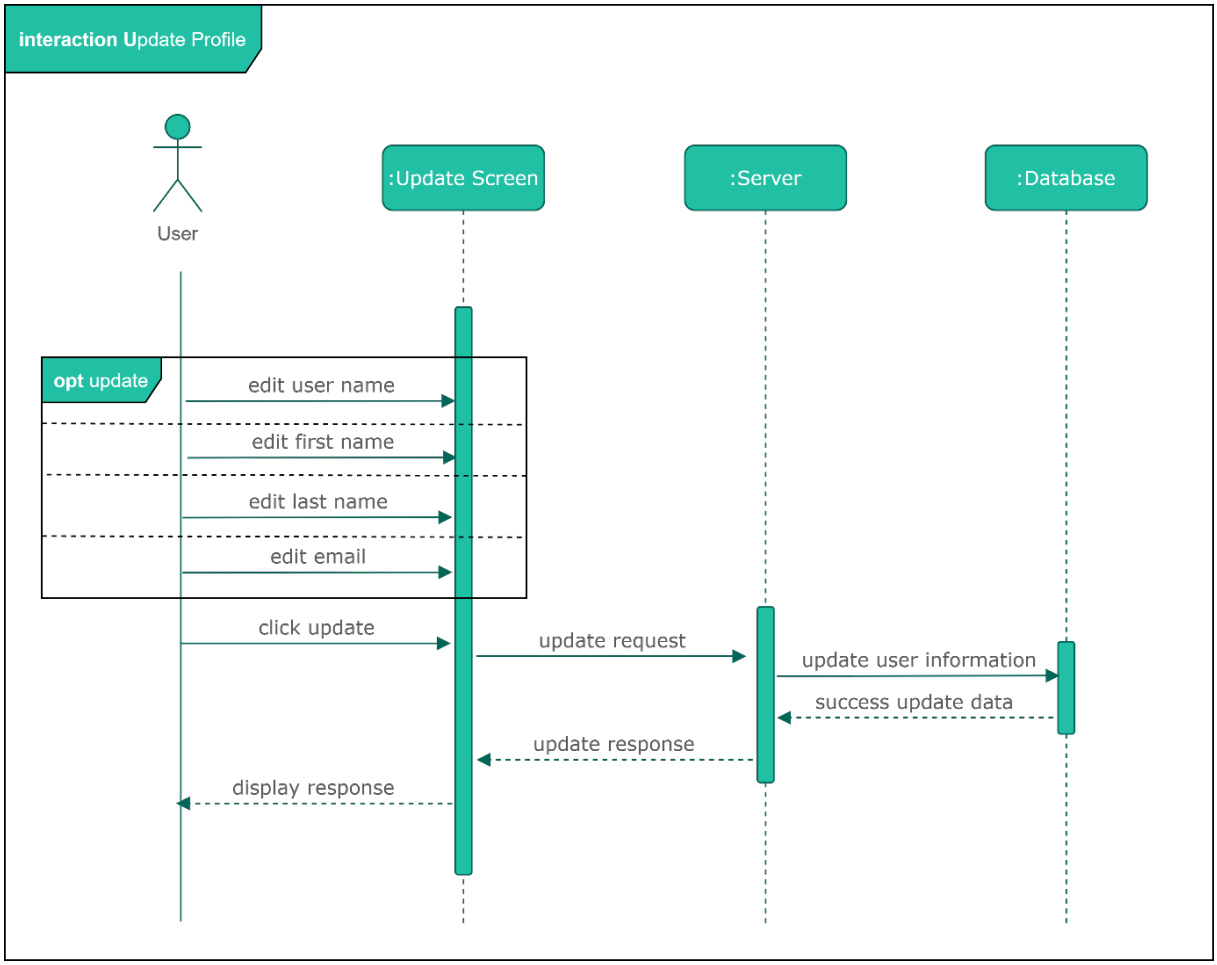

Figure 2.5: Edit Profile User Sequence Diagram

Post a Request

this sequence diagram describes the process for user of posting request for delivering a parcel from point A to point B: see Figure 2.6

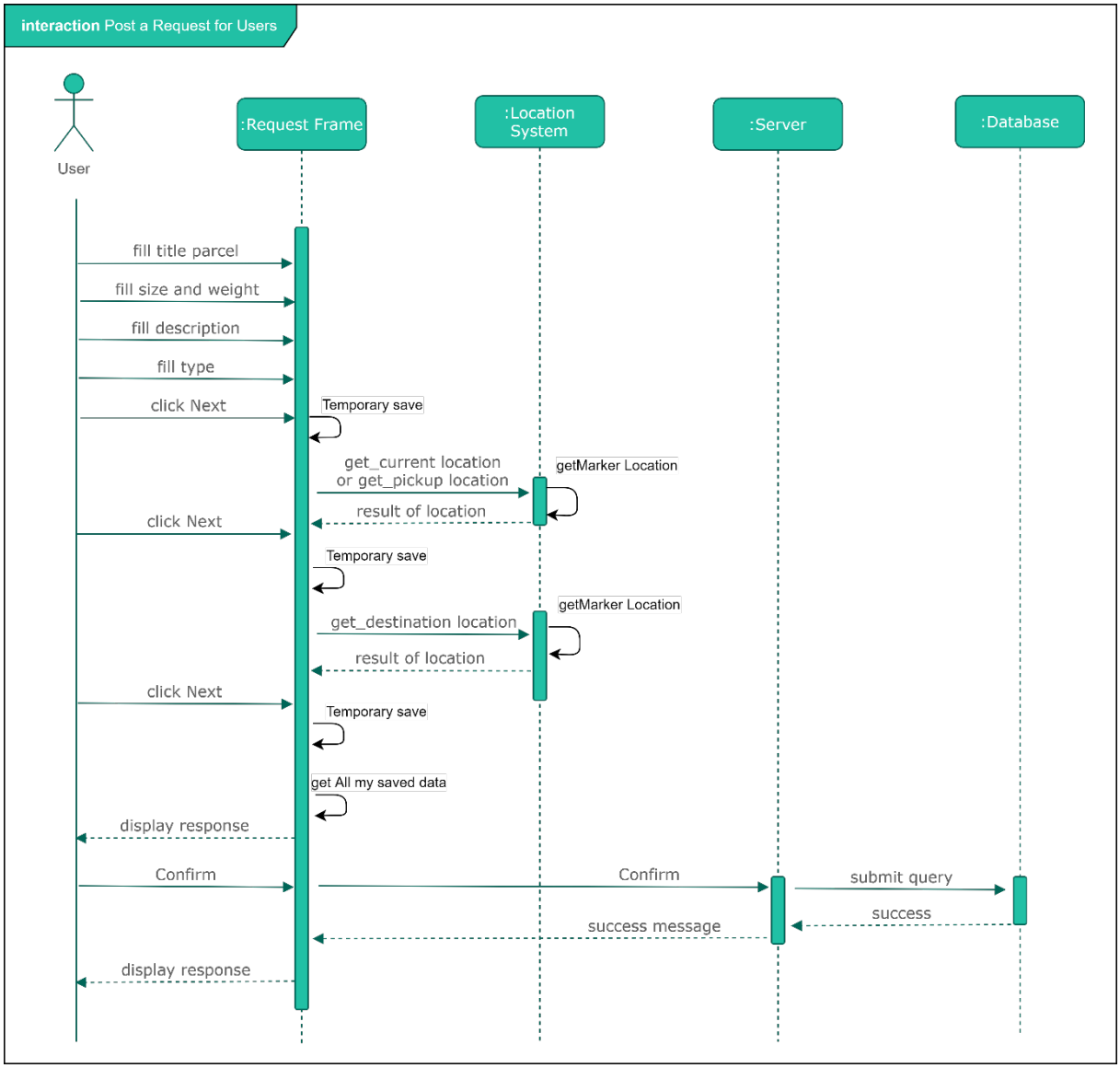

Figure 2.6: Post Request User Sequence Diagram

Update a Request

this sequence diagram describes the process for user of updating request for delivering a parcel from point A to point B: see Figure 2.7

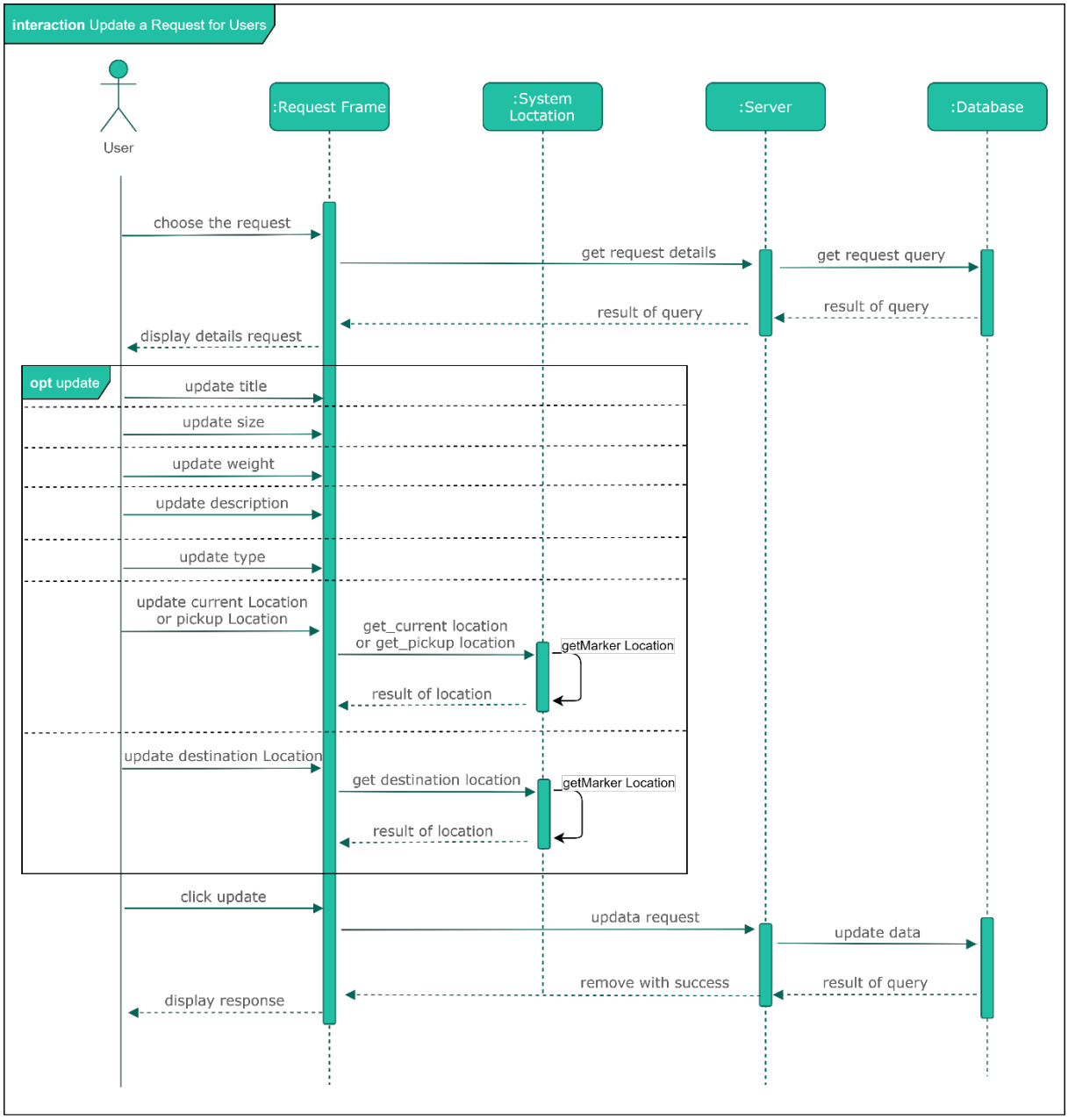

Figure 2.7: Update Request User Sequence Diagram

Cancel a Request

this sequence diagram describes the process for user to cancel their requests before confirmation: see Figure 2.8

