DEMOCRATIC AND PEOPLE'S REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH MOHAMED BOUDIAF UNIVERSITY - M'SILA

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
N°:…….

DOMAIN: **Mathematics And Computer Science**
FIELD : **Computer Science**
OPTION: **SIGL(Information System And Software Engineering)**

A Dissertation in Fulfillment for the Requirements of the Degree of Master

Directed by: Daikach Said

Entitled

Towards an Information System based Optimization for Fuel Delivery Management (NAFTAL/M'sila study case)

Defended before the jury composed of:

Mr. Brahimi Mahmoud	University of M'sila	Supervisor
	University of M'sila	Rapporteur
	University of M'sila	Examiner

Academic year: 2020/2021

DEMOCRATIC AND PEOPLE'S REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH MOHAMED BOUDIAF UNIVERSITY - M'SILA

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
N°:…….

DOMAIN: **Mathematics And Computer Science**
FIELD : **Computer Science**
OPTION: **SIGL(Information System And Software Engineering)**

A Dissertation in Fulfillment for the Requirements of the Degree of Master

Directed by: Daikach Said

Entitled

Towards an Information System based Optimization for Fuel Delivery Management (NAFTAL/M'sila study case)

Defended before the jury composed of:

Mr. Brahimi Mahmoud	University of M'sila	Supervisor
	University of M'sila	Rapporteur
	University of M'sila	Examiner

Academic year: 2020/2021

Dedication

بفضل الله في المقام الأول والأخير.

آباؤنا وأمهاتنا من أعظم هدايا الله لنا ، دعونا نكرمهم باحترام

وحب وتقدير ولا ننسى في صلواتنا.

إلى أمي أمباركة العظيمة وإلى أبي زيان العظيم وإلى جدي سعيد

العظيم الذين ساعدوني ودعموني وشجعوني ونصحوني وضحوا

بالكثير من أجل أن أكون رجلا صالحا ، شكرا لكم على كل شيء.

إلى إخواني الأعزاء خالد ، وسارة الذاني سعداني وشركاني الأوقات

الجيدة والسيئة في حياتي وسعداني على اجتياز الأوقات الحرجة ولا ننسا

ابنة اختي صغرة حبيبة.

لأصدقاء الأعزاء توفيق خزار، الفاتح محمد ياسين نواري ،

محمد عبد المجيد سعيدي ، ماجد جنيدي، محمد الأمين مفتاح

، وكثير منهم ربما لم

اذكركم لاكنكم موجدون بي فعل في ذهني وقائمة طويلة .

بفضل كل عائلاتنا وأصدقائنا الذين ساعدونا ودعمونا

من قريب أو بعيد عبر حياتنا...

شكرا لكم جميعا

Acknowledgements

whatever I say, I won't be able to express my deep thanks to the teaching staff in my faculty specially my supervisor **Dr.Brahimi Mahmoud** who accepted to be my supervisor and helped and supported me with the necessary information in my dissertation. It is impossible for me to list his efforts.

In addition, I would like to extend my gratitude to the Senior Developer **LAKHDAR MAHMOUDI** who has helped, supported and shown me. Clearly, he feels responsible for me. Really, I appreciate that.

And I thank all the NAFTAL workers who have helped me whenever you need information.

I thank my friends Mohamed Abdelmadjid Saidi ,
Mohamed Elamine Meftah And Yacine Hedjouli **.**

And thank you all.

Summary

General introduction

General introduction 2

Chapter I : Optimization

1 Introduction 5
2 What is Combinatorial Optimization? 5
2.1 Combinatorial Problems 5
2.1.1 Optimisation Problems 5
2.2 Why optimization ? 6
3 Solving optimization problems 6
3.1 Identify the problem 6
3.2 Modeling the problem 6
3.2.1 Solving models 6
4 How can we Solve Hard Problems? 7
4.1 Example: Traveling Salesman Problem 7
4.2 Solving the Traveling Salesman Problem 7
4.3 The Two Poles of Problem Solving 8
4.4 Optimization: Getting Good Approximate Solutions in Reasonable Time 8
4.4.1 Constructive Heuristics 8
4.4.2 Local Search 9
4.4.3 Population-Based Metaheuristics and their Hybrids 9
4.4.4 Exact Methods vs. Approximate Methods 9
4.4.5 Everything is an Anytime Algorithm 10
5 Formal Definition of Optimization Problem 10
5.1 Single-Objective Optimization Problem 10
5.2 More Examples 11
5.2.1 Travel to Shanghai 11
5.2.2 Function Minimization 11
6 Classification of optimization problem 12
6.1 Mathematical program 12
7 Optimization vs decision 13
7.1 Problem 13
7.2 Different versions of problems 13
7.3 Polynomial versus exponential 14
7.4 Polynomial versus exponential 14
7.5 Complexity 14
7.5.1 class P 14
7.5.2 class NP 14
7.5.3 classes P and NP 15

7.5.4 NP-completeness 15
7.6 Decision vs optimization problems they are 15
8 Optimization methods 15
8.1 Exact methods 16
8.1.1 Linear, Continuous Problems 16
8.1.2 Linear, Discrete Problems 17
8.2 Approached methods 17
8.2.1 Introduction 17
8.2.2 Heuristic 17
8.2.3 Metaheuristics 17
8.2.4 Heuristics vs metaheuristics 18
8.2.5 Classification of metaheuristics 18
8.3 Exact method vs approximate method 19

Chapter II : Company Presentation (NAFTAL)

1 Introduction 21
2 History and organizational chart of the host organization 21
2.1 Presentation of NAFTAL SPA 21
2.1.1 History 21
2.1.2 Presentation of NAFTAL 22
2.2 Missions and Means of the NAFTAL Company 22
2.2.1 Main missions 22
2.2.2 Objectives 23
2.2.3 Company resources 23
2.2.4 Commercial network 24
2.2.5 Designation and missions of the main structures 25
2.2.6 Products Marketed by NAFTAL 25
2.3 The organizational chart and organization of the company NAFTAL SPA 27
2.3.1 Organization chart of the NAFTAL Company 27
2.3.2 organization of the NAFTAL SPA company 29
3 Presentation of the Marketing Division 32
3.1 The Organization Chart of the Marketing Branch 32
3.2 Missions of the Marketing Branch 33
4 Presentation of the Finance and Accounting Direction 34
4.1 Organization chart of the Finance & Accounting Direction 34
4.2 Main missions of the Finance & Accounting Direction 35

Chapter III :The Conceptual Study

1 Introduction 37
2 UML Diagrams 37

3 Use Case Diagram 38
3.1 Purpose of Use Case Diagrams 38
3.2 Use Case Diagram Representation 39
4 Class Diagram 40
4.1 Purpose of Class Diagrams 40
4.2 Class Diagram Representation 41
5 Interaction Diagram 42
5.1 Purpose of Interaction Diagrams 42
5.2 Sequence Diagram Representation 42
5.2.1 Client Sequence Diagram Representation 43
5.2.2 Chauffeur Sequence Diagram Representation 44
6 Activity Diagrams 45
6.1 Purpose of Activity Diagrams 45
6.2 Distribution Stations Activity Diagrams Representation 46
6.3 Analyze Orders Activity Diagrams Representation 47

Chapter IV : Implementations

1 Introduction 49
2 Technical environment and tools 49
2.1 Microsoft Visual Studio 49
2.2 Desktop .Net Framework 50
2.2.1 NET Framework 50
2.2.2 C Sharp 50
2.3 Windows Presentation Foundation 51
2.4 DevExpress 52
2.5 SQL Server 2019 52
2.5.1 Presentation 52
2.5.2 Strengths of SQL Server 52
2.6 Material Design 52
3 Test 53
3.1 Home 53
3.2 Distribution Stations 54
3.2.1 List Distribution Stations 54
3.2.2 In addition Distribution Stations 54
3.2.3 in Updated Distribution Stations 55
3.2.4 Order Statistics 56
3.3 Client 57
3.3.1 Client List 57
3.3.2 In addition client 57

3.3.3 in updated Client 58
3.3.4 Order Statistics 58
3.4 Trucks 59
3.4.1 Trucks‘ List 59
3.4.2 In addition Trucks 59
3.4.3 in updated Trucks 60
3.4.4 Maintenance 60
3.4.5 Shipping Movement 63
3.5 Driver 64
3.5.1 Driver List 64
3.5.2 in addition driver 64
3.5.3 in updated Driver 65
3.6 : Order 66
3.6.1 Add Capacity 66
3.6.2 Order List 66
3.6.3 Orders Analysis 67
3.6.4 Travel 67
4 Conclusion 68
General Conclusion
General Conclusion 70

List Of Tables

Chapiter I : Optimization

Table 1.1: classification of optimization problem..... 13

Chapiter II : Company Presentation (NAFTAL)

Table 2.1: Market share of service stations..... 24

LIST of FIGUERS

Chapiter I : Optimization

Figure 1.1: A sketch of the time-quality trade-off we make with optimization on the example of random and optimal solutions for the Traveling Salesman Problem. 8
Figure 1.2: Metaheuristic Evolutionary algorithm Genetic algorithm Mathematical optimization 19

Chapter III :The Conceptual Study

Figure 3.1: Use Case Diagram Representation. 39
Figure 3.2: Class Diagram Representation. 41
Figure 3.3: Client Sequence Diagram Representation : 43
Figure 3.4: Chauffeur Sequence Diagram Representation : 44
Figure 3.5: Distribution Stations Activity Diagrams Representation : 46
Figure 3.6: Analyze Orders Activity Diagrams Representation : 47

Chapter IV : Implementations

Figure 4.1: Splash Screen 53
Figure 4.2: Homepage 53
Figure 4.3: List Distribution Stations 54
Figure 4.4: List Client on Distribution Stations 54
Figure 4.5: Add Location on Distribution Stations 55
Figure 4.6: Modify Distribution Stations 55
Figure 4.7: Distribution station order statistics 56
Figure 4.8: Distribution station order statistics 57
Figure 4.9: Add client information 57
Figure 4.10: Modification client information 58
Figure 4.11: client order statistics 58
Figure 4.12: Trucks ‘list 59
Figure 4.13: Add Truck and Trailer 59
Figure 4.14: Modify Truck and Trailer 60
Figure 4.15: List Control maintenance truck 60
Figure 4.16: List Control oil Filter Changing truck 61
Figure 4.17: List Control trailer greasing truck 61
Figure 4.18: List Control greasing truck 62
Figure 4.19: List Control oil changing truck 62
Figure 4.20: List Control tire changing truck 63
Figure 4.21: List Control shipping movement 63
Figure 4.22: Driver List 64
Figure 4.23: Add driver 64
Figure 4.24: Update driver 65
Figure 4.25: Add capacity 66
Figure 4.26: Order List 66
Figure 4.27: Orders Analysis 67
Figure 4.28: Travel list 67

General Introduction

General introduction :

The constant increase in conductors, resulting in significant fuel consumption, has led to increased demand for distributors of this vital substance and thus to the problem of transport and delivery of this substance, so distribution and delivery companies have had to find practical solutions to how to deliver this product to the customer in the right time and location at the optimal quantities and at the lowest possible cost.

In this project, we propose a method to solve the problem of transport planning in a national production and transport institution for petroleum materials and their derivatives. (NAFTAL), with the aim of optimizing this function by preparing a plan for the transfer of the largest quantity of product (fuel) from supply to demand to reduce the total costs incurred by the enterprise, eventually proposing an optimal transport plan based on what is available to the enterprise.

Objective of this work:

The objective of the work entitled Design & implementation of the information system for fuel delivery management (NAFTAL/M'sila study case)"is the creation of an intelligent information system to search for minimum trajectories with as many requests as possible and with the lowest cost and best coverage for all regions, taking into account factors. (Time, quantity required, quantity available, distance), which makes it easier for the distribution company to decide how, where and when to deliver from one point to another in a way that is available, easy and at a minimum cost of Time.

Memory organization:

With regard to the methodological plan for our work, our thesis is divided into four theses.

Chapters :

- Chapter I will discuss some concepts of improvement and TSP (Travelling salesman problem) as well as some important concepts.
- Chapter II sets out a general definition of a company (NAFTAL) in which we describe its structure, features and, in particular, how it operates administratively, as well as certain schemes and materials.
- Chapter III presents the conceptual study of our system, which is the main structural and behavioral graphs that give the user a way to visualize the manipulation of the system.
- Finally, we will discuss the required tools used to implement our system; we will also present the main services provided by our solution.

Chapter I : Optimization

1 Introduction

With the big progress in mathematics and computer science, optimization provides now an enormous assistance for solving many real problems encountered in our daily life. The managing and the optimization of the transport under some criteria represent one of the fields that benefit strongly from these solutions. In this chapter, we will present the optimization field with their fundamental concepts and techniques.

2 What is Combinatorial Optimization? :

Combinatorial analysis is the mathematical study of the arrangement, grouping, ordering, or sele'ction of discrete objects, usually finite in number.Traditionally, combinator ialists have been concerned with questions of existence or of enumeration.[1]

2.1 Combinatorial Problems :

Combinatorial problems arise in many areas of computer science and other disciplines in which computational methods are applied, such as artificial intelligence, operations research, bioinformatics, or electronic commerce. Prominent examples are tasks such as finding shortest/cheapest round-trips in graphs, finding models of propositional formulae, or determining the 3D-structure of proteins.[2]

2.1.1 Optimisation Problems :

Many practically relevant combinatorial problems are optimization problems rather than decision problems. *Optimization problems* can be seen as generalisations of decision problems, where the solutions are additionally evaluated by an *objective function* and the goal is to find solutions with optimal objective function values.

combinatorial optimisation problem, we distinguish two variants:

- the *search variant*: given a problem instance, find a solution with minimal (or maximal, respectively) objective function value;
- the *evaluation variant*: given a problem instance, find the optimal objective function value (i.e., the solution quality of an optimal solution).

Additionally, for each optimisation problem, we can define:

- *associated decision problems*: given a problem instance and a fixed solution quality bound b, find a solution with an objective function value smaller than or equal to *b* (for minimisation problems; greater than or equal to *b* for maximisation problems) or determine that no such solution exists.[2]

2.2 Why optimization ?

Optimization can provide many advantages such as:

- Countless applications -science, engineering, business, economics
- Some form optimization is used in every company
- Every process can potentially be optimized
- Minimize: time, cost, risk ……
- Maximize: profit, quality, efficiency …….

A combinatorial optimization problem is either a maximization problem or a minimization Problem with an associated set of instances. Without loss of generality we will restrict ourselves to minimization problems, because every maximization problem can easily be converted into a Minimization problem.

3 Solving optimization problems:

The solving of an optimization problem needs three steps which are [3]:

3.1 Identify the problem:

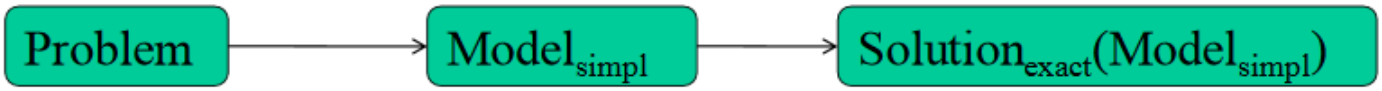
- Identify the decision problem
- Identify internal / external objectives
- Determine the input (parameters)
- Determine Constraints

3.2 Modeling the problem :

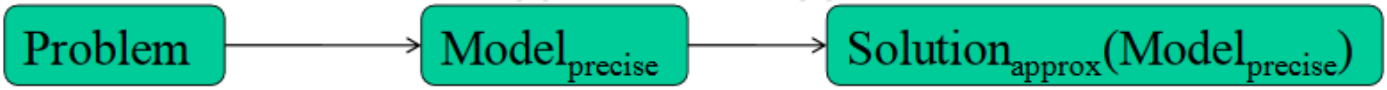
Model is simplification of reality. the quality of the solution depends on the quality of the model. the founded solution is for the abstract model and not the original .

There are two common approaches:

- Simplified model, exact approach



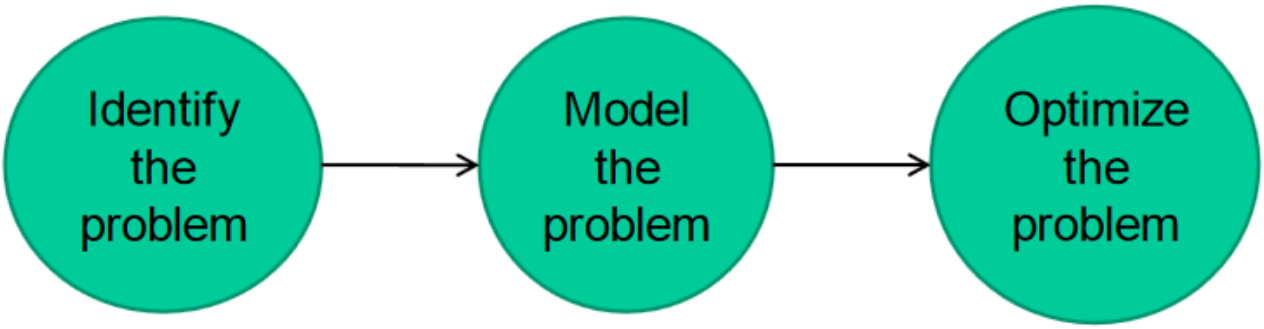
- Precise model, approximate approach :



3.2.1 Solving models :

After we have defined a model of the original problem, the model can be solved by some kind of algorithm (usually an optimization algorithm). An algorithm is a procedure (a finite set of well-defined instructions) for accomplishing some task. an algorithm starts in an initial state and

terminates in a defined end-state. The goal of an algorithm to find a solution (either specific values for the decision variables or one specific decision alternative) with minimal or maximal evaluation value.



4 How can we Solve Hard Problems?: [4]

4.1 Example: Traveling Salesman Problem :

One of the most well-known hard combinatorial optimization problems is the Traveling Salesman Problem (TSP), which can be formulated as follows: Given is a set of n cities and the distances between each pair of two cities. What is the shortest possible tour that visits all cities and returns back to its starting point? A candidate solution is a permutation of the n cities, i.e., a sequence which contains each city exactly once. It describes the order in which the cities are to be visited and the way back is implied.

There are |X|=n! many such permutations, which turns out to be quite a large number even for moderate n as shown already in Figure 1, with 10! being 3'628'800 and 20!≈2.4×1018. It turns out that we do not yet have an exact algorithm which can solve any possible TSP in 1.9999n steps or less. And chances are that we will never have one.

4.2 Solving the Traveling Salesman Problem :

But what does it mean that we cannot guarantee to find the solution of a TSP in reasonable time? First of all, the word "solution" is used here in a very strict and tight sense, in the sense of "exact" and "globally optimal": A solution to the TSP in this classical sense is only the shortest possible round-trip tour through the n cities.

But what if we widen this definition a little bit: Actually, every possible visiting sequence of the n cities which contains each city exactly once (and returns back to the first city) could be considered a solution. We then simply have solutions of different quality, which we thus call candidate solution.

Even if we have a TSP with 10000 cities, we actually can obtain one such solution very quickly, within n steps, simply by writing down all n cities directly in the same order as in the problem specification. Although there is a tiny chance that this might already be the exact best-possible solution to the TSP, it very, very likely is will be a very very bad (candidate) solution. Anyway, we thus have established we may not be able to guarantee to find the best possible solution for hard problems quickly, but we definitely can find some candidate solution.

4.3 The Two Poles of Problem Solving :

We now know the two polar opposites of problem solving: On one hand – for most problems – we can find some random solution of poor quality quickly. On the other hand, finding the best-possible solution may take far too long to be feasible.

Such a situation is not satisfying and if we face it, what we would want is something in between.

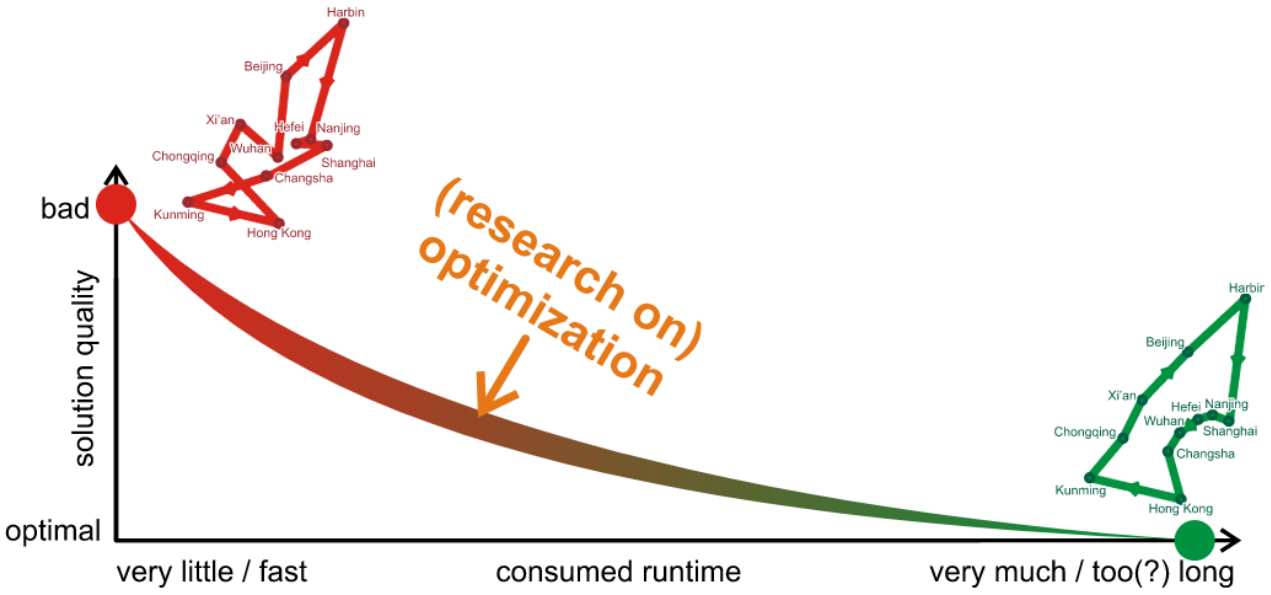


Figure 1.1: A sketch of the time-quality trade-off we make with optimization on the example of random and optimal solutions for the Traveling Salesman Problem.

4.4 Optimization: Getting Good Approximate Solutions in Reasonable Time :

This "in between" is one of the things that optimization does. On the left end of the scale in Figure 1, we find the afore-mentioned random solutions which do not make use of any information from the problem. But we usually have a lot of information: In the TSP, for instance, we know the distances between the cities.

4.4.1 Constructive Heuristics :

Constructive heuristics are methods which usually create a single candidate solution in a meaningful, one-shot fashion by utilizing information from the problem. Heuristics for the TSP, we

could apply a nearest neighbor heuristic which creates a tour step-by-step by first picking a (random) city and then iteratively adding the city nearest to the previously added one to the tour. This means we need something in the scale of 0.5n2 steps, as every time we want to add a city, we have to check all not-yet added cites. It will normally produce much better solutions than creating random solutions, but usually also not find the best possible tour.

4.4.2 Local Search :

If we move up the time scale, we arrive at the family of local search algorithms, which start with a single solution and try to iterative improve it. Assume we have created a solution for the TSP using a heuristic. We could now check whether we can swap two cities in the tour. If this results in a better tour, we take it. If not, we try again. Usually, there exist many different such refinement operators.

In the TSP, besides swapping cities, we could reverse a leg of the tour or rotate the cities a leg to the left or right.

Also, there exist several ways for applying the moves in each step of the local search: We could first try all n(n-1)/2 city pairs that can be swapped and actually swap the one which improves the tour the most, if any. Or we could try the swaps and pick the first improving swap.

A local search cannot guarantee that we find the optimal solution either. We may arrive in a dead end, a local optimum. If we use the city-swap operator, this could be a tour which cannot be improved by swapping two cities but needs a more complex change.

Thus, local search methods can be enhanced with a variety of additional measures, such as restarting if no improvement can be found (but, of course, remembering the best solution so far).

4.4.3 Population-Based Metaheuristics and their Hybrids :

In order to prevent running into local optima to quickly, the concept of population-based metaheuristics exists, mainly known under the labels Evolutionary Computation, Evolutionary Algorithm, and Genetic Algorithm (which all have slightly different meanings).

Here, a set multiple of solutions in maintained and iteratively refined. However, these algorithms are usually way to slow to be of practical use. They are thus hybridized with local search: The resulting Memetic Algorithms can guard against getting stuck too early while having the speed of local search.

4.4.4 Exact Methods vs. Approximate Methods :

Exact optimization algorithms naturally solve a problem to optimality, incurring the potential large, exponential runtime. It should be noted, however, that the runtime is only exponential in the

worst case, e.g., in TSPs with particularly nasty distributions of cities. In many real problems, exact algorithms might be able to run in a reasonable time. The Concorde TSP Solver has solved a TSP with 85900 cities to optimality.