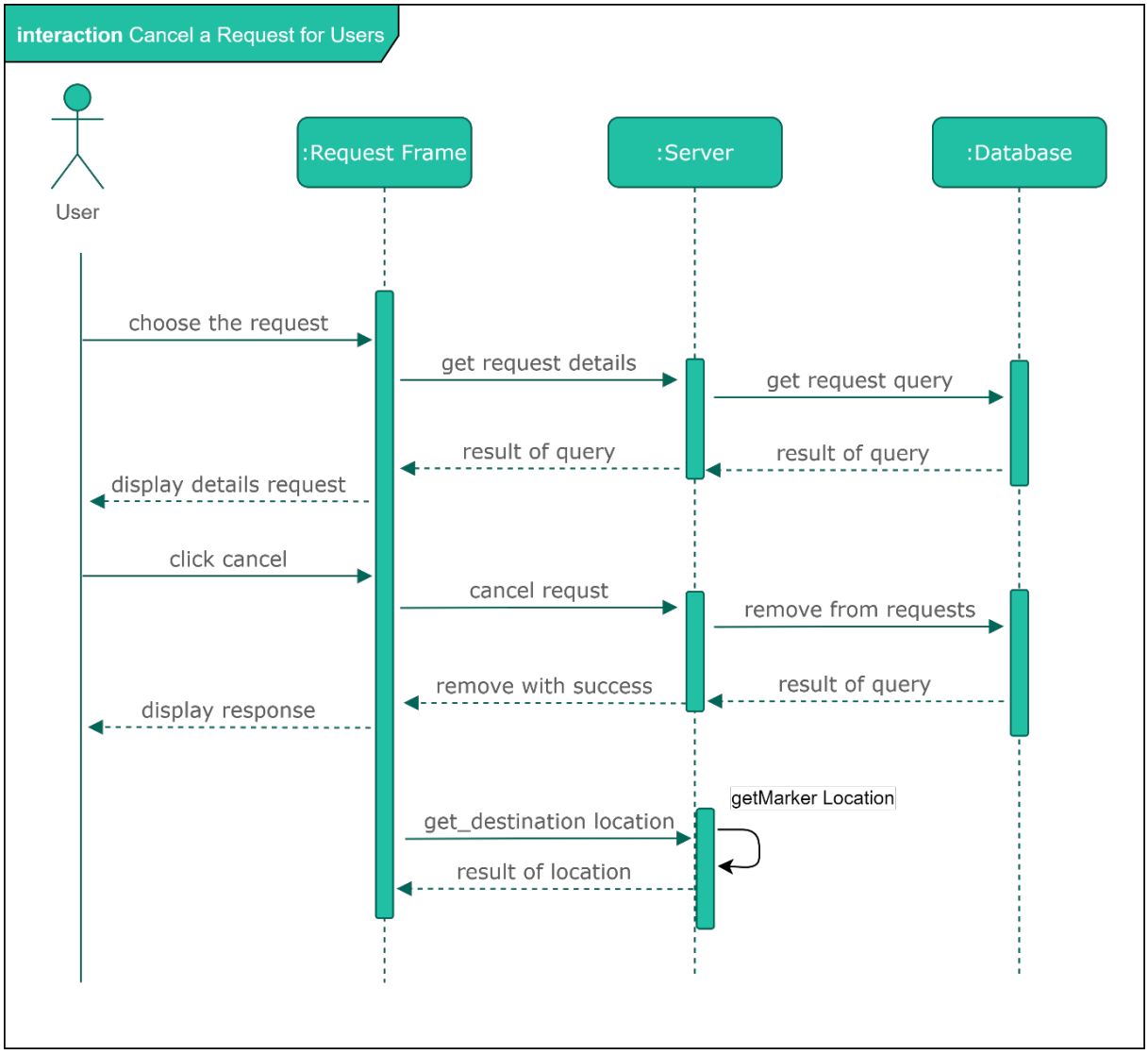

Figure 2.8: Cancel Request User Sequence Diagram

Rate the Driver

this sequence diagram describes the process for user rating the driver that deliver successfully/failed: see Figure 2.9

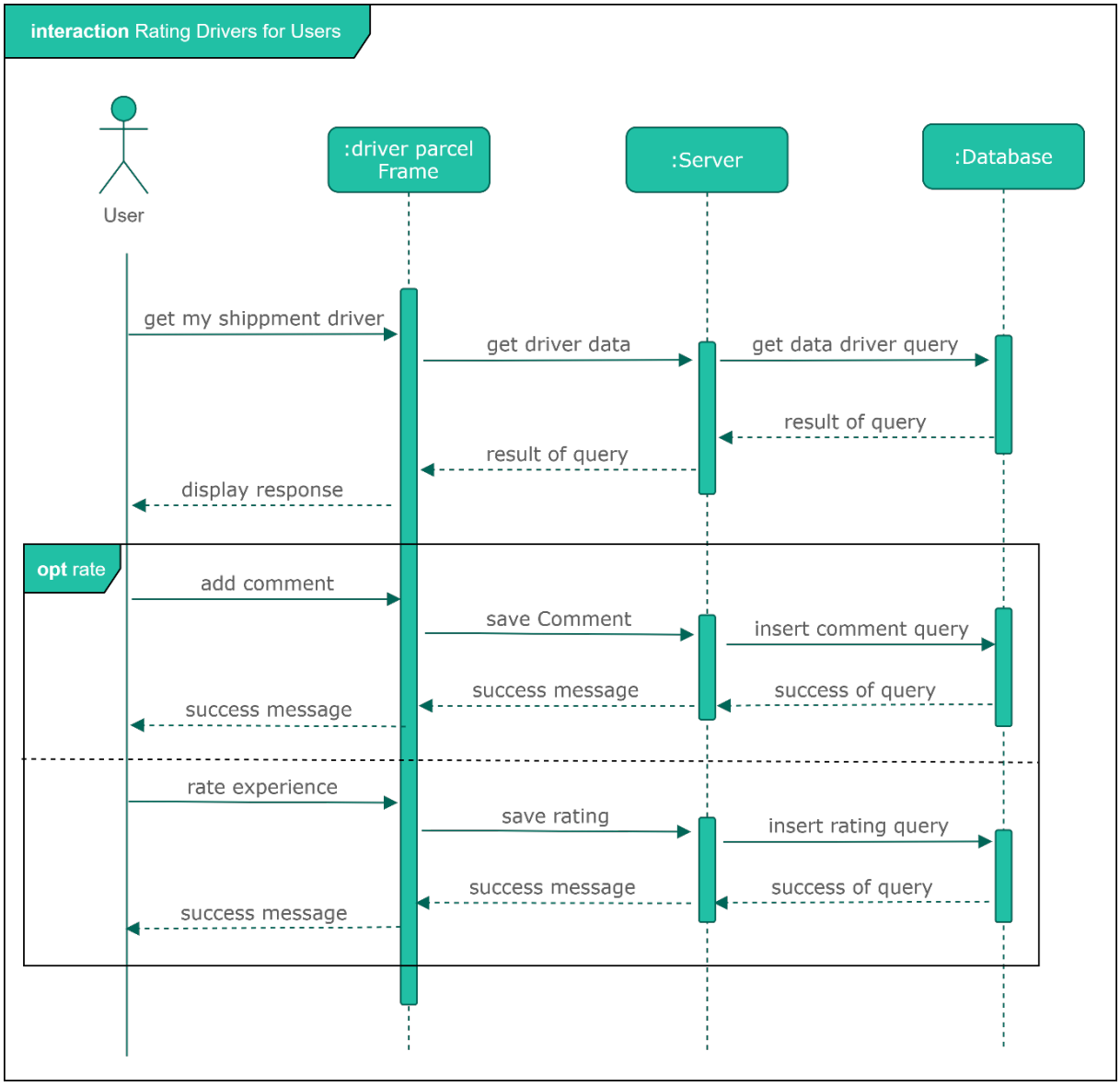

Figure 2.9: User Posting a Rate for Driver Sequence Diagram

2.2.5 Driver Sequence Diagrams

Register Driver

this sequence diagram shows the process of registration of a new Driver: see Figure 2.10

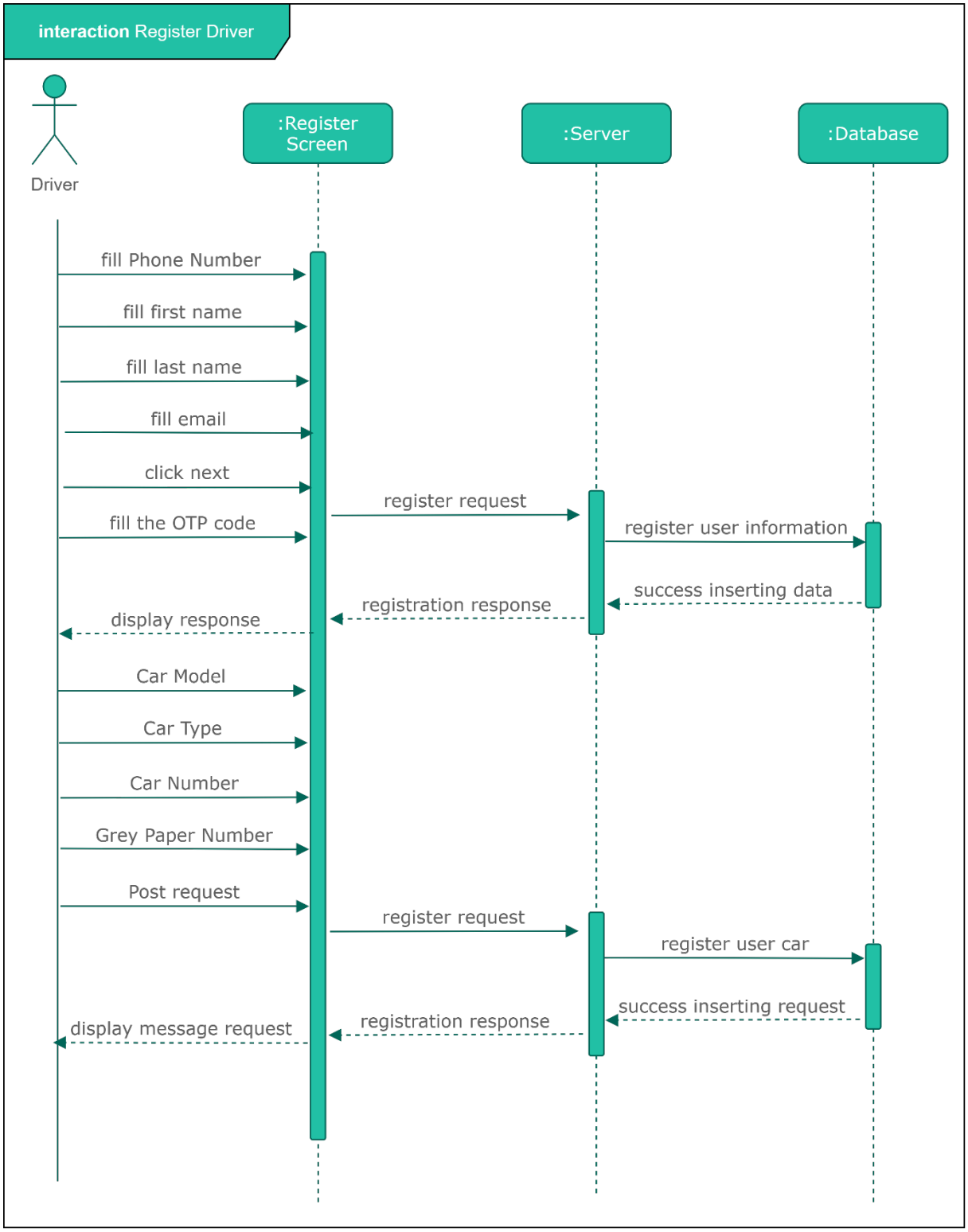

Figure 2.10: Driver Registration Sequence Diagram

Login

this sequence diagram describes the process of signing in to the platform with an existing account for drivers: see Figure 2.11

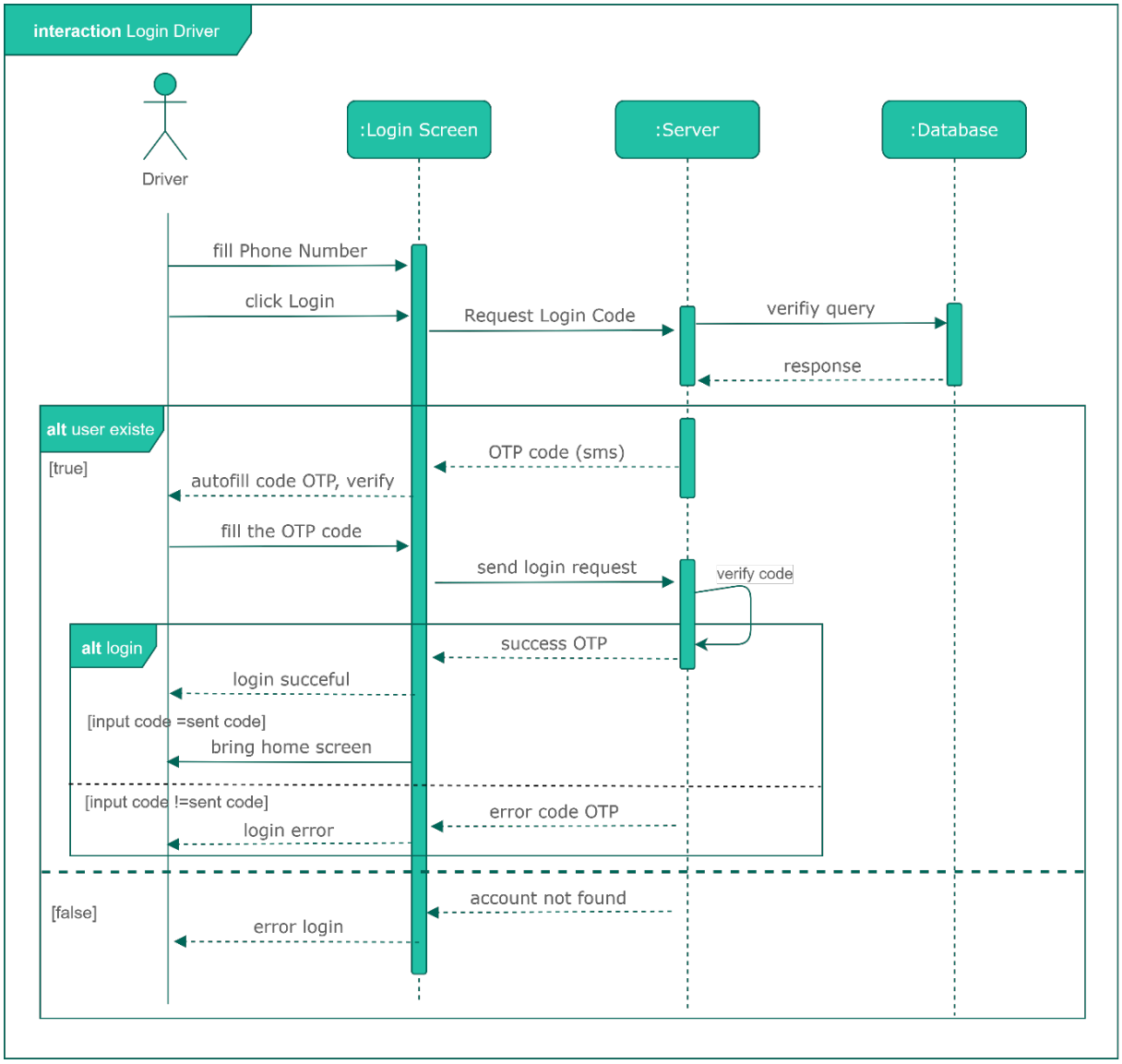

Figure 2.11: Driver Login Sequence Diagram

Update car driver

this sequence diagram describes the process of editing driver information remove n car inside the profile: see Figure 2.12

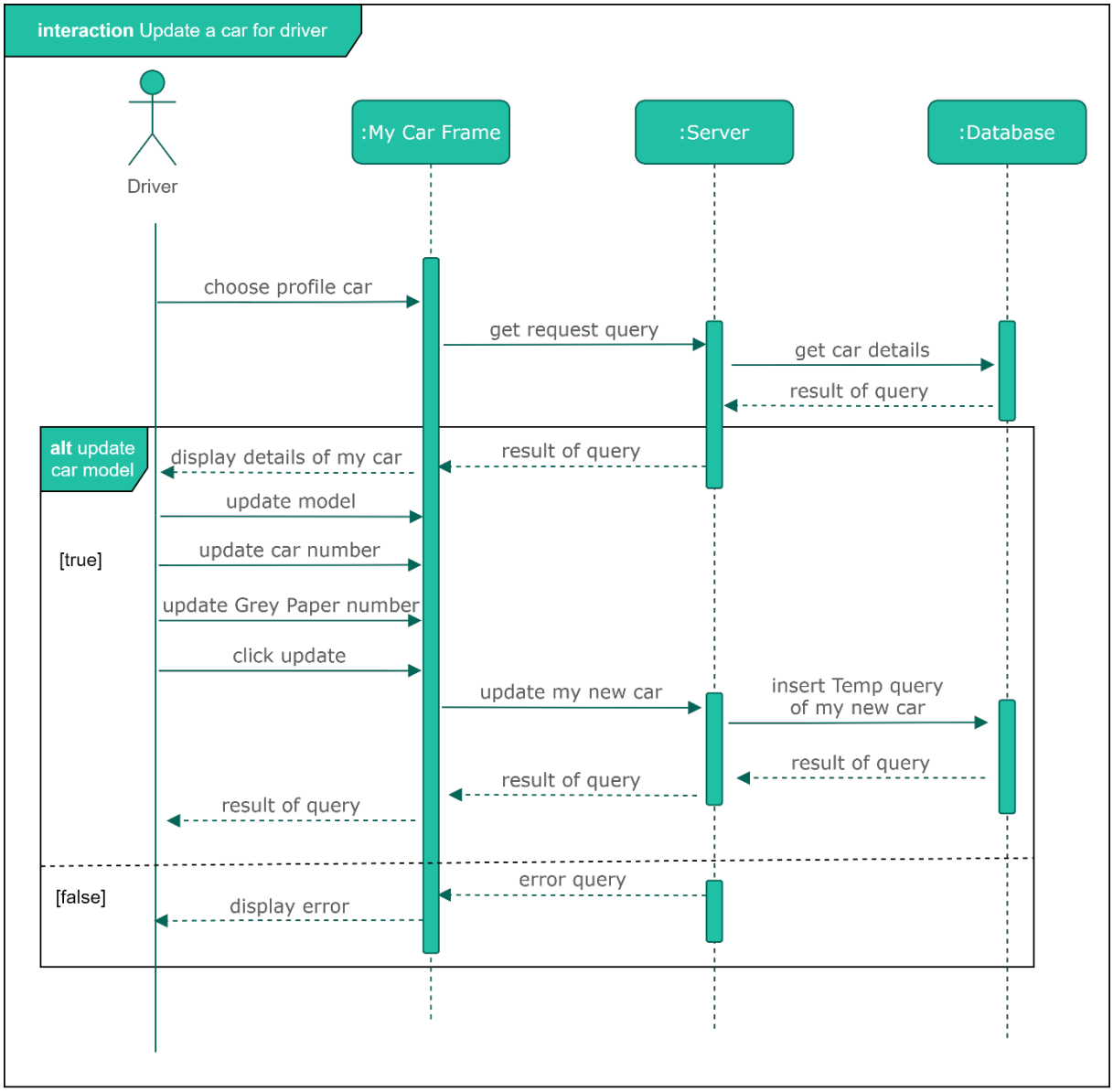

Figure 2.12: Driver Update Car Information Sequence Diagram

Update real time location for driver

this sequence diagram describes the process of sharing the real time location of drivers with the DB: see Figure 2.13