On the other hand, although metaheuristics and local search usually do not guarantee to find the optimal solution, they might actually do so, and they might do quite quickly. The LKH algorithm is a local search which has found the solution for the same 85900-city problem – but probably much faster than Concorde: To solve a 13'509 city-problem, Concorde needed three months on a cluster while LKH did the job in half a day on a desktop computer. Of course, not guarantee that the solution it found was optimal. We only know this because of Concorde.

Thus, both exact and approximate methods have drawbacks and advantages.

4.4.5 Everything is an Anytime Algorithm :

Finally, it should be noted that most optimization methods are anytime algorithms. These are algorithms that can be stopped and provide an approximate solution at any time. This is easy to see for local search, for example: Once it has an initial solution, we can always stop it and just take the best solution it found so far. This is also true for all metaheuristics and, interestingly, also many exact methods.

Branch and bound algorithms iteratively and recursively divide the search space into regions and only search those which may improve upon the best solution they have so far. They repeat this until there is no such region left, which then means that the best-so-far solution is the best-possible solution. But, of course, we can stop them earlier and just take the best-so-far solution (losing the guarantee to get the best-possible solution in the process.

From this perspective, most optimization algorithms become comparable, processes of the development of solution quality over time …but this is another post altogether.

5 Formal Definition of Optimization Problem :

So far, we have discussed roughly what an optimization problem is, a set of possible solutions X and a way to compare them. Then we have explored different types of algorithms for solving such problems. Let us now give a formal definition of optimization problems and some more examples.[4]

5.1 Single-Objective Optimization Problem :

How can we compare solutions? The most common way to do so is to define an objective function f which assigns a real value to each candidate solution $x \in X$, i.e., $f: X \mapsto R$. f represents an

optimization criterion, a quality or utility measure that we can understand. We call this basic problem type a single-objective optimization problem.

- Minimization
- Maximization

5.2 More Examples :

5.2.1 Travel to Shanghai :

Let's say I want to travel from my home in Hefei to Shanghai. Obviously, there are several options to do so. Candidate solutions are, for example, x_1="hard seat ticket for slower train K462", x_2="second class seat in faster train D3016, x_3= "airplane MU5468 of the China Eastern Airlines", and so on. The space of solutions is then $X=x_1,x_2,x_3,\ldots$. As objective function f1, I could choose the required travel time in hours. It turns out that $f(x_1)\approx5.4$, $f(x_2)\approx3.4$, and $f(x_3)\approx1.2$. Amongst the three listed example choices, the airplane would be the best one – under criterion f_1.

Since are only so-and-so many possible means of transportation available that I can use from my home in Hefei to Shanghai, the solution space x_1 of this problem is finite. Problems with finite solution space are called combinatorial optimization problems.

The solution space X of this example has no clear structure and there are no direct relationships between the candidate solutions. There may be classes of solutions (such as "slow trains") but that's about all what we can see at first glance. Due to the lack of structure in the solution space, I would probably have to enumerate all possible solutions to find the optimum.

5.2.2 Function Minimization :

Another, this time purely artificial, optimization task is to find the minimum of a five-dimensional variant of Ackley's function $a(x) = \mathbb{R}^5 \mapsto \mathbb{R}$ with

$$a(x) = e + 20 - 20\exp\left(-0.2 \times \sqrt{\frac{1}{5}\sum_{i=1}^{5} x_i^2}\right) - \exp\left(\sqrt{\frac{1}{5}\sum_{i=1}^{5}\cos(2\pi x_i)}\right)$$

The space of possible solutions here is a subset of the five-dimensional Euclidean space $\mathbb{R}^5$, e.g., $X_2 = [-10,10]^5$.A candidate solution would be a five-dimensional real vector. If we want to find the minimum of a(x), then we can simply use it as objective function directly, i.e., set f(x)=a(x).

This time, we have uncountable infinitely many potential candidate solutions $x \in X$. This task thus is a so-called continuous or numerical optimization problem. Interestingly, X has a much clearer

structure than the previous solution space: I can take an existing solution $x_1 \in X$ and modify it a little bit, maybe by adding a vector of very small numbers, and get new, similar point x_2 as result. If $x_2 \approx x_1$, then probably also $f(x_2) \approx f(x_1)$. This feature is called causality. It is quite important for optimization and I will maybe talk about it at another time. Furthermore, f is steady and symmetric. If we have two reasonably similar solutions (vectors) x_1 and x_2, we can combine them in a meaningful way in order to get another solution x_3 which similar to both (and hopefully unites the different good characteristics of both) by setting

$x_3 = 0{,}5\ (x_1 + x_2)$. All of these properties may be useful when trying to minimize f_2 and may help us to find the solution (which happens to be $(0,0,0,0,0)^T$) in a finite amount of steps.

6 Classification of optimization problem :

Most optimization models commonly used : [3]

- Mathematical programing
- Constraint programing

6.1 Mathematical program:

Minimise / maximise $f(x_1,\ldots..x_n) \longrightarrow$ objective function ($\mathbb{R}^n \mapsto \mathbb{R}$) subject to :

- ➢ Inequality constraints

  $g_1(x_1,\ldots.x_n) \geq b_1$

  .....

  $g_m(x_1\ldots\ldots x_n) \geq b_m$

- ➢Equality constraints

  $h_1(x_1\ldots\ldots x_m) = c_1$

  ……

  $h_m(x_1\ldots\ldots..x_m) = c_m$

Optimization problem modeled using mathematical programing can be classified

according to the nature of :

- The objective function f and constraint function g_i and h_i
- The decision variable x
- The type of input parameters

Classification	Refers to
Linear vs nonlinear programming	The type of objective function and / or Constraint
Discrete vs continuous programming	The type of variable
Stochastic vs deterministic programming	The nature of parameters in the constraint Or the objective function
Finite vs non finite programming	The number of decision variable
Constrained vs non-constraint	Whether the constraints are defined or not

Table 1.1: classification of optimization problem

7 Optimization vs decision:

7.1 Problem :

- A problem just contains a description of the problem together with the parameters that describe the input of the problem.
- Values have not been assigned to the parameters.
- For example: the Partition problem
  - Given n non-negative integral values $a_1, a_2, \dots, a_n$ does there exist a subset S of the index-set$\{1,2, \dots, n\}$,such that

$$\sum_{j \in s} a_j = \frac{1}{2} \sum_{j=1}^{n} a_j$$

  - The input parameters are and n and $a_1, a_2, \dots, a_n$

7.2 Different versions of problems :

- Decision problems.
  - Answer is yes or no.
- Optimization problems.
  - Answer is a number representing an objective value.
- Construction problems.
  - Answer is some object (set of vertices, function, …).
- Counting problems.
  - How many objects of some kind exists?

7.3 Polynomial versus exponential **:**

- An algorithm with running time $O(n^k)$, where K is a given constant, is called a **polynomial** algorithm.
- An algorithm with running time, $O(c^n)$, where c>1 is a given constant, is called an **exponential** algorithm.

7.4 Polynomial versus exponential :

- In general a polynomial algorithm is preferred over an exponential algorithm, because of the scalability (effect of increasing the size of the problem on the running time).
- For many problems polynomial algorithms exist, but for many others they have not been found yet. Major question: are we to blame when we cannot find a polynomial algorithm for some problem ?
- This has been a major topic of research in the area of **Computational Complexity** .

7.5 Complexity :

7.5.1 class P :

- A decision problem belongs to the class **P** if there is a solution algorithm with a running time that is polynomial in the input size
  - In practice: encoding and corresponding concrete problem is assumed very implicitly.

7.5.2 class NP :

A decision problem belongs to the class NP if:

- Any solution y leading to `yes' can be encoded in polynomial space with respect to the size of the input x.
- Checking whether a given solution leads to `yes' can be done in polynomial time with respect to the size of (x,y).

A) Examples of many NP problems :

- Hamiltonian Path,
- Maximum Independent Set,
- Vertex Cover,
- Satisfiability,
- Integer Linear Programming
  - Easy to show for 0/1 programming,
  - non-trivial in general

7.5.3 classes P and NP :

- Were originally and formally defined in terms of Turing machines

Alternative definition of NP :

- Class of decision problems P, for which there exists a **Non-Deterministic Turing Machine** that can solve any yes instance in polynomial time.

7.5.4 NP-completeness :

A) A problem P is NP-complete, if:

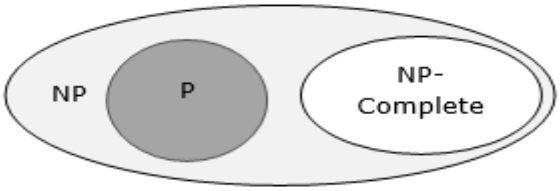
1. $P \in NP$.
2. For every $P' \leq_P P$.

B) A problem P is NP-hard, if:

1 For every $P' \in NP : P' \leq_P P$.

NP-hardness also used as term

- for problems that are not a decision problem, e.g. the optimization version of an NP-complete decision problem
- for problems that are 'harder than NP'.



7.6 Decision vs optimization problems they are :

- An optimization problem can be turned into a decision problem by introducing a threshold value y.
- In case of a minimization problem M , the decision variant becomes:
  - Given an instance of M together with a threshold value y , does there exist a feasible solution with outcome value ≤y?
- If optimization problem M can be solved in polynomial time, then its decision variant can be decided in polynomial time.
- If an optimization problem H has an NP-complete decision version L, then H is called NP-hard.[5]

8 Optimization methods :

Generally, optimization algorithms can be divided in two basic methods:

- Exactes
- Approchees

8.1 Exact methods :

deterministic and probabilistic algorithms [6]

- deterministic

  Linear, Continuous Problems

  Linear, Discrete Problems

- Probabilistic

  Heuristic Optimization Methods

8.1.1 Linear, Continuous Problems :[6]

Since many problems of practical relevance can be modeled as linear optimization problems, linear optimization methods are an important research theme in OR and are well established in many companies for planning and optimization purposes. Problems are linear if

- the objective function depends linearly on the decision variables and
- all relations among the variables are linear

In linear problems, we assume a number of limited resources (each decision variable describes the consumption of one of the resources) and an objective function that models how the quality of the overall solution depends on the use of the limited resources. Representative examples of linear problems are resource allocation problems, production problems, or network flow problems.

Linear problems are commonly called linear programming (LP) problems. The word “programming” is not used in the sense of computer programming (where programming usually means writing a set of computer instructions) but in the sense of “planning”. This is due to the fact that “linear programming” was introduced very early and before the word “programming” was associated with computer software.

This section describes linear optimization problems and gives a brief overview of concepts that can be used for solving such problems :

- Linear Optimization Problems
- Simplex Method
- Simplex and Interior Point Methods

8.1.2 Linear, Discrete Problems : [6]

Representative optimization methods for combinatorial optimization problems are decision tree-based enumeration methods and cutting plane methods. The use of decision trees allows us to formulate the process of problem solution as a sequence of decisions on the decision variables of the problem. Common methods working with decision trees are uninformed and informed graph search algorithms like depth or A*-search, branch-and-bound approaches, or dynamic optimization. The optimization methods presented in this section can not only be used for linear problems but can also be applied to non-linear optimization problems :

- Integer Linear Problems
- Uninformed and Informed Search
- Branch and Bound Methods
- Dynamic Programming
- Cutting Plane Methods

8.2 Approached methods : [7]

8.2.1 Introduction :

Despite the evolution of computer science and mathematical methods, there are optimization problems with a prohibitive size of the admissible solution space. In many real applications it is impossible to find the optimal solution due to the dynamics of the studied system, the constraints and the number of variables. In view of these difficulties, most specialists use approximate search techniques. The use of an approximate optimization method does not guarantee an exact optimal solution but a good solution in a reasonable computation time.

8.2.2 Heuristic :

A heuristic is a computational method that quickly provides a feasible solution, not necessarily optimal or exact, for a difficult optimization problem. It is a concept used among others in combinatorial optimization, graph theory, algorithm complexity theory and artificial intelligence . In general, a greedy algorithm is precisely a heuristic, for example:

- the return of change by successive divisions;
- the nearest neighbor method for the traveling salesman problem.

8.2.3 Metaheuristics :

Metaheuristics are a set of methods used in operations research to solve optimization problems known to be difficult. Solving a combinatorial optimization problem is to find the optimum of a function, among a finite number of choices, often very large. There are many concrete applications, whether in industrial production, transport, economics, wherever there is a need to minimize digital functions, in systems where a large number of parameters simultaneously intervene.