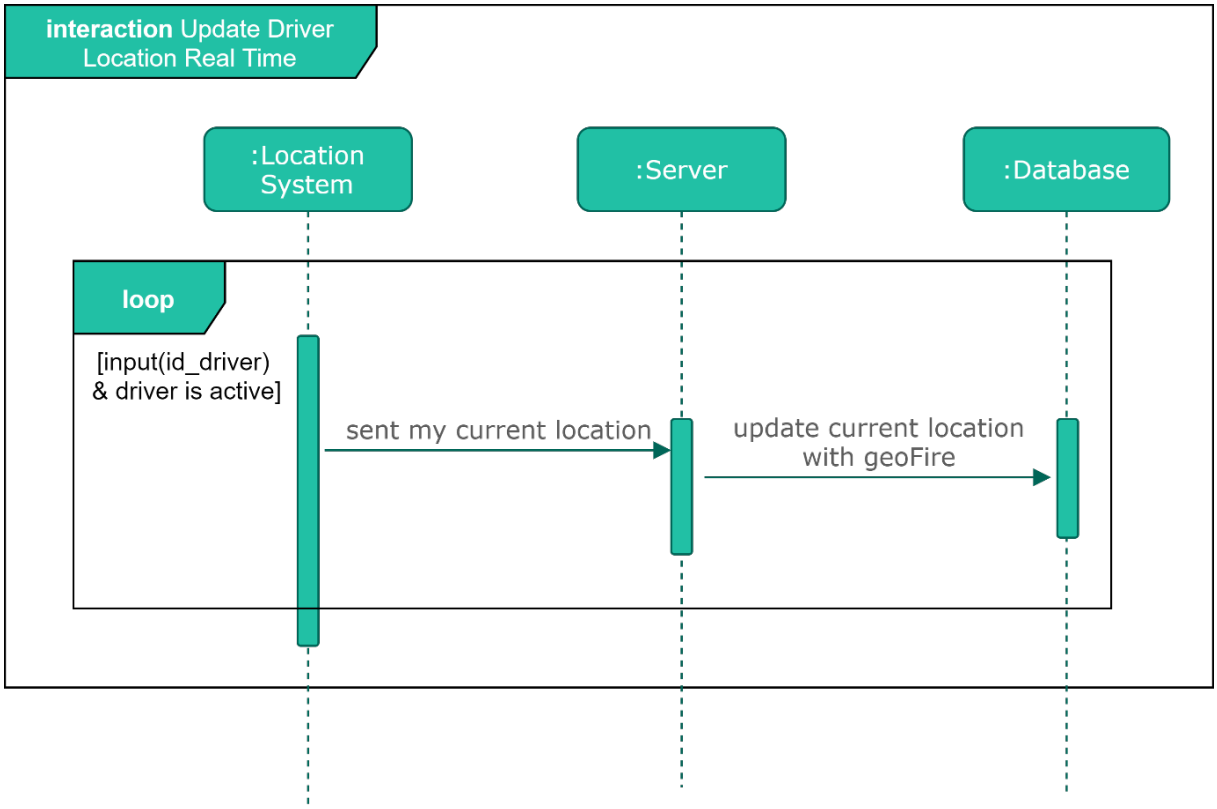

Figure 2.13: Driver Share RealTime Location Sequence Diagram

Show request of clients

this sequence diagram describes the process of sharing the non-accepted request of client in order to consult it and accept the offer: see Figure 2.14

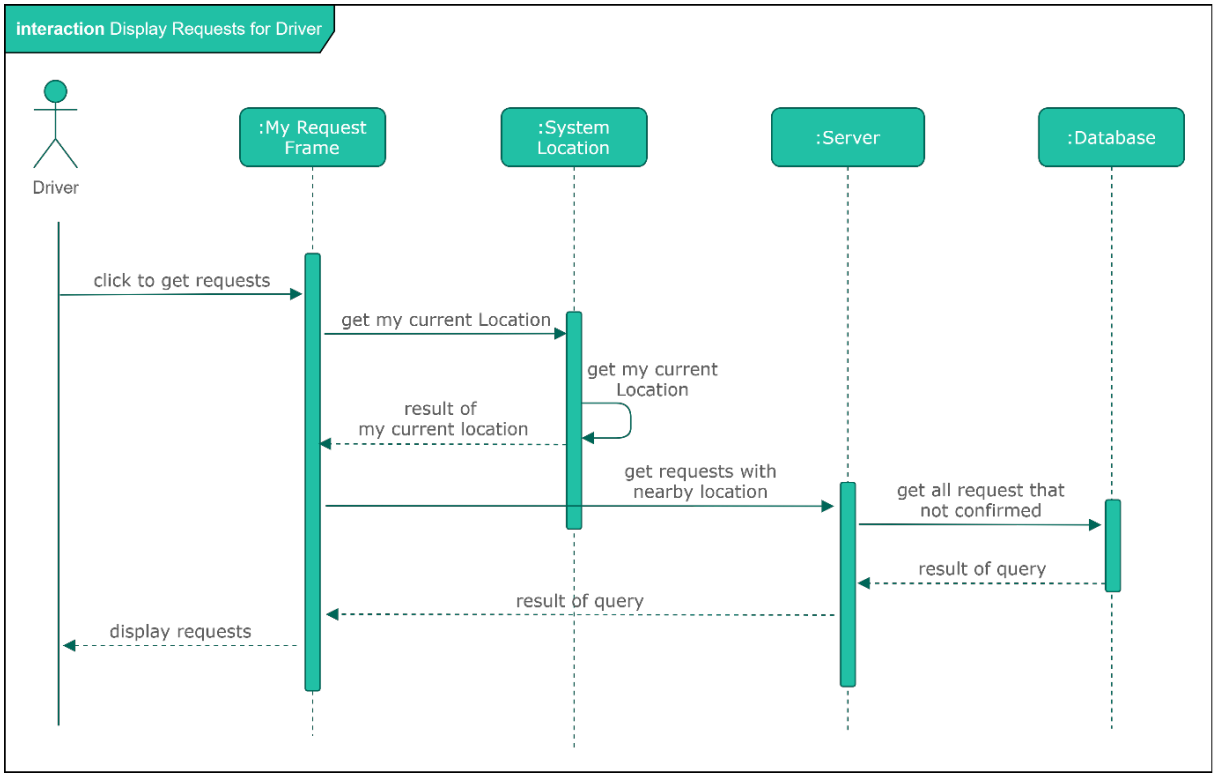

Figure 2.14: Driver Listing the Requests Sequence Diagram

Accept the request of client

this sequence diagram describes the process for driver to accept the request of client and deliver it: see Figure 2.15

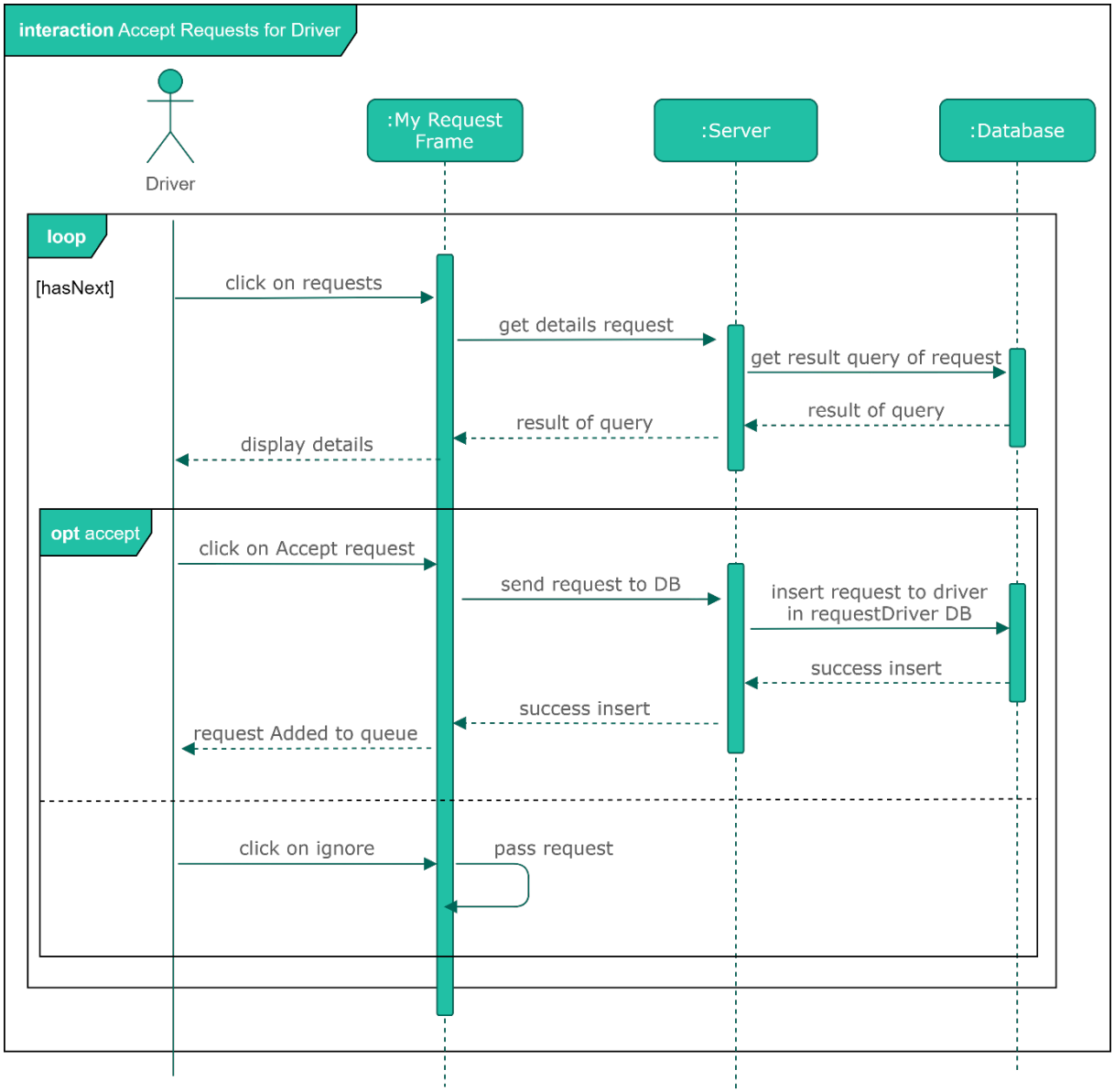

Figure 2.15: Driver Accept Requests Sequence Diagram

Find best path to start

this sequence diagram describes the process for driver to start following the shortest and fasted path to deliver parcels to the clients: see Figure 2.16

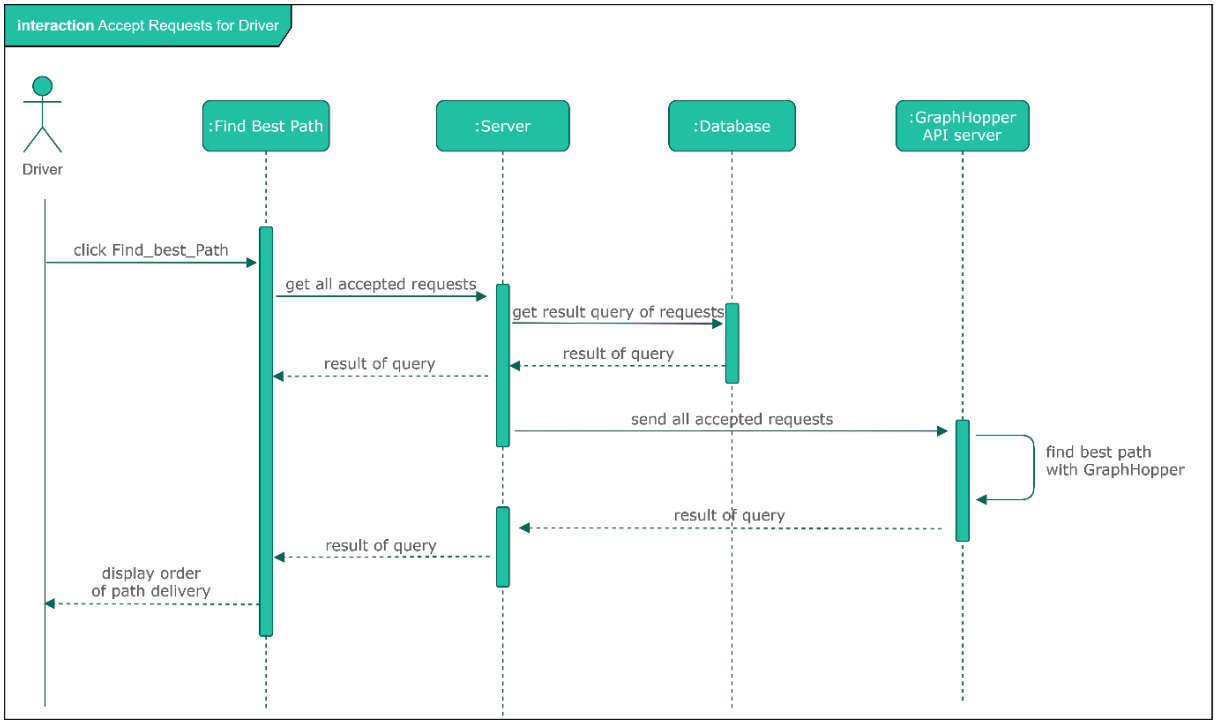

Figure 2.16: Driver Following Best Path Sequence Diagram

2.2.6 Administration Diagram

This diagram consists of the part of administration office, where the startup will get the benefits of the service, by calling each driver to visit the center and pay his part of work. see Figure 2.17

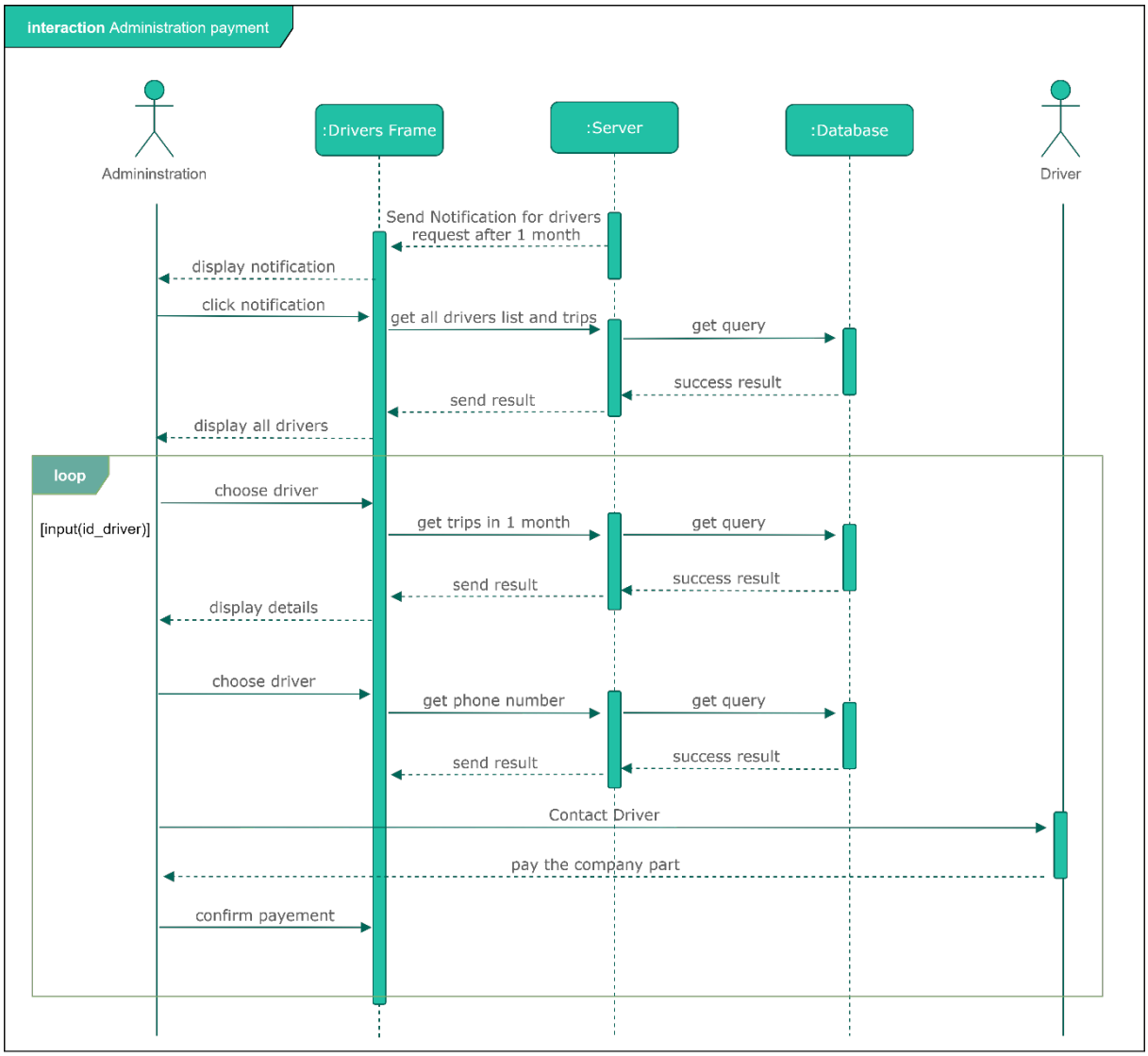

Figure 2.17: Administration get payment of driver

2.3 Code Architecture

2.3.1 Clean architecture

It's the organization of the code into classes or files or components or modules. And it's how all these groups of code relate to each other. The architecture defines where the application performs its core functionality and how that functionality interacts with things like the database and the user interface. see Figure 2.18

Figure 2.18: clean architecture and the complex one

Clean architecture refers to organizing the project so that it's easy to understand and easy to change as the project grows. This doesn't happen by chance. It takes intentional planning.
Separation of code in different modules or sections with specific responsibilities making it easier for maintenance and further modification.
Loose coupling; flexible code anything can be easily be changed without changing the system, and Easily Testable.
Over the last several years we have seen a whole range of ideas regarding the architecture of many systems. These includes:

- Hexagonal Architecture.[2]
- Onion Architecture.[23]
- Screaming Architecture.[19]
- DCI.[6]
- BCE.[8]

Though these architectures all vary somewhat in their details, they are very similar. They all have the same objective, which is the separation of concerns. They all achieve this separation by dividing the software into layers. Each has at least one layer for business rules, and another for interfaces.
Each of these architectures produce systems that are:

1. Independent at the Frameworks: The architecture does not depend on the existence of some library of feature laden software. This allows you to use such frameworks as tools, rather than having to cram your system into their limited constraints.

2. Testable: The business rules can be tested without the UI, Database, Web Server, or any other external element.

3. Independent of UI: The UI can change easily, without changing the rest of the system. A Web UI could be replaced with a console UI, for example, without changing the business rules.

4. Independent of Database: You can swap out Oracle or SQL Server, for Mongo, BigTable, CouchDB, or something else. Your business rules are not bound to the database.

5. Independent of any external agency: In fact, the business rules simply don't know anything at all about the outside world.

The clean architecture came as a development of all this constraint, to manage and integrating the manageability of the project through many rules. see Figure 2.19

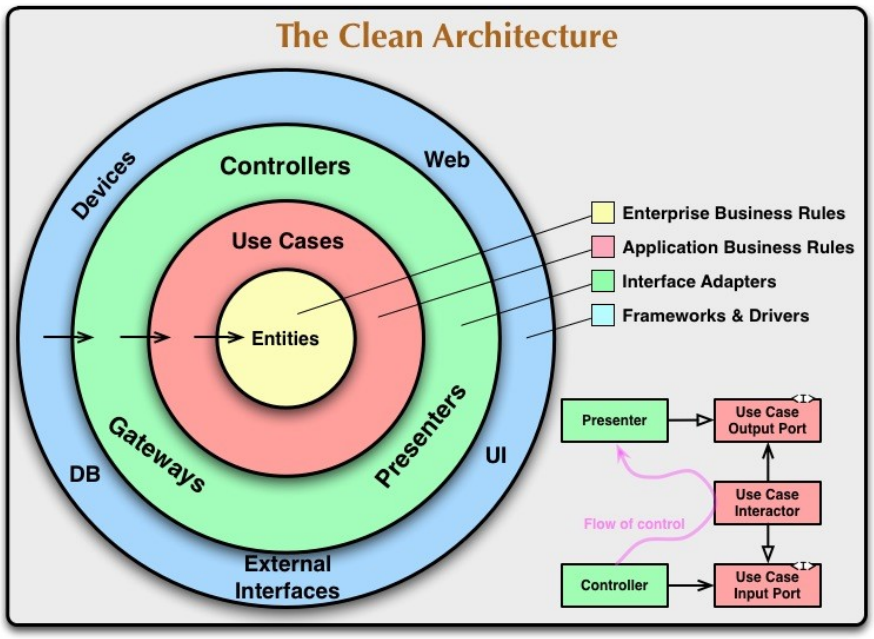

Figure 2.19: Detailed Guide on Android Clean Architecture

2.3.2 Layers of clean architecture

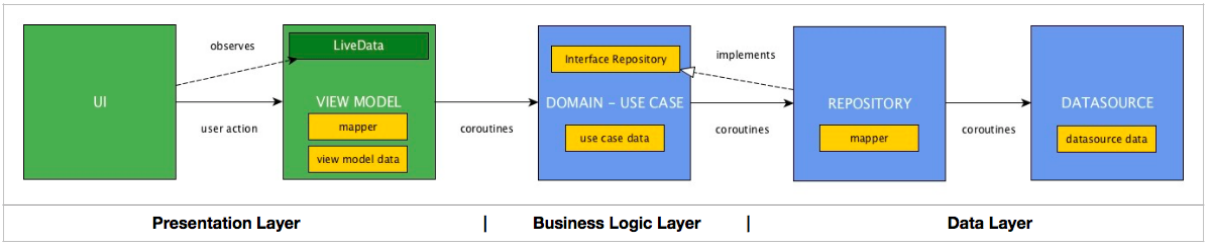

Figure 2.20: Layers of a clean architecture on Android with View-Model in the presentation layer

The presentation layer

The presentation layer is the user layer, the graphical interface that captures the user's events and shows the results. It also performs operations such as verifying that there are no formatting errors in the user's data entry and formatting data to be displayed in a certain way.
In our example, these operations are shared between the UI layer and the View-Model layer (we will explain this model next in this chapter):

- The UI layer contains the activities and fragments, capturing user events and displaying data.
- The View-Model layer formats the data so that the UI shows them in a certain way and verifies that the user's entries have the correct format.

Instead of using View-Model we can use another layer that does this function, the important thing is to keep the idea of the responsibilities of each layer.
In our example, the UI layer shows the results of the beers and the View-Model is the one that tells you the type of color you should use since it checks the alcoholic graduation range so that the UI knows which color to paint each one.

The business logic layer

In this layer all the rules that a business must comply with our business. For this, they receive the data provided by the user and perform the necessary operations. In our example, the ordering of beers from lowest to highest alcoholic strength are the business rules for what the Use-Case class will do. It is the most stable layer and the one that indicates what is happening in the software architecture developed.