8.2.4 Heuristics vs metaheuristics :

In order to improve the behavior of an algorithm in its exploration of the solution space of a given problem, the use of a heuristic method (from the Greek verb heuriskein, which means "to find") makes it possible to guide the process in its search for optimal solutions.

In [FEI 63] a heuristic is defined as an estimation rule, a strategy, a trick, a simplification, or any other kind of system which drastically limits the search for solutions in the space of possible configurations. In [NEW 57] we specify that a heuristic process can solve a given problem, but does not offer the guarantee of doing so. In practice, certain heuristics are known and targeted to a particular problem.

Metaheuristics, for its part, is placed at an even more general level, and intervenes in all situations where the engineer does not know an efficient heuristic to solve a given problem, or when he considers that he does not have the time needed to determine one.

8.2.5 Classification of metaheuristics :

Metaheuristics can be classified according to various criteria, but most often, they are categorized according to the number of solutions generated: we distinguish between single-solution metaheuristics and population-based metaheuristics of solutions. However, other classifications that may be overlapped to fully understand the method in question(Figure 1.2)

A) Metaheuristics which use a candidate solution:

- Local search LS
- Simulated annealing SA
- Tabu search TS
- Greedy randomized adaptive search GRASP

B) Metaheuristics which use a population solution:

B .1 Evolutionary algorithms :

- Evolution strategies ES
- Evolution programming EP
- Genetic algorithm GA
- Genetic programming GP

B .2 Swarm intelligence algorithm:

- Particle swarm optimization PSO
- Ant colony optimization ACO
- Artificial bee colony ABC

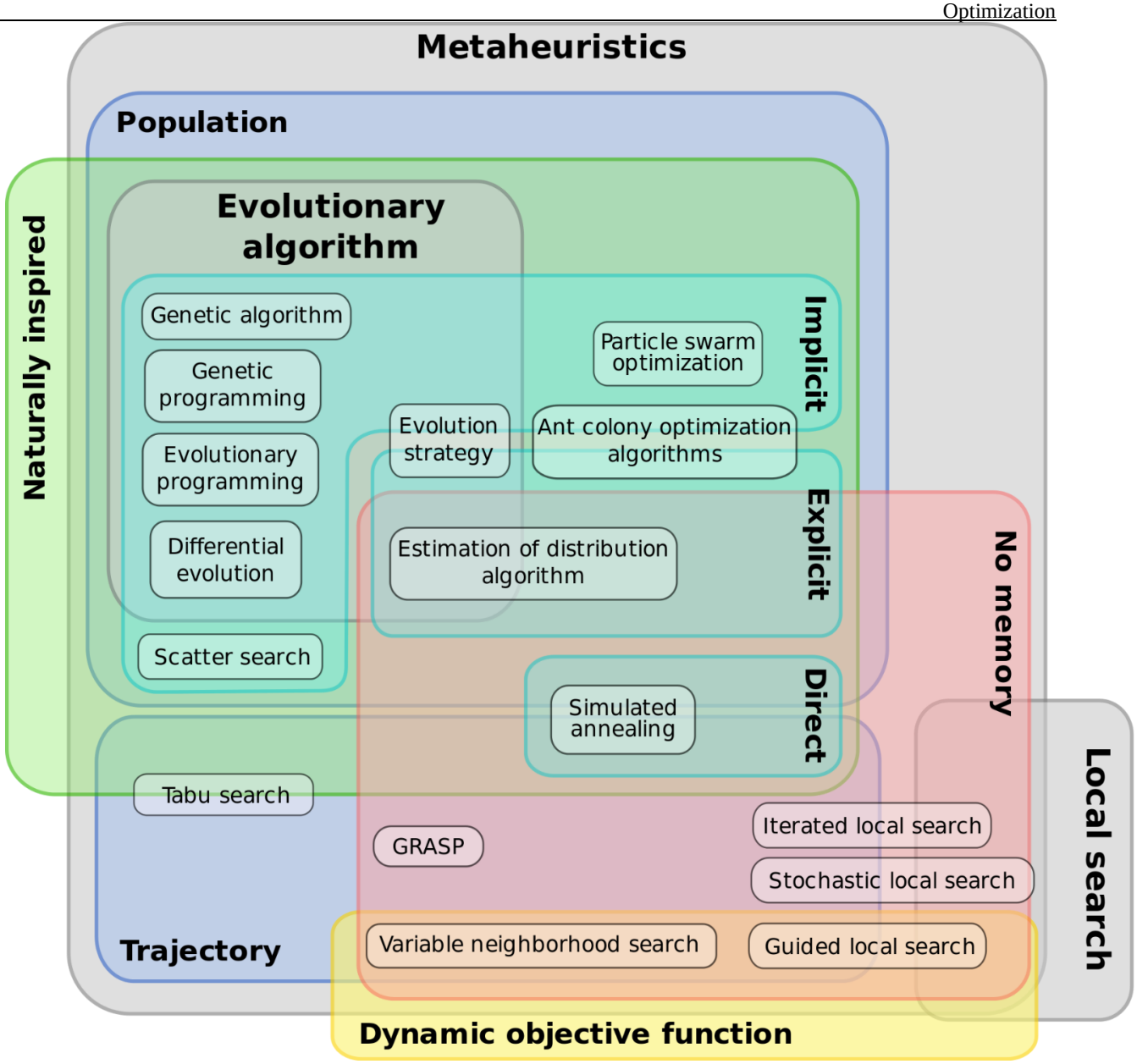


Figure 1.2: Metaheuristic Evolutionary algorithm Genetic algorithm Mathematical optimization

8.3 Exact method vs approximate method :

An exhaustive search by explicit enumeration of all the solutions to solve a difficult optimization problem is unthinkable because of the computation time involved. In the case of the traveling salesman problem, for example, the search space grows in (n-1) !, where n is the number of cities to visit, which quickly exceeds the computational capacities of any computer. With only 50 cities, 49 will have to be evaluated! trips, is 6.080×10^{62} trips. This is the combinatorial explosion. Nevertheless, the resolution of such an optimization problem can be done in an exact way, by carefully modeling the problem, then by applying an ad-hoc algorithm, which immediately rules out the examination of certain configurations, of which we know d 'Or already that they can not be optimal.

Chapter II : Company Presentation (NAFTAL)

1 Introduction

In this chapter we will present our host organization (NAFTAL) in order to clarify how the transport optimization can assist this company in their daily tasks. In addition, the role of this chapter is to determine the main keys and tasks that must be considered in our solution.

2 History and organizational chart of the host organization:

2.1 **Presentation of NAFTA**L **SPA:**

2.1.1 History:

In 1952 oil was discovered for the first time, during colonial times. The hydrocarbons sector was owned by foreign firms ESSO, TOTAL, SHELL, BP…

This is how SONATRACH was created on 31-12-1963 by decree n ° 63-491, its mission was to transport and market Hydrocarbons. Subsequently, it was extended its powers by decree n ° 66-296 of 22-09-1966 in the field of research, production, transport, processing and marketing of hydrocarbons.

To carry out all of these activities, it has made considerable efforts in investing, training its own executives and creating its own service companies (example ENAFOR, GTP, ENTP, NAFTEC and NAFTAL

From the **SONATRACH** company, the **ERDP** company was created by decree No. 80/101 of April 06, 1980.

Entered into activity on 1 1982, it is responsible for the refining industry and the distribution of petroleum products under the acronym NAFTAL (the name of NAFTAL comes from "NAFT" which means "petroleum" in Arabic, and "AL" in reference. to "AL DJAZAIR".

In 1987, the refining activity was separated from the distribution activity, so the name of the company changed following the separation of activities. By decree 87-189 “NAFTEC” is responsible for oil refining and “NAFTAL” is responsible for the marketing and distribution of petroleum products. This name comes from the combination of terms:

- NAFT: Oil
- AL: Algerian

Within the framework of the global restructuring plan for the hydrocarbons sector, the intermenstrual committee decided on November 19, 1997 to attach, at the organizational level, the company "NAFTAL" to **SONATRACH**.

For this purpose it was transformed into a joint stock company (SPA) a 100% subsidiary of SONATRACH on April 18, 1998 with a share capital of 6,650,000,000 DA and increased to 15,650,000,000 DA at the end of 2002

2.1.2 Presentation of NAFTAL :

A) NAFTAL identity :

- **NOM :** The National Company for the Marketing and Distribution of Petroleum Products.

LEGAL NATURE: SPA**.**
SOCIAL FUND: Its capital is 15,650,000,000 DA.
HEAD OFFICE: Route des dunes BP 73, msila Alger.
NUMBER OF EMPLOYEES: : Around 30,607 : 6,256 contract workers and 24,351 permanent.

TURNOVER: 307 Billion DA
NET RESULT 2012: 7 737 357 107 DA

2.2 Missions and Means of the NAFTAL Company :

2.2.1 Main missions :

NAFTAL's main mission is the distribution and marketing of petroleum products on the national market including those intended for aviation and marine, LPG, fuels, solvents, aromatics, paraffins, bitumen, lubricants and tires.

It supplies the country with nearly (7.5 million tones) of oil equivalent, or 51% of the national final energy consumption.

Also, its missions are to:

- Store, transport and / or have transported any petroleum product marketed on national territory;

- Ensure the application and compliance with measures relating to industrial safety, safeguard and protection of the environment, in conjunction with the organizations concerned;
- Carry out any market study on the use and consumption of petroleum products;
- Define and develop an audit policy, (in each branch there is an Audit department) design and implement integrated information systems;
- Develop and implement actions aimed at an optimal and rational use of infrastructures and means;
- Ensure the application and compliance with measures related to the internal security of the company in accordance with regulations.
- Develop a quality brand image.

2.2.2 Objectives :

Being a public company, NAFTAL has a lucrative objective to gain a profit and a social objective which is that of ensuring the needs of citizens in products and their availability across the national territory.

2.2.3 Company resources :

To accomplish its missions, NAFTAL has considerable human and material potential which is made up as follows:

A) Material Potential :

- Overall, it has:
- 67 distribution and storage centers for NAFTAL products;
- 52 multi-product centers (CMP);
- 23 fuel centers (CBR);
- 13 Pneumatic lubricating centers;
- 41 LPG filling centers, and 07 bulk LPG centers;
- 14 Bitumen Centers;
- 55 aviation and marine refueling depots;
- 49 LPG storage relay depots;
- 1,576 service stations including 901 belonging to private PVA;
- 3,250 distribution vehicles and 1,750 handling equipment;
- 750 km of LPG and Fuels pipeline.

B) Human Potential:

The global workforce of the company NAFTAL SPA is 30,607 Agents:

- 24 165 permanent staff

- 5,442 temporary agents (contractual)
  - 596 Senior managers
  - 6,319 executives
  - 10,196 Masters
  - 13,496 Executions

2.2.4 Commercial network :

NAFTAL has 1,847 service stations which constitute the points of sale to customers. This network provides 85% of fuel sales. These service stations are operated under (04) legal statuses:

- Direct management (335 GD):

NAFTAL assets managed directly by company agents. These stations cover 33% of the market.

- Free management (339 GL):

As the company's assets managed by third parties, their market share is 11%.

- Ordinary resellers (275 RO):

Stations apartment to third parties contractually bound to NAFTAL by equipment loan, it covers 7% of the market.

- Authorized point of sale (604 P.V.A):

Carried out entirely by private investors, they provide 49% of the market.

The following table summarizes these shares as follows:

LEGAL STATUS	MARKET SHARE
G.D	33%
G.L	11%
R.O	7%
P.V.A	49%

Table 2.1: Market share of service stations

Source : NAFTAL

2.2.5 Designation and missions of the main structures:

CODE	DESIGNATION	MISSION
D.E.S.P.E	Executive Department Strategy, Planning and Economy	Assist the GD in the development of a company policy, the planning of medium and long-term objects
D.E.F	Executive Finance Department	Cumulate the units Coordination and Centralization of the Company's accounts
D.E.R.H	Human Resources Executive Department	Responsible for development and concerning training, staff movement.
BRANCHE CLPB	Fuels, Lubricants and Bitumen Division	Development and monitoring of national program policies by market and activities of Fuels, Lubricants, Tires, special products and bitumen.
DIVISION GPL	Liquefied Petroleum GAS Division	Management of butane and propane products
DIVISION AVM	Division Aviation - Marine	Supply and Marketing of products for aviation and navy

Source : NAFTAL

2.2.6 Products Marketed by NAFTAL:

A) Fuels :

They are mixtures of hydrocarbons of mineral or synthetic origin obtained either by distillation of crude oil or by mixing certain constituents in suitable proportions.