The data layer

This layer is where the data can be accessed. These operations are divided between the Repository layer and Data-Source:
The Repository layer is the one that performs the logic of data access. Your responsibility is to obtain them and check where they are, deciding where to look at each moment.
For example, you can first check the database and, if they are not, search them on the web, save them in the local database and now return the saved data. That is, it defines the flow of access to the data. In our example, it asks beers directly to the data layer that communicates with the API.
The Data-Source layer is what the implementation performs in order to access the data. In our example, it is the one that implements the logic to be able to access the API data of beers.

2.3.3 MVVM Architecture

MVVM or Model View View Model is an architectural pattern for implementing user interfaces and it consists of 3 parts: see Figure 2.21 see Figure 2.22

1. Model: This holds the data of the application. It cannot directly talk to the View. Generally, it's recommended to expose the data to the View Model through Observable.
2. View: It represents the UI of the application devoid of any Application Logic. It observes the View Model.
3. View Model: It acts as a link between the Model and the View. It's responsible for transforming the data from the Model. It provides data streams to the View. It also uses hooks or callbacks to update the View. It'll ask for the data from the Model

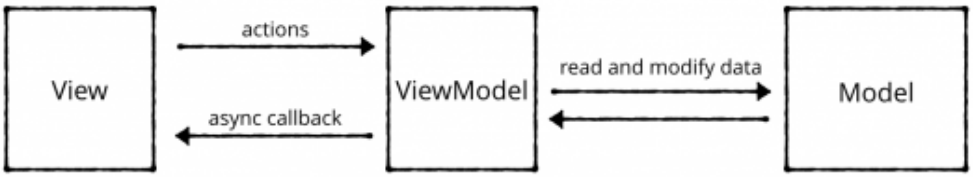

Figure 2.21: MVVM Presentation

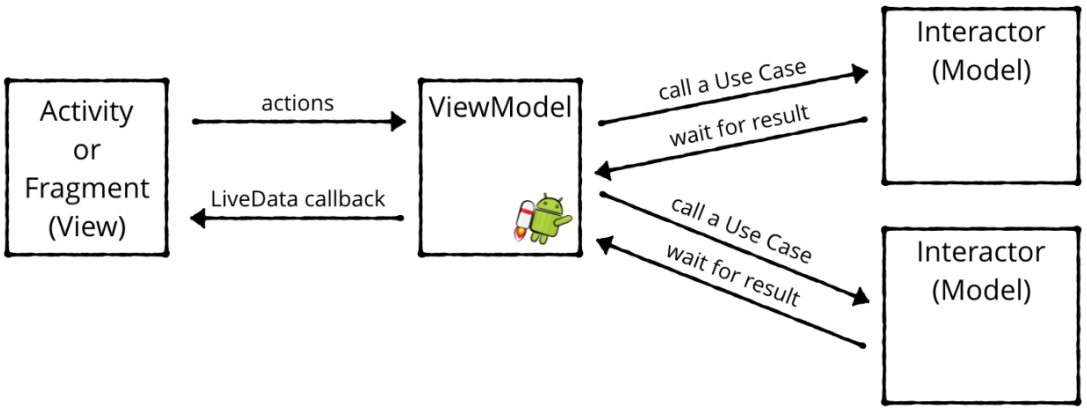

Figure 2.22: MVVM Simplified Presentation

2.3.4 Usage of MVVM inside the clean architecture

MVVM separates the view (Activity's and Fragments), which is enough for small projects, but when the codebase becomes huge, the View-Models start bloating (like spaghetti code), and separating responsibilities becomes hard. It clearly abstracts the logic of the actions that can be performed in the application. In Clean Architecture, instead of relying on Models, it will communicate with Interactors from the Use Case layer.

2.3.5 Disadvantages of Clean Architecture

It has a slightly steep learning curve. How all the layers work together may take some time to understand, especially if it coming from patterns like simple MVVM. It adds a lot of extra classes and subclasses, so it's not ideal for low-complexity projects. It also took long time to start the project in order to arrange the work in form of tree.

2.3.6 The reason of choosing MVVM

MVVM provides an advantage over the other architectures, as the code became more and more complex. It will avoid wasting time and minimize maintenance cost and error corrections. Also the main architecture that is supported and promoted by Google for application developments, add to that. This architecture is the most suitable for multi-service application. It allows for greater flexibility in adding new functionalities to the application, MVVM is also recommended to use for development in teams.

2.4 GraphHopper routing engine

GraphHopper is an open-source routing library and server written in Java and provides a web interface called GraphHopper Maps, As well as a routing API over HTTP. It runs on the server, desktop, Android, iOS.

GraphHopper can be configured to use different algorithms such as Dijkstra, and its bidirectional versions. To make routing fast enough for long paths and avoid heuristically approaches, GraphHopper uses contraction hierarchies by default. In the Java Magazine from Oracle, the author, Peter Karich, describes the techniques necessary to make the system memory efficient and fast.
Furthermore, GraphHopper is built on a large test suite including unit, integration and load tests. [9]
The GraphHopper Directions API consists many web services, that can implement and used in the code to achieve the best result of the optimization, and here are sets of the web services API [10]:

- Routing API
- Route Optimization API
- Isochrone API
- Map Matching API
- Geocoding API
- Cluster API.

We consist of using only few API's, to reach the goal of the crowd delivery optimization, for the pickup and delivery operation in our project.
Route Optimization API, can solve a variety of vehicle routing issues, including the classical traveling salesman problem (where we give a set of cities, and the problem is to find the shortest possible route that can the man visits every city exactly once and returns to the starting point). It can even handle vehicle routing problems complicated by business-specific side constraints; this API will handle:

- Timing: when the client and driver have an opening or specific delivery times, it can define multiple time windows. Additionally, it can specify the amount of time the driver needs to serve customers.
- Traffic: the optimization of routes based on historic travel time data aggregated from millions of anonymously-collected data points.
- Pricing: the price here is one of the most important factors for the customers, it will offer the best cost for each client, based on the fastest road, and the timing to get the parcel delivered, the GraphHopper will calculate all this factors together, and finally we can get the most sufficient result to work with it.

2.4.1 Route Optimization API

The Route Optimization API gets several locations and vehicles as input and calculates the best route for every vehicles, where several constraints like capacity or time windows can be added. By choosing the starting point and the destination point, and adding multi stop points, after that it will get the best result to follow the path, starting ascendant from 1 to 7 until the driver reach the destination.

(This example with 9 Willaya's, starting from Laghouat, Djelfa, Boussada, Msila, Bourdj Bou Ariridj, Bouira, Ain Ouassara, Tiaret, Blida, then Algiers) This tool can dynamically change it state of result each time it will find traffic jam. see Figure 2.23

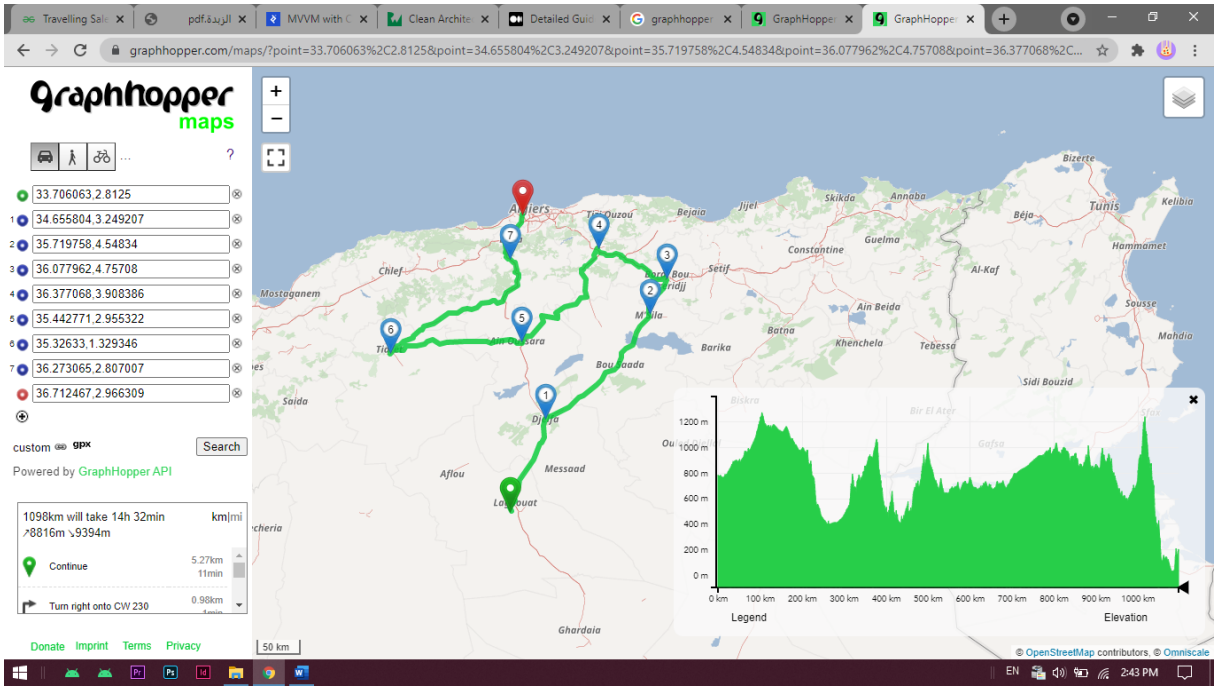

Figure 2.23: Simulation with GraphHopper of the Route Optimization API

We can use this tool as an API to show the best path for drivers, to pick their parcels they want to deliver, by choosing the parcel client and accept it, then get their location and the passing by them all.
(This example made in Laghouat Willaya, by specifying in the local region with 6 points of gathering) see Figure 2.24

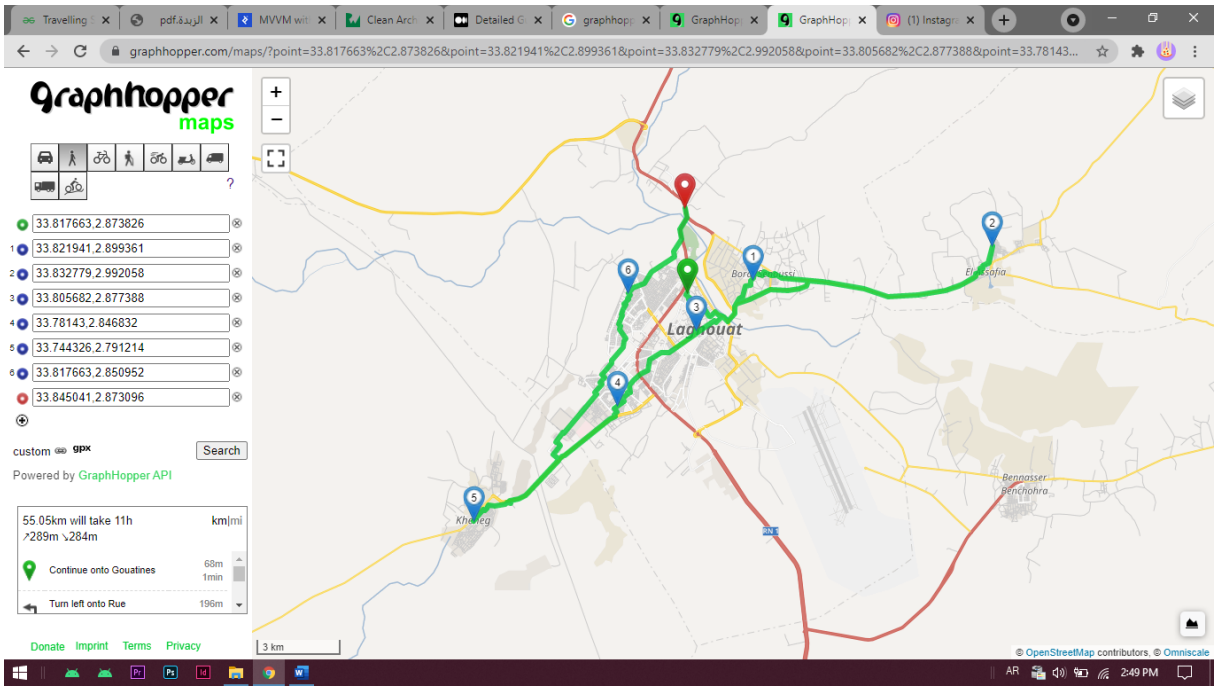

Figure 2.24: Simulation with GraphHopper of the Route Optimization API

GraphHopper can already offer a production ready road routing service which is also easy to set up. Furthermore, GraphHopper won't take high administration costs as it is already very robust and used under high load in some companies for several months. Last but not least GraphHopper is easy to customize and can satisfy a wide range of different business needs.

2.5 Conclusion

In this chapter, we have seen the abstraction of the functionalities of the platform in the form of UML diagrams that will form the basis of the next phase:
implementation, which we are going to see in the next chapter, also we were able to use existing solution to develop our API that will offer the best solution.

Chapter 3

Implementation & Strategy

3.1 Introduction

In this chapter we will talk about everything related to the implementation of the platform:

1. The Hardware used for the development and testing of the project
2. The software used for the development of this project.
3. The Architecture we chose to implement the project
4. Other android architectural patterns
5. The reason we chose MVVM architecture
6. System Functionality

3.2 Hardware Configuration

3.2.1 Development

Asus Vivobook X541UJ:

- CPU: Intel Core i5-6500U @2.7-3.5Ghz (2 cores, 4 threads, 4MB L3 cache)
- RAM: 8GB DDR4 @2133 MHz
- Disk: 256GB SSD

3.2.2 Testing

Xiaomi Redmi Note 9 pro

- Chipset: Mediatek Helio G90T @2.05 Ghz(8 cores)
- RAM: 6GB LPDDR4X RAM
- Storage: 64 GB UFS 2.1
- OS: Android 11 API level 29

3.3 Software

3.3.1 Android Studio

Android Studio is an IDE (Integrated Development Environment) made by google for the development of android applications. It is based on Jet-brain's Intellij IDE. It is available to download on the three major desktop operating systems: Windows, Linux and Mac OS.

3.3.2 Google Cloud

Google Cloud (also known as Google Cloud Platform or GCP) is a provider of computing resources for developing, deploying, and operating applications on the Web. Although its cloud infrastructure does serve as the host for applications, Google cloud is mainly a service for building and maintaining original applications, which may then be published via the Web from its hyper-scale data center facilities.
One of the most used service, is the Firebase, made by google, which offer the hosting of the mobile application services, and web applications.

3.3.3 NoSQL

NoSQL Database is a non-relational Data Management System, that does not require a fixed schema. It avoids joins, and is easy to scale. The major purpose of using a NoSQL database is for distributed data stores with humongous data storage needs, NoSQL is used for Big data and real-time web apps.[22]
NoSQL (not only sql) database technology stores information in JSON documents instead of columns and rows used by relational databases, and retrieving data and store it dynamically with not much rules of previous sql, nosql is capable to handle the data rapidly and respond to the demands.

Why we chose NoSQL database?

- The data changing over time and it is not structured.
- The data we need to read and write must be in real time.
- The data is growing continuously.
- Each document can have its own unique structure.
- It fits with firebase structure, the RealTime Database.

3.4 System Functionality

3.4.1 Client App

The functionalities shown here are available to all the users of the application:

3.4.2 Register Client

This screen is for user registration, where he will fill the phone number, and wait to receive the OTP code to confirm that he is a real client, then a new screen will appear to complete registration of full name, email, and Willaya, then clicking to creating account button, after that the operation of registration will be done successfully, it will display the Home Screen next. see Figure 3.1

3.4.3 Login Client

This is the first screen that greets the user after launching the app, if client is already have account, he will only confirm the phone number, if he doesn't have account, he can click down the Sign-Up button to take him to the registration section: see Figure 3.2

3.4.4 Home

This is the home screen of our application in which the user can Access easily, if the user is logged in then this is the first screen that's will be shown to him: see Figure 3.3 this Home Screen will show 4 main sections in the app

- Tracking a Parcel.
- Sending a Parcel.
- Showing the history of request and confirmed parcels.
- Showing general statistics about the client and cost parcels.

Figure 3.1: Register Screens

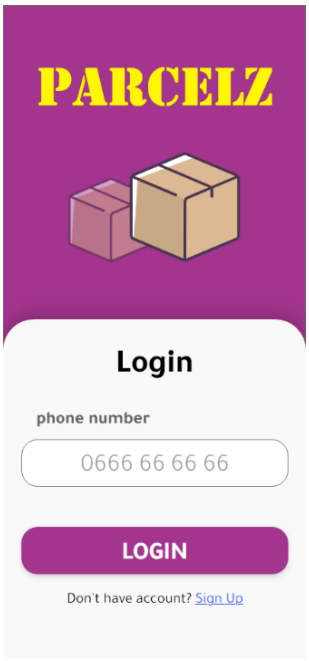

Figure 3.2: Login Screen

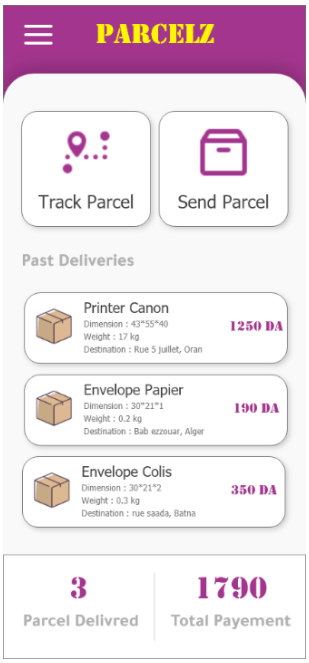

Figure 3.3: Main Frame User

3.4.5 Sending a Parcel

This section contains the general service of requesting a parcel to deliver by: see Figure 3.4 & Figure 3.5

- Specifying general data; title, dimension, type, and description.
- Specifying the pickup location from the map.
- Specifying the destination location from the map.
- Showing general details to confirm the request, and specifying if its frangible, and the delivery mode.

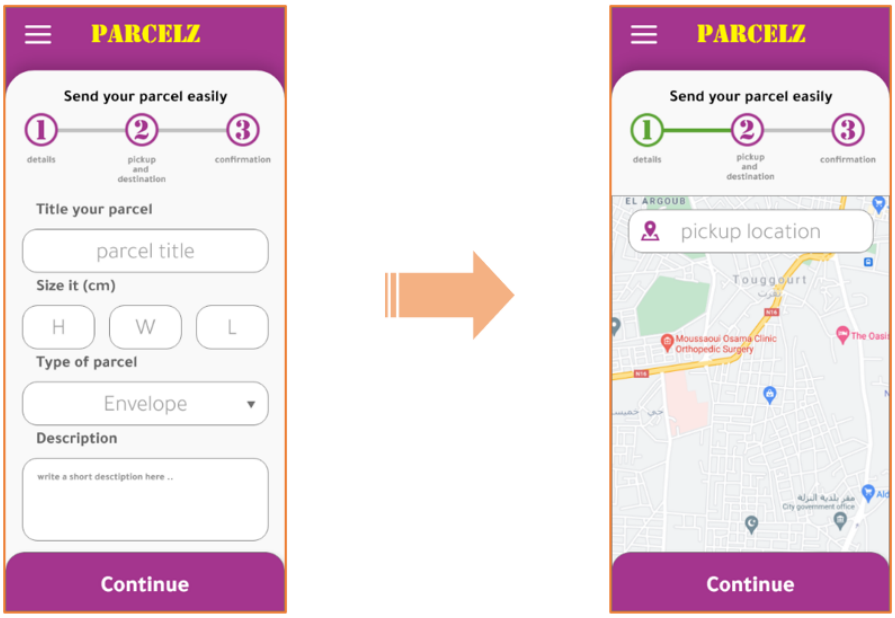

Figure 3.4: fill details of parcel request and location

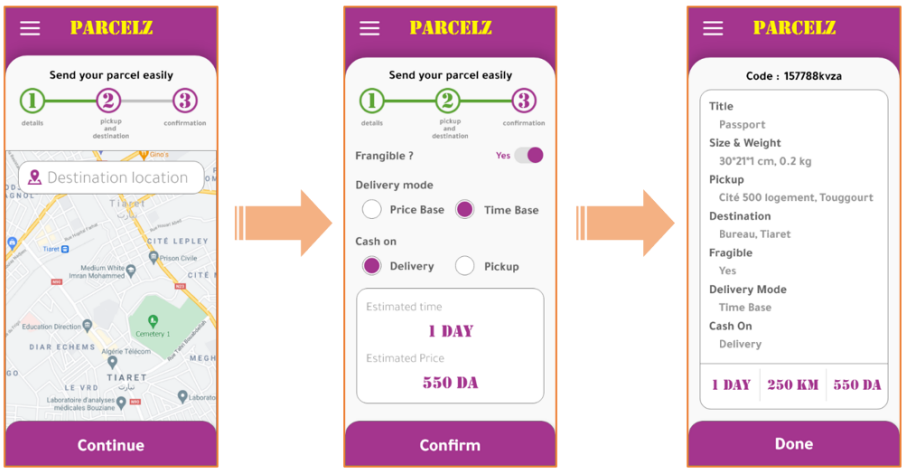

Figure 3.5: general confirmation details

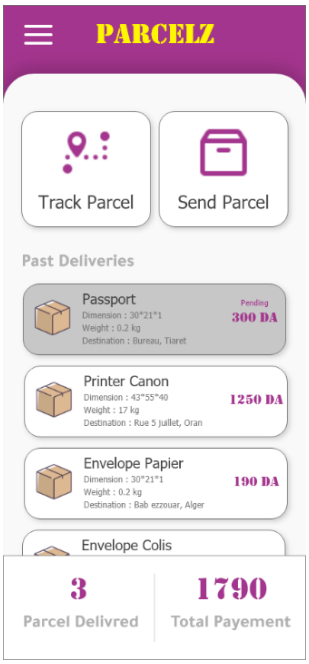

Figure 3.6: display request clients

Figure 3.7: Show Details of selected parcel

After confirmation the request it will display the confirmed details, with the unique code that has been generated automatically, after that in the Home screen, the section of history, as it will be marked with dark-grey color meaning that is not confirmed yet: see Figure 3.6