Distillation makes it possible to collect different petroleum cuts, each representing a well-defined product.

The fuels sold are:

- Normal Essence
- Super Essence
- Unleaded gasoline
- Diesel fuel
- G.P.L / C

- Kerosene
- Heavy fuel oil

These product categories supply:

- Diesel engines
- Internal combustion engines.
- Large, slow diesel engines.

B) Lubricants :

lubricants are made exclusively from a mixture of mineral base oils derived from petroleum or Synthetic oils or sometimes both types of oil at the same time, added of course to various chemicals called additives.

Lubricants are classified according to the following ranges:

- HME: petrol engine oil
- HMD: diesel engine oil
- HTR: transmission oil
- SPA: automotive specialty oil
- HIN: industrial oils
- GRS: fat.

The main applications of lubricants are:

- Maintenance of industrial equipment
- Technical components
- Production
- Transport.

C) Bitumen :

It is a refinery product, it is obtained by the vacuum distillation of the residue from atmospheric distillation, following which a viscoelastic residue of black color is obtained at the bottom of the vacuum column.

Bitumen occupies a special place because it is used, not as fuel, but as building materials. It will therefore not be destroyed, but will remain in the works in which it is used, among other things, in the construction and maintenance of pavements.

The bitumen market is highly dependent on road maintenance and development works funded by the state and local communities.

D) L.P.G :

L.P.G, or liquefied petroleum gas, is a mixture of butane and propane in proposals which provides it with liquid storage under reasonable pressure under extended temperature conditions.

L.P.G comes from oil and gas production fields where they are most often flared, and especially from refining plants.

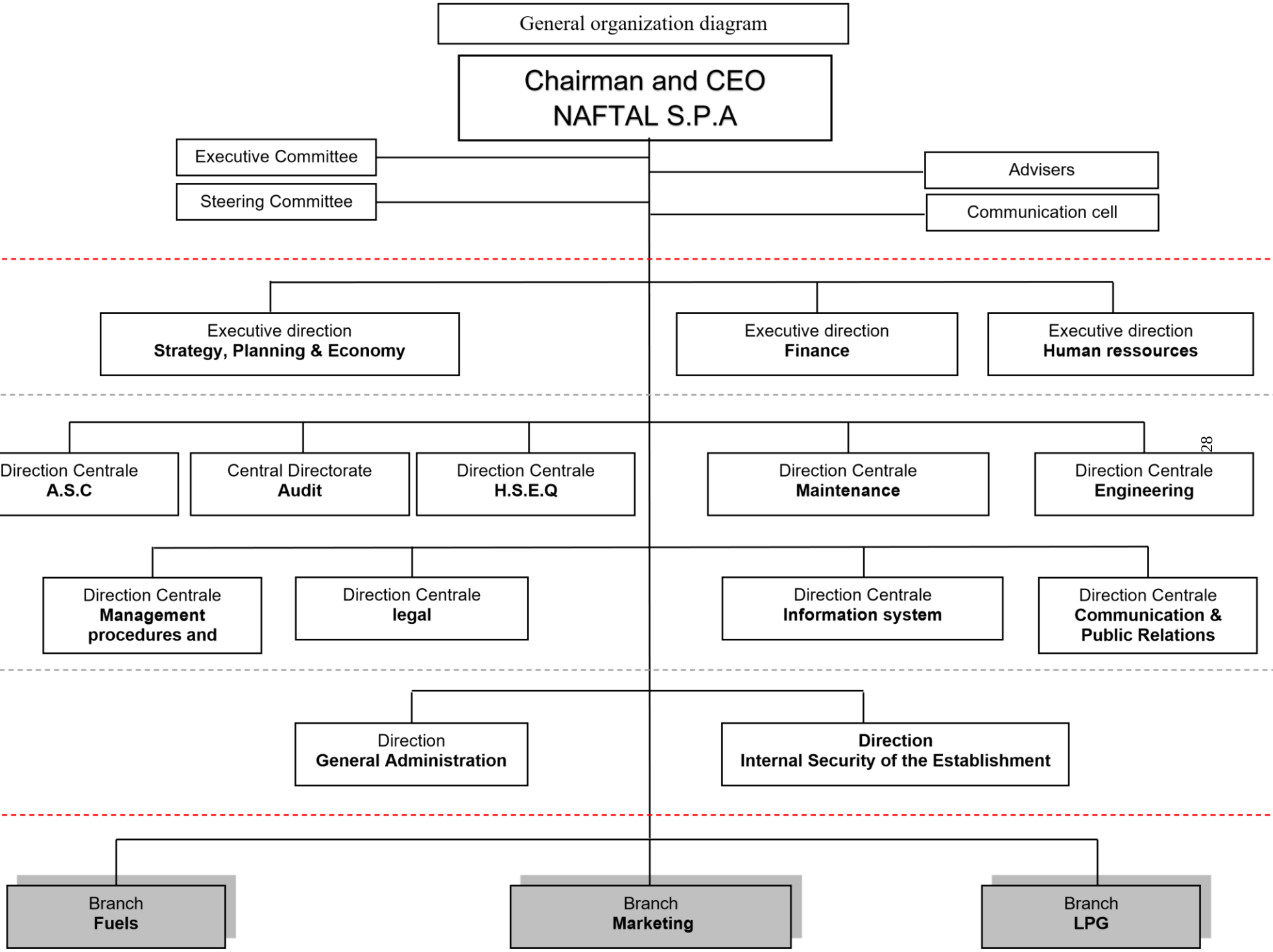
Butane and propane are subsidized by the state to facilitate consumption by various industrial economic activities (propane) and by domestic uses (butane).

E) The Tires : see section 4

2.3 The organizational chart and organization of the company NAFTAL SPA :

2.3.1 Organization chart of the NAFTAL Company :

General organization diagram



2.3.2 organization of the NAFTAL SPA company:

The purpose of the organization function is to group the resources of the company in an orderly manner and to distribute the individuals according to an acceptable model allowing them to fulfill the tasks assigned to them.

This model must establish relationships of authority between the various structures, it is a question of dividing the overall objective of the company into various individual tasks and grouping them into services, departments, directorates, under the responsibility of a manager to whom the necessary authority will be delegated.

NAFTAL's organization is structured around central structures responsible for defining the policy for monitoring and controlling the company's activities, and operational structures around petroleum products.

These operational structures are responsible for distributing all the products marketed by NAFTAL in the fields of influence of one or more wilayas.

The organization of NAFTAL is made up of three main structures, namely:

- General structure
- Functional structure
- Operational structures

A) General structure:

The general management of the company is ensured by the Chairman and Chief Executive Officer, it is responsible for policy and general orientations, and is made up of:

- A Steering Committee .
- An Executive Committee .
- Counselors .

B) Functional structures of NAFTAL:

The company's activities are decentralized and cover all distribution needs, through distribution units (Districts). It also benefits from regional logistical support through three maintenance units (UNM East and West the DMR) and four port units (UNP).

EXECUTIVE DIRECTIONS:

- **Strategy, Planning & Economy Executive Department**: Responsible for the development of the corporate strategy planning policy.

- **Executive Finance Department**: Responsible for defining and controlling the Company's financial, accounting and legal policies,
- and constantly ensure that the quality of NAFTAL's accounts is maintained with transparency.
- **Human Resources Executive Department**: Responsible for developing and monitoring the Company's human resources policy.

CENTRAL DIRECTIONS:

- **Procedures & Management Control Department**: Development of management and control procedures applied by the Company.
- **Audit Department:** Coordinate audit missions at the level of the central and operational structures of the Company, and ensure the application of NAFTAL rules and procedures.
- **Hygiene, Health, Environment & Quality Department**: Develop the company's H.S.E.Q policy and determine the modalities for its implementation.
- **Information System Department**: Participate in the company's IT development plan.
- **Research & Development Department**: Participate in the development of major research with the aim of improving the activities of the Company.
- **Social & Cultural Affairs Department**: Ensures the monitoring and management of the Company's social and cultural works.
- **Communication & Public Relations Department**: Contribute to
- Identification of the communication resources necessary to achieve the Company's objectives (as part of a coherent strategy)

SUPPORT DIRECTIONS:

- **General Administration Department**: Ensures the administrative and financial management of the Company's headquarters as well as personnel management in accordance with the regulations in force.
- **Internal Security Department of the Establishment**: Ensure the protection of the Company's assets as well as the safety of staff.

C) OPERATIONAL STRUCTURES: there are three branches:

- Fuels branch
- G.P.L branch
- Marketing branch

C .1 Fuels branch

The fuel branch ensures the supply and storage of petroleum products in all NAFTAL centers.

The maintenance and control of all transport vehicles.

C .2 LPG branch

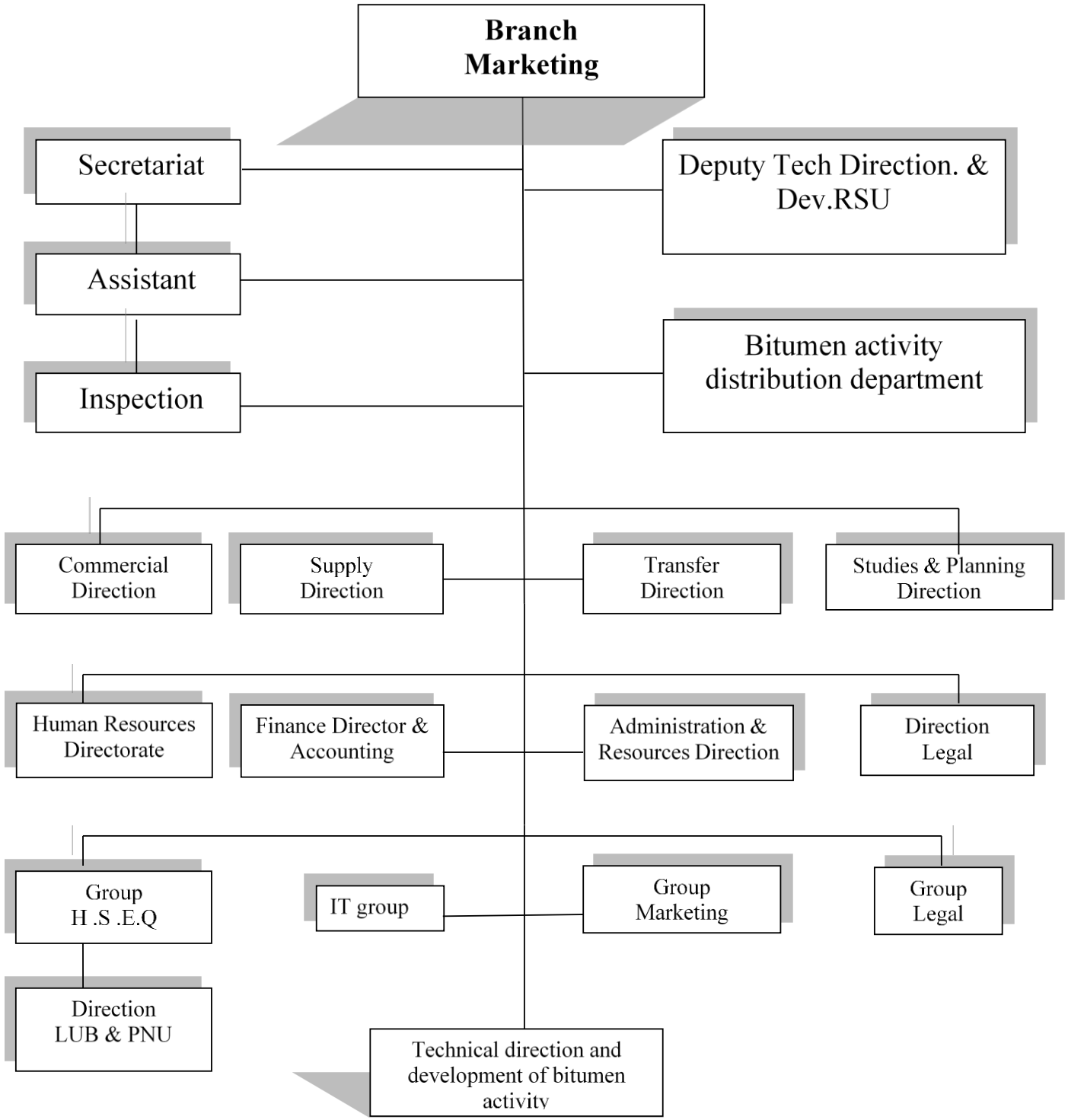
- Ensures the availability of the LPG product and its distribution throughout the country, especially during the winter period.
- Periodic checks on all LPG points of sale to ensure the quality of services and compliance with safety standards for citizens .

C .3 Marketing branch

- Its mission is the Marketing of petroleum products Fuels, Lubricants, Tires and Bitumen and other products intended for the automobile or the automobile

3 Presentation of the Marketing Division :

3.1 The Organization Chart of the Marketing Branch



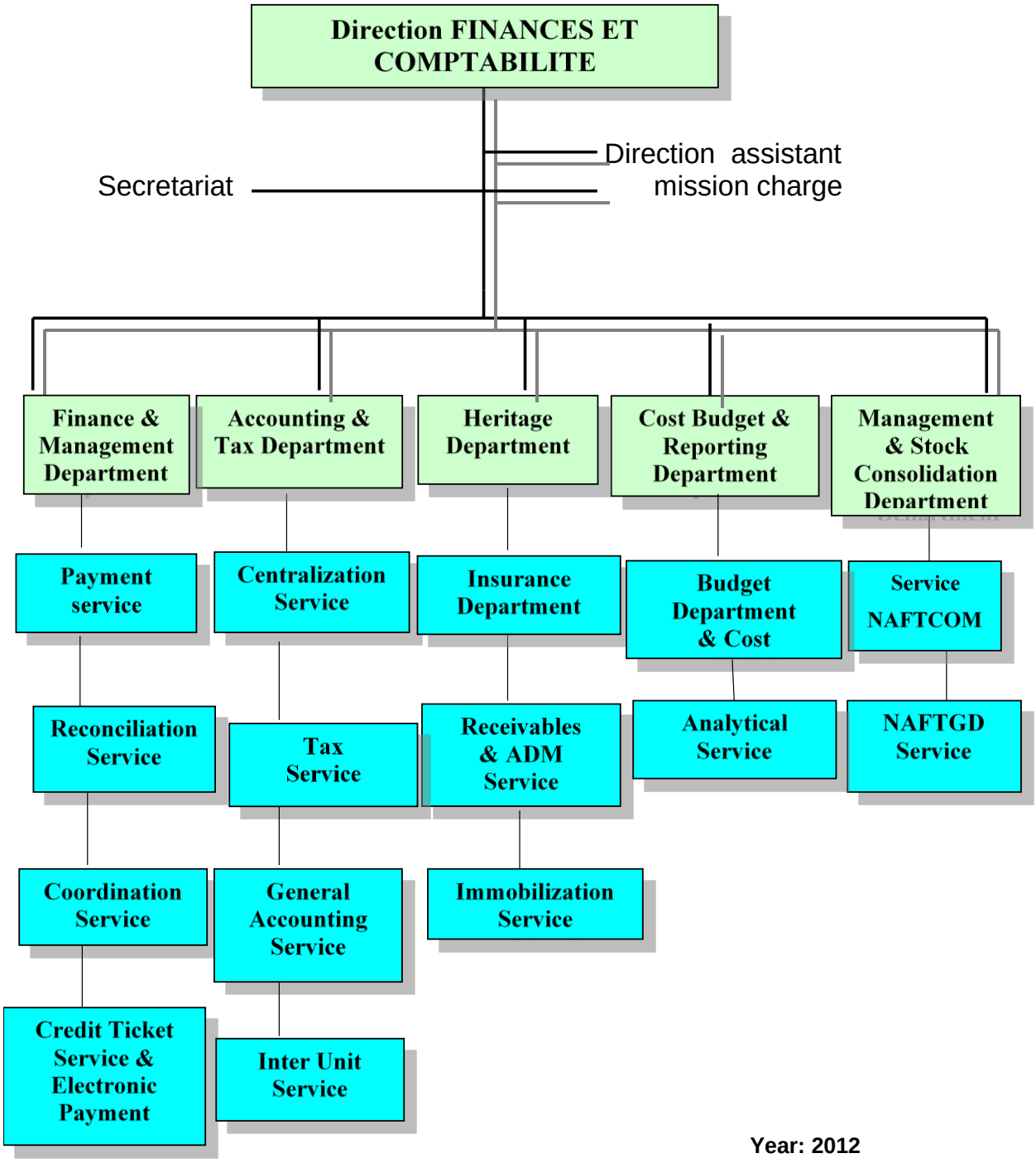
3.2 Missions of the Marketing Branch:

The marketing branch is responsible for:

- Market petroleum products, Fuels, Lubricants, Tires and Bitumen and other products intended for the automobile or motorist.
- Propose all measures to adapt the organization of the branch to the strategic vision of the company.
- Put in place efficient management systems and clean up and update the nomenclature of workstations;
- Provide services to all points of sale.
- Manage, monitor and control relationships with fuel, lubricants, tires and bitumen customers (network of service stations, large consumers & other customers).
- Develop and implement a network marketing strategy (GD, GL, PVA & services).
- Manage, control and monitor the sales network in Direct management.
- Carry out regular inspections to ensure the quality of services, compliance with safety and management standards.
- Evaluate and proceed with the purchase of equipment necessary for the maintenance and renovation of the network.
- Develop and implement a network development strategy (modernization, rehabilitation and compliance).
- Promote the development of LPG / Fuel.
- Initiate and coordinate communication actions intended to promote and popularize products.
- Initiate, direct and supervise market studies.
- Control and evaluate the sales force of Districts (Units).
- Promote a policy of improving the quality of service and customer reception.
- Define a production plan by gas station, LPG / C conversion center (Sirghaz) and other points of sale and monitor the execution.
- Ensure that the brand image and the quality of services are maintained at all points of sale.
- Ensure the financial balance of marketing activities and, if necessary, recommend and implement corrective measures.
- Analyze, consolidate and recommend for the approval of the hierarchy the operating and investment plans and budgets of the Marketing Branch.

4 Presentation of the Finance and Accounting Direction :

4.1 Organization chart of the Finance & Accounting Direction :



4.2 Main missions of the Finance & Accounting Direction:

- Define, enforce and control the financial, accounting and tax policies of the units of the branch.
- Coordinate and monitor the financial situation of the branch (treasury, accounting, taxation, insurance, etc.).
- Ensure that the branch's accounts are kept properly by ensuring that the work of analyzing, updating and revising the accounts for the year and producing the balance sheet is carried out in accordance with the standards and best conditions .
- Ensure the application within the branch of working procedures applicable to the various areas of activity of the management;
- Ensure the development and execution of financing plans;
- Assist the decentralized units in the management of the branch's fixed assets.
- ensure the financial compliance of commercial transactions.
- constantly ensure the development of professional knowledge of its staff by offering appropriate training actions.
- Empower staff in the performance of their tasks, encourage and promote the development of professional knowledge and skills.
- constantly ensure that the quality of the branch's accounts is maintained.

Chapter III : The Conceptual Study

1 Introduction:

The conceptual study is one of the most important steps in any computer science project, to conduct conceptual study we need to follow a conceptual method that responds to the user's requirements and gives a satisfied results of the user's needs. In this chapter we tackle the conceptual step of our solution using the UML language as modeling way.

2 UML Diagrams

Actually there aren't a lot of choices to deal with the conception problem and we don't have to think too much because UML offers a standard way to visualize the system's architectural blueprints, The Unified Modeling Language - UML - is OMG's most used specification, and the way the world models not only application structure, behavior, and architecture, but also business process and data structure.

UML, along with the Meta Object Facility (MOF), also provides a key foundation for OMG's Model-Driven Architecture, which unifies every step of development and integration from business modeling, through architectural and application modeling, to development, deployment, maintenance, and evolution, so in this chapter we will be interested of the conceptual study of the system.[8]

In order to give the user a clear vision of our system we use a combination of two kinds of diagrams, the first one is structural diagrams, and the second one is Behavioral diagrams. From the structure diagrams we choose two diagrams:

1. Use case Diagram.
2. Class Diagram.

And from the behavioral diagrams we choose also two diagrams:

1. Activity Diagram.
2. Sequence Diagram

Now after we have explained what types of diagrams we will present, let's start presenting some UML diagrams of our system.

3 Use Case Diagram:

To model a system, the most important aspect is to capture the dynamic behavior. Dynamic behavior means the behavior of the system when it is running/operating.[9]

Only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML, there are five diagrams available to model the dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature, there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. Use case diagrams consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

Hence to model the entire system, a number of use case diagrams are used.

3.1 Purpose of Use Case Diagrams :

The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and Statechart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

When the initial task is complete, use case diagrams are modelled to present the outside view . In brief, the purposes of use case diagrams can be said to be as follows :

- Used to gather the requirements of a system.
- Used to get an outside view of a system.
- Identify the external and internal factors influencing the system.
- Show the interaction among the requirements are actors.

3.2 Use Case Diagram Representation:

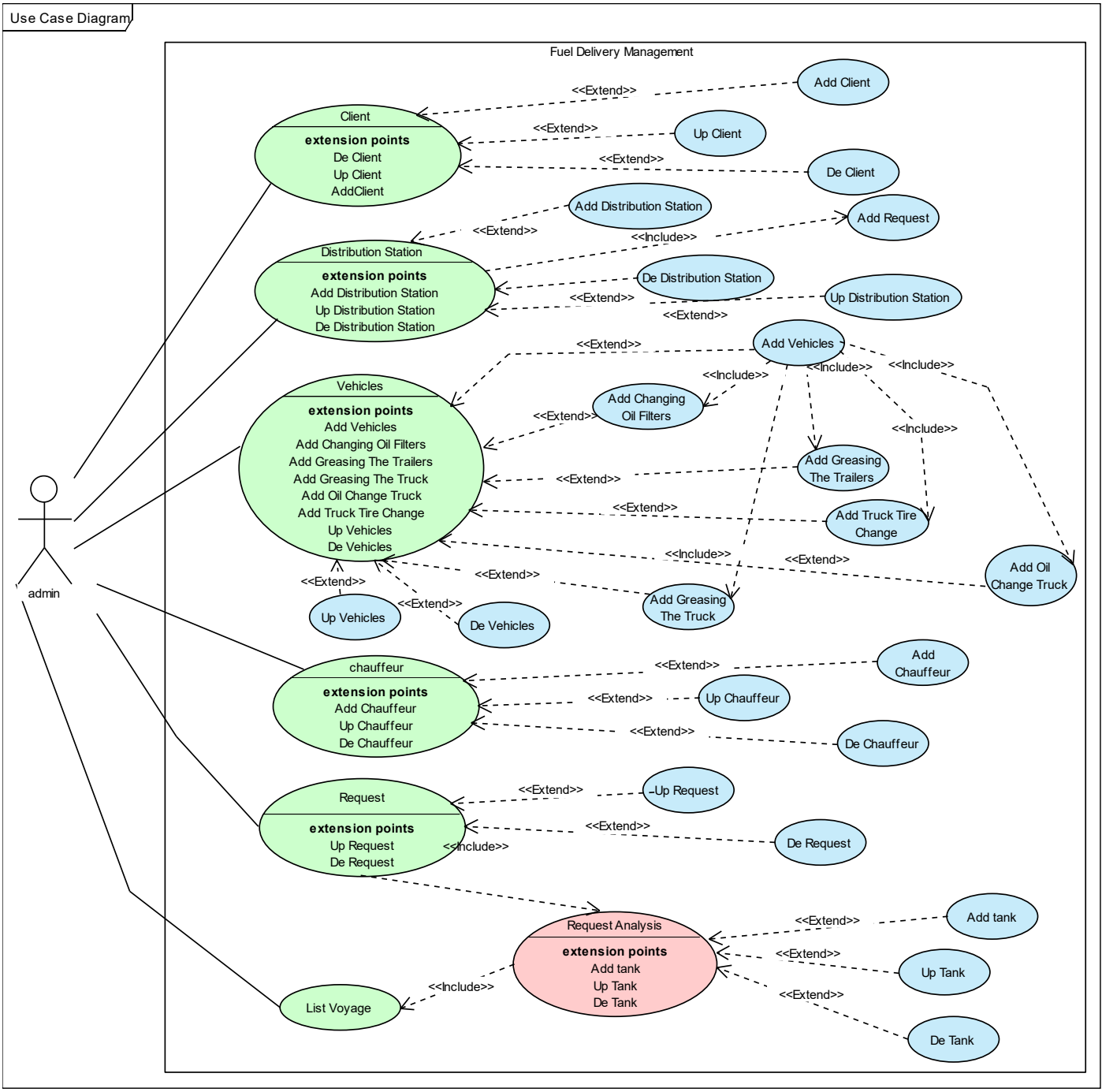


Figure 3.1: Use Case Diagram Representation.

4 Class Diagram:

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.[10]

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of objectoriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

4.1 Purpose of Class Diagrams :

The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community.

The purpose of the class diagram can be summarized as −

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

4.2 Class Diagram Representation:

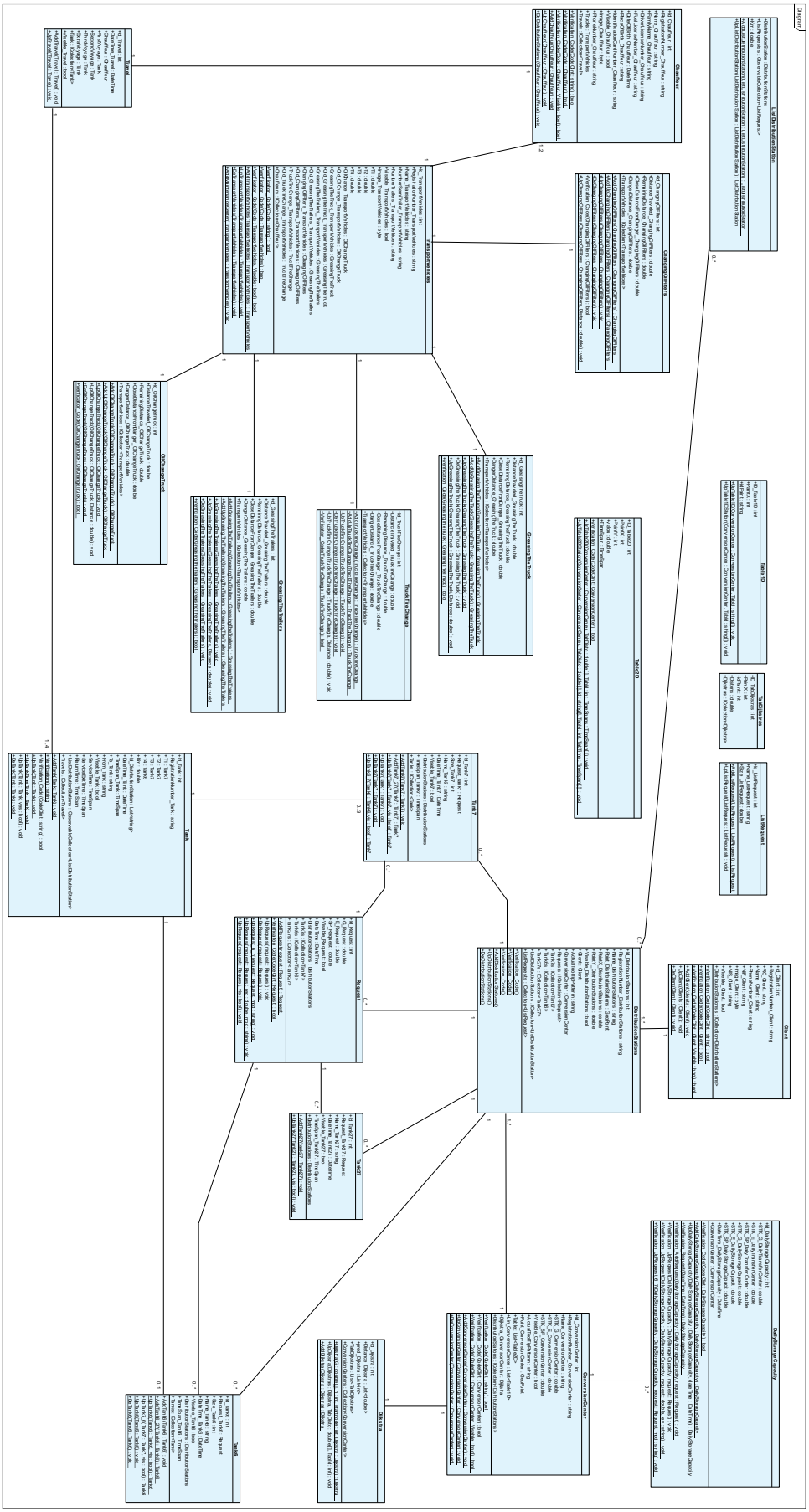


Figure 3.2: Class Diagram Representation.

5 Interaction Diagram:

From the term Interaction, it is clear that the diagram is used to describe some type of interactions among the different elements in the model. This interaction is a part of dynamic behavior of the system.[10]

This interactive behavior is represented in UML by two diagrams known as **Sequence diagram** and **Collaboration diagram**. The basic purpose of both the diagrams are similar.

Sequence diagram emphasizes on time sequence of messages and collaboration diagram emphasizes on the structural organization of the objects that send and receive messages.

5.1 Purpose of Interaction Diagrams :

The purpose of interaction diagrams is to visualize the interactive behavior of the system. Visualizing the interaction is a difficult task. Hence, the solution is to use different types of models to capture the different aspects of the interaction.

Sequence and collaboration diagrams are used to capture the dynamic nature but from a different angle.

The purpose of interaction diagram is −

- To capture the dynamic behaviour of a system.
- To describe the message flow in the system.
- To describe the structural organization of the objects.
- To describe the interaction among objects.

Here we will chose sequence diagram to represent our system.

5.2 Sequence Diagram Representation:

The sequence diagram emphasizes on time sequence of messages between the different parts of the running system, in this system we have two parts administration part and user part , so to model the entire system we need to present the two activity diagrams.

5.2.1 Client Sequence Diagram Representation :

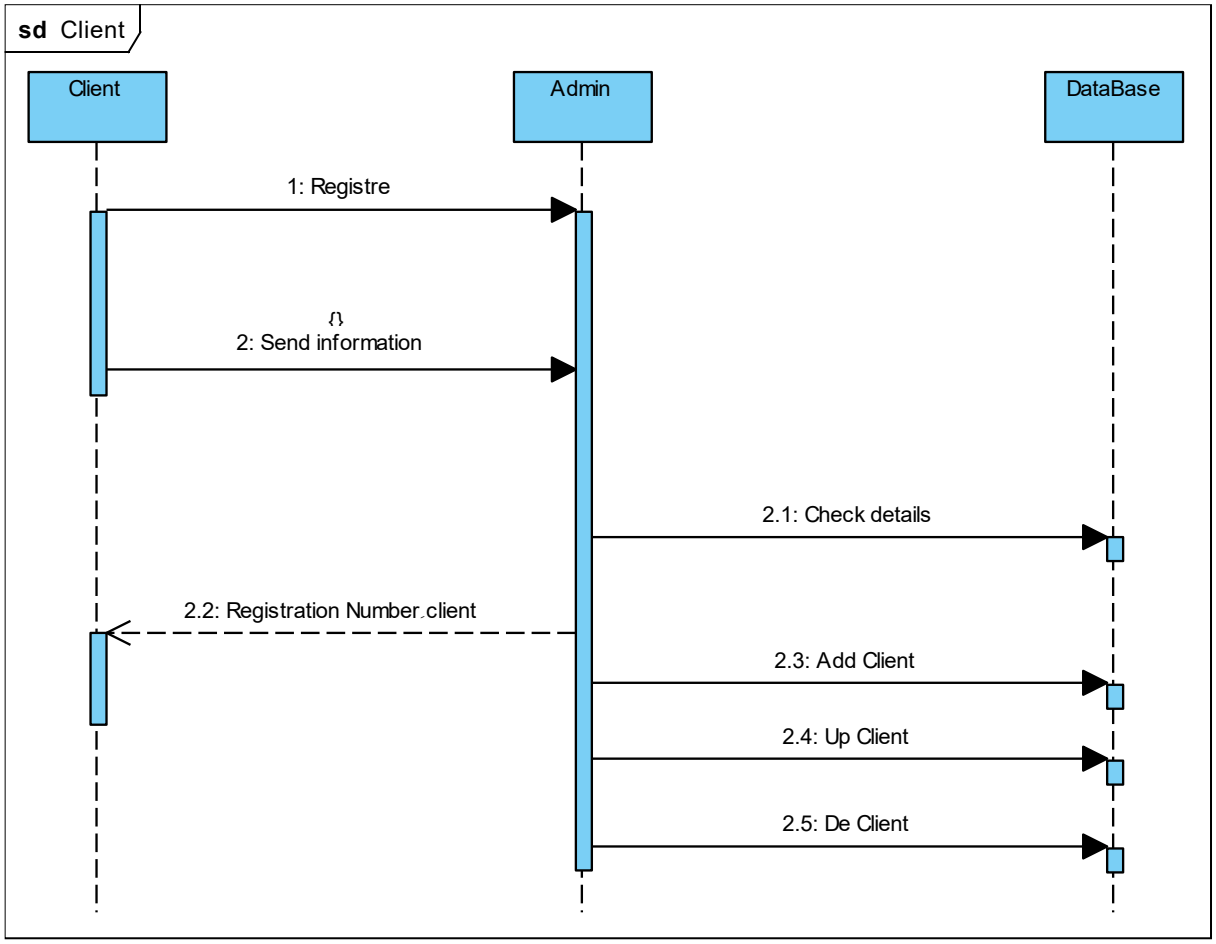


Figure 3.3: Client Sequence Diagram Representation :

5.2.2 Chauffeur Sequence Diagram Representation:

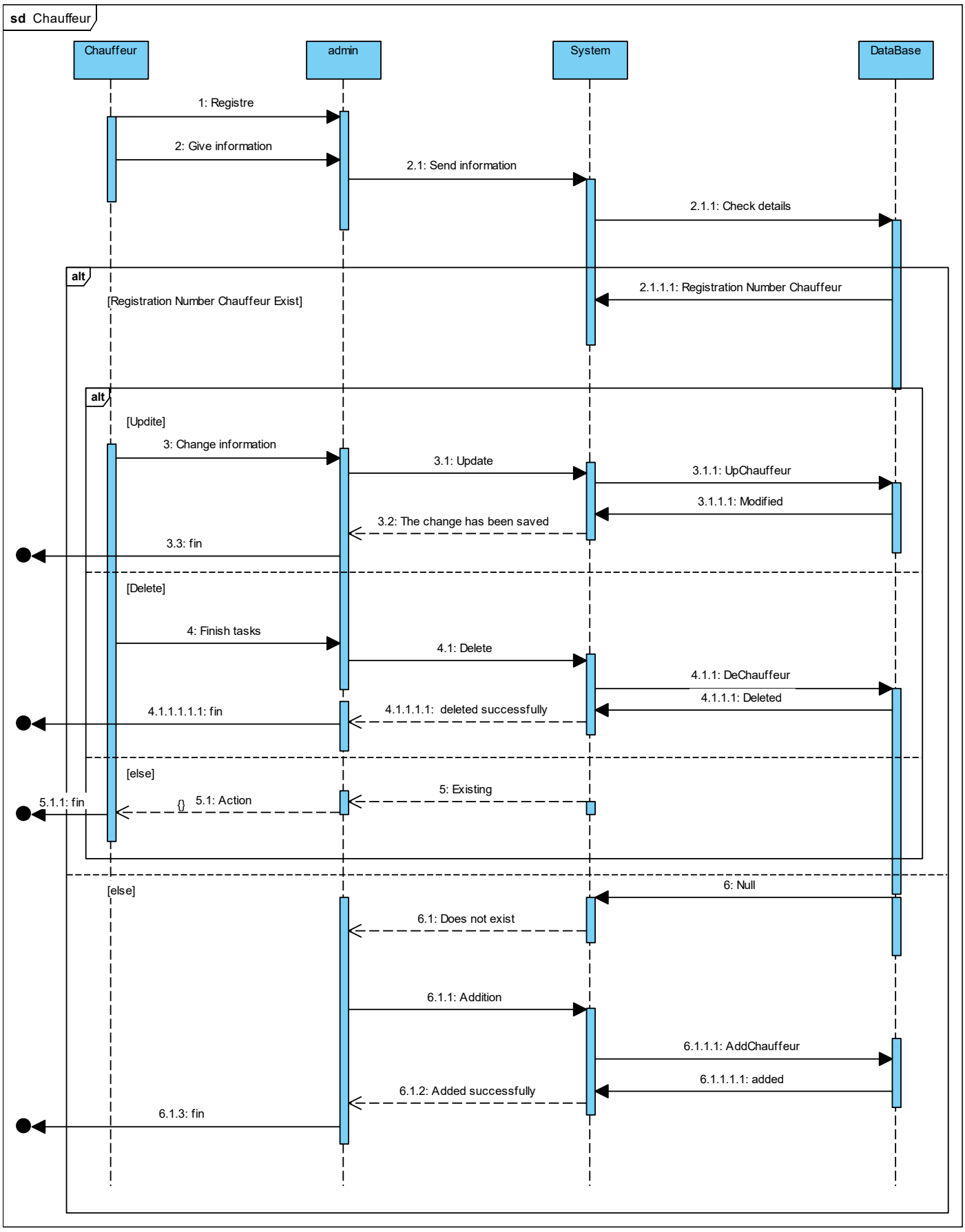


Figure 3.4: Chauffeur Sequence Diagram Representation :

6 Activity Diagrams :

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.[10]

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc

6.1 Purpose of Activity Diagrams :

The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

The purpose of an activity diagram can be described as −

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

6.2 Distribution Stations Activity Diagrams Representation :

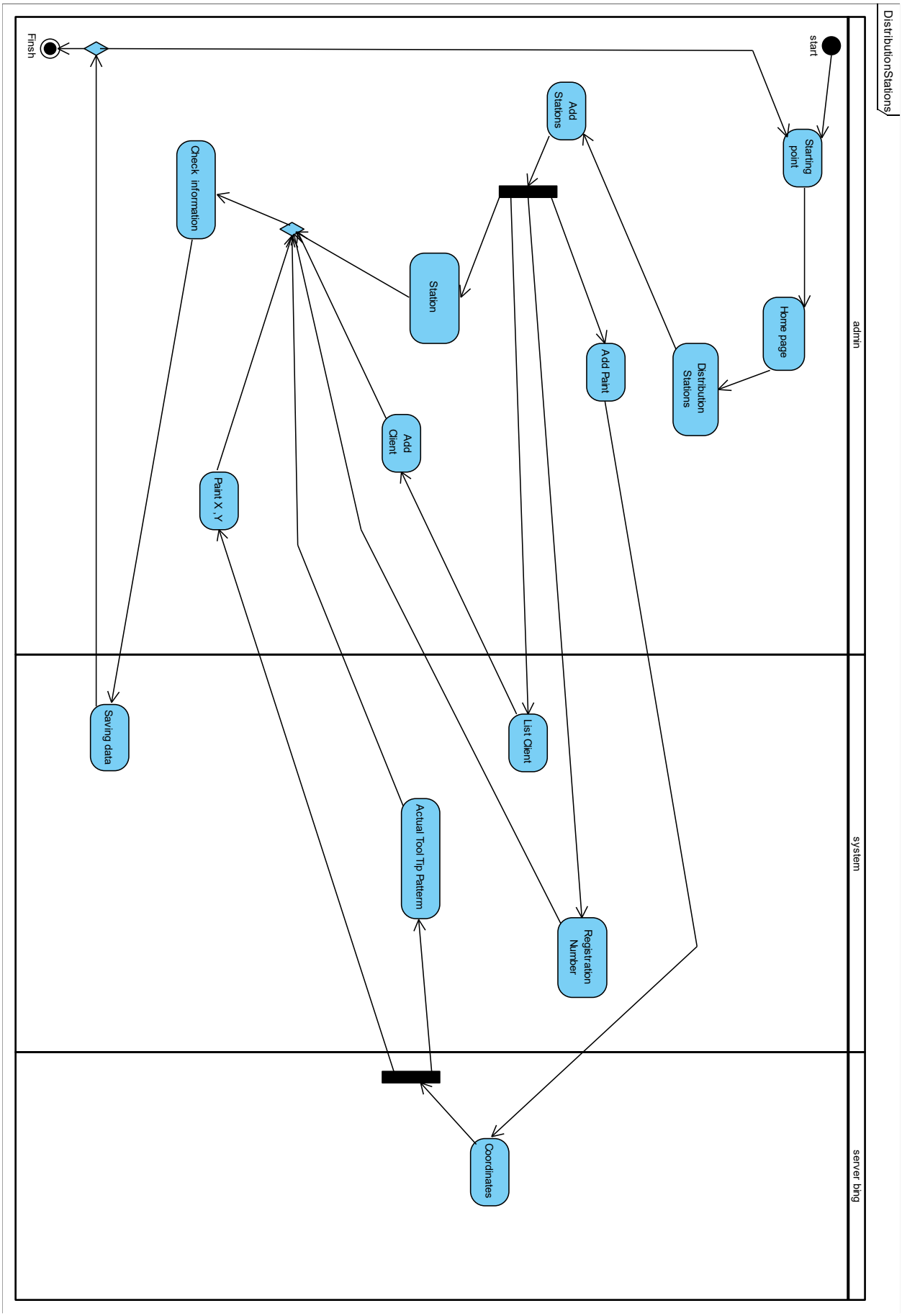


Figure 3.5: Distribution Stations Activity Diagrams Representation :

6.3 Analyze Orders Activity Diagrams Representation :

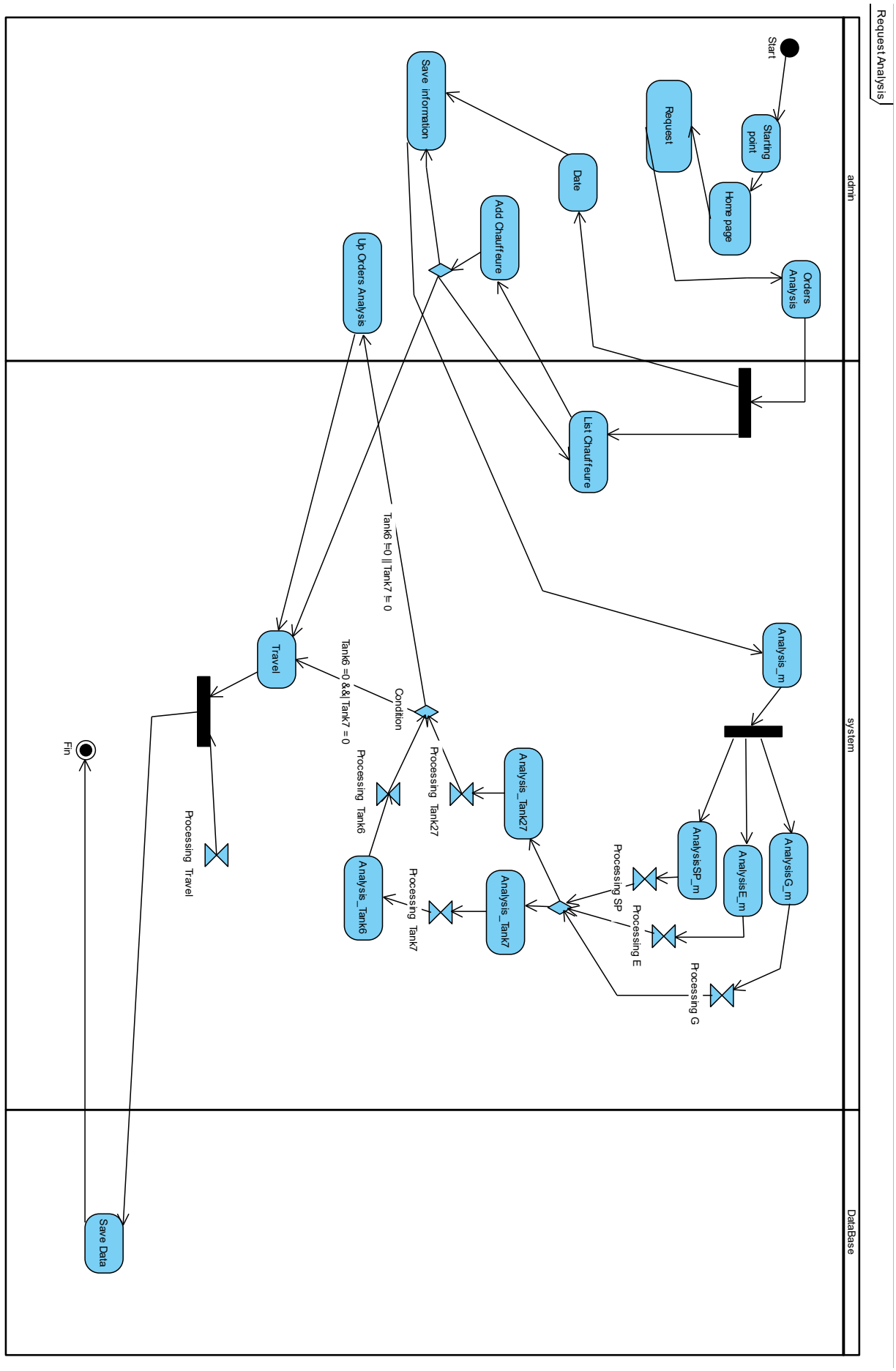


Figure 3.6: Analyze Orders Activity Diagrams Representation :

Chapter IV : Implementations

1 Introduction:

This chapter exposes the technical details associated to our system, the development environment, choice of programing languages and the database-management systems test and maintenance of our system. Besides, this chapter contains a set of interfaces of the main functionalities provided by our solution with their explications.

2 Technical environment and tools:

2.1 Microsoft Visual Studio :

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that expand the functionality at almost every level—including adding support for source control systems (like Subversion and Git) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Azure DevOps client: Team Explorer).

Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C,[11] C++, C++/CLI, Visual Basic .NET, C#, F#,[12] JavaScript, TypeScript, XML, XSLT, HTML, and CSS. Support for other languages such as Python, [13] Ruby, Node.js, and M among others is available via plug-ins. Java (and J#) were supported in the past.

The most basic edition of Visual Studio, the Community edition, is available free of charge. The slogan for Visual Studio Community edition is "Free, fully-featured IDE for students, open-source and individual developers".

As of March 2021 the current production-ready Visual Studio version was 2019, with older versions such as 2012 and 2013 on Extended Support, and 2015 and 2017 on Mainstream Support.[14]

2.2 Desktop .Net Framework:

To develop the application we have chosen **.NET Framework** as Desktop application frame work.

2.2.1 NET Framework :

The **.NET Framework** (pronounced as "*dot net*") is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large class library called Framework Class Library (FCL) and provides language interoperability (each language can use code written in other languages) across several programming languages. Programs written for .NET Framework execute in a software environment (in contrast to a hardware environment) named the Common Language Runtime (CLR). The CLR is an application virtual machine that provides services such as security, memory management, and exception handling. As such, computer code written using .NET Framework is called "managed code". FCL and CLR together constitute the .NET Framework.

FCL provides the user interface, data access, database connectivity, cryptography, web application development, numeric algorithms, and network communications. Programmers produce software by combining their source code with .NET Framework and other libraries. The framework is intended to be used by most new applications created for the Windows platform. Microsoft also produces an integrated development environment for .NET software called Visual Studio.[15]

A) Architecture :

A .1 Common Language Infrastructure:

Common Language Infrastructure (CLI) provides a language-neutral platform for application development and execution. By implementing the core aspects of .NET Framework within the scope of CLI, these functions will not be tied to one language but will be available across the many languages supported by the framework.

A .2 Common Language Runtime:

NET Framework includes the Common Language Runtime (CLR). It serves as the execution engine of .NET Framework and offers many services such as memory management, type safety, exception handling, garbage collection, security and thread management. All programs written for .NET Framework are executed by the CLR.

2.2.2 C Sharp :

C# (pronounced see sharp, like the musical note C♯, but written with the number sign) is a general-purpose, multi-paradigm programming language encompassing static typing, strong

typing, lexically scoped, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines.[16]

C# was developed around 2000 by Microsoft as part of its .NET initiative and later approved as an international standard by Ecma (ECMA-334) in 2002 and ISO (ISO/IEC 23270) in 2003. It was designed by Anders Hejlsberg, and its development team is currently led by Mads Torgersen, being one of the programming languages designed for the Common Language Infrastructure (CLI). The most recent version is 9.0, which was released in 2020 in .NET 5.0 and included in Visual Studio 2019 version 16.8.

2.3 Windows Presentation Foundation :

Windows Presentation Foundation (**WPF**) is a free and open-source graphical subsystem (similar to WinForms) originally developed by Microsoft for rendering user interfaces in Windows-based applications. WPF, previously known as "Avalon", was initially released as part of .NET Framework 3.0 in 2006. WPF uses DirectX and attempts to provide a consistent programming model for building applications. It separates the user interface from business logic, and resembles similar XML-oriented object models, such as those implemented in XUL and SVG [17]

A) XAML :

Extensible Application Markup Language (XAML) ,Following the success of markup languages for web development, WPF introduces eXtensible Application Markup Language (XAML; /ˈzæməl/), which is based on XML. XAML is designed as a more efficient method of developing application user interfaces. [18] The specific advantage that XAML brings to WPF is that XAML is a completely declarative language, allowing