3.4.6 Tracking Parcel

This screen displays the details of searched parcel, and follow his state in real time: see Figure 3.7

3.5 Driver App

3.5.1 Register Driver

This screen is for driver registration, where he will fill the phone number, and wait to receive the OTP code to confirm that he is a real driver, then a new screen will appear to complete registration of full name, email, and Willaya, then clicking to create account button, after that the operation of registration will be done successfully. see Figure 3.8

3.5.2 Login Driver

This is the first screen that greets the Driver after launching the app, if driver is already have account, he will only confirm the phone number, if he doesn't have account, he can click down the Sign-Up button to take him to the registration section: see Figure 3.9

3.5.3 Register Car

After the last screen a new screen will appear, which is about register a car information: see Figure 3.10

Figure 3.8: Registration steps

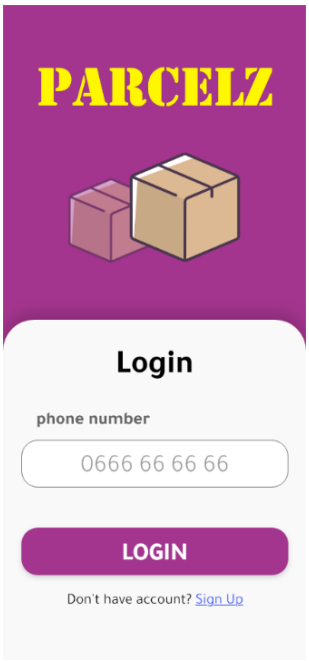

Figure 3.9: Login Screen

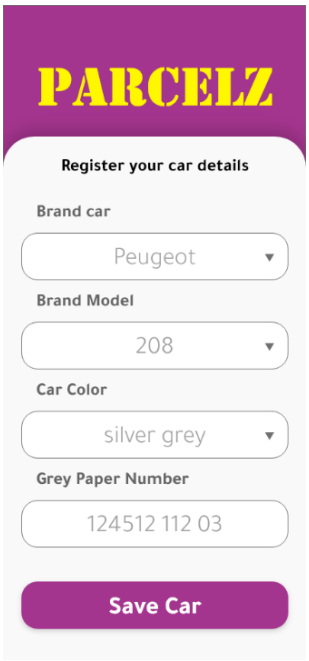

Figure 3.10: Register Car details for driver

3.5.4 Home Screen Driver

This screen has the activating button to show the status of driver if he is available. the status of driver will let him receive the requests for the nearby clients. see Figure 3.11

3.5.5 Accept request of client

This frame will show the details of parcel to deliver it with all the needed details. see Figure 3.12

3.5.6 Get My Path

This Frame will show all my accepted requests, to follow the pickup path, and start the trip. see Figure 3.13

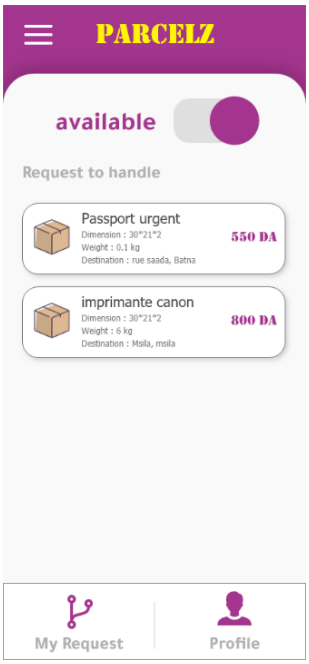

Figure 3.11: Home Screen of Driver App

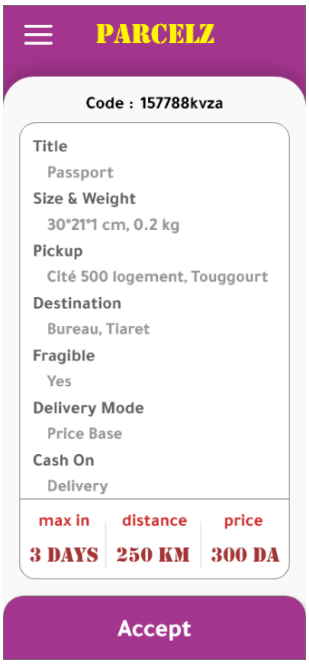

Figure 3.12: Accept request of client

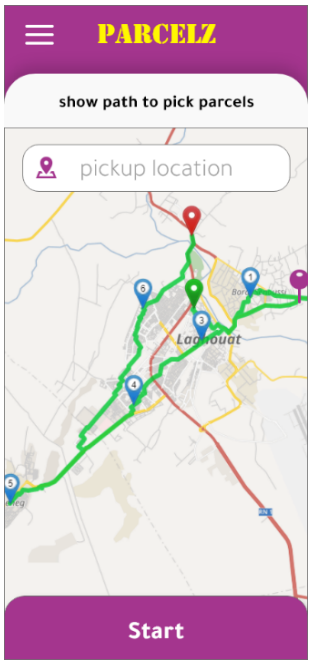

Figure 3.13: show path driver to start trip of pickup

General Conclusion

Our work was inspired by a lack of a real solution to the problem of local and national delivery. We aimed for our platform to be the link between possible clients and drivers to make the delivery service more widely available and convenient.

In the first chapter we talked about transport in general and we gave a general tool and view on it. Then we emphasized on the importance of crowd shipment, the effects and implications of the current technological development on the field. We then gave a view on some related solution as well as compared them to each other from different aspects.

In the second chapter we defined the structure of our platforms through UML diagrams that helped us make up our minds on how to implement the system's functionality. And the architecture and tools that will help us finishing the project.

Finally in the third chapter we listed the hardware used for the development and testing of this work and we detailed our choices in software, then we provided screenshots to further clarify the functionality of the system in more detail.

3.6 Future work

We plan to further develop this system in the future. Thinking to improve the work by adding the following functionalities, which is not available and not injected in our platform yet.
Future of same-day shipping: Using the Artificial Intelligence, we will be able to handle numerous data to solve the problem of latency, for example:
When a client wants to send his parcel from Laghouat to Algiers in just 1 day, this is going to be challenging, but using our solution, the system will be able to see the middle point drivers, and try to figure out who will pick the parcel from the next point of stop.
The Client parcel from Laghouat to Algiers in less than 24H.
The drivers' data:

- One driver going to Djelfa after 2 hours.
- One driver going from Djelfa to Algiers in the next 8 hours.
- One driver going from Djelfa to Setif in 3 hours.
- One driver going from Setif to Algiers in the next 7 hours.

3.7 Limitation and outlines

- One of the limitations, the project was not completely finished, hoping that the next year students will finish the study and increasing this solution to keep the work on.
- The MAPs API was not for free, and the number of queries are limited only for one month trial version.

Bibliography

[1] *Amazon Drivers Are Hanging Smartphones in Trees to Get More Work*. https://www.bloomberg.com/news/articles/2020-09-01/amazon-drivers-are-hanging-smartphones-in-trees-to-get-more-work. Accessed: 2021-06-08.

[2] *Android Modularization: Hexagonal Architecture with Kotlin and MVVM*. https://medium.com/@Marchosiax/android-modularization-hexagonal-architecture-with-kotlin-and-mvvm-part-1-db338833d6c2. Accessed: 2021-05-22.

[3] Jose Caceres-Cruz et al. "Rich vehicle routing problem: Survey". In: *ACM Computing Surveys (CSUR)* 47.2 (2014), pp. 1–28.

[4] *Cargomatic company*. https://cargomatic.com/about/. Accessed: 2021-06-15.

[5] *Crowdsourced delivery explained: making same day shipping cheaper through local couriers*. https://www.businessinsider.com/crowdsourced-delivery-shipping-explained. Accessed: 2021-05-12.

[6] *Data, context and interaction*. https://en.wikipedia.org/wiki/Data,_context_and_interaction. Accessed: 2021-05-23.

[7] Min Ding et al. "Emotional bidders—An analytical and experimental examination of consumers' behavior in a priceline-like reverse auction". In: *Management Science* 51.3 (2005), pp. 352–364.

[8] *Entity-control-boundary*. https://en.wikipedia.org/wiki/Entity-control-boundary#:~:text=The%20entity%2Dcontrol%2Dboundary%20(,in%20the%20use%2Dcase%20realization. Accessed: 2021-05-23.

[9] *GraphHopper*. https://en.wikipedia.org/wiki/GraphHopper. Accessed: 2021-05-30.

[10] *GraphHopper documentation*. https://en.wikipedia.org/wiki/GraphHopper. Accessed: 2021-05-31.

[11] Brian Kallehauge et al. "Vehicle routing problem with time windows". In: *Column generation*. Springer, 2005, pp. 67–98.

[12] Sohaib Lafifi. "Vehicle Routing Problems with Resources Synchronization". PhD thesis. Université de Technologie de Compiègne, 2015.

[13] Abdul Kadar Muhammad Masum et al. "Solving the vehicle routing problem using genetic algorithm". In: *International Journal of Advanced Computer Science and Applications* 2.7 (2011), pp. 126–131.

[14] John Miller, Yu Nie, and Amanda Stathopoulos. "Crowdsourced urban package delivery: Modeling traveler willingness to work as crowdshippers". In: *Transportation Research Record* 2610.1 (2017), pp. 67–75.

[15] *Official site of EMS*. https://www.ems.dz/a-propos/. Accessed: 2021-05-23.

[16] *Official site of Yalidine*. https://yalidine.com/#a-propos. Accessed: 2021-05-23.

[17] *Official site of Ziouane Express*. https://ziouane-express.com/page/about. Accessed: 2021-05-23.

[18] *Parcel delivery ;The future of last mile*. https://www.mckinsey.com/~/media/mckinsey/industries/travel%20transport%20and%20logistics/our%20insights/how%20customer%20demands%20are%20reshaping%20last%20mile%20delivery/parcel_delivery_the_future_of_last_mile.ashx. Accessed: 2021-05-27.

[19] *Screaming Architecture*. https://blog.cleancoder.com/uncle-bob/2011/09/30/Screaming-Architecture.html. Accessed: 2021-05-22.

[20] Yoshiaki Shimizu, Tatsuhiko Sakaguchi, and Jae-Kyu Yoo. "A hybrid method for solving multi-depot VRP with simultaneous pickup and delivery incorporated with Weber basis saving heuristic". In: *Journal of Advanced Mechanical Design, Systems, and Manufacturing* 10.1 (2016), JAMDSM0004–JAMDSM0004.

[21] *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, 2002. ISBN: 9780898714982 9780898718515. DOI: 10.1137/1.9780898718515.

[22] *Types of NoSQL Databases*. https://www.guru99.com/nosql-tutorial.html. Accessed: 2021-06-20.

[23] *Understanding Onion Architecture*. https://www.codeguru.com/csharp/csharp/cs_misc/designtechniques/understanding-onion-architecture.html. Accessed: 2021-05-22.

[24] *What Is Crowdsourcing and How Does It Work?* https://www.thestreet.com/personal-finance/education/what-is-crowdsourcing-15026002. Accessed: 2021-06-14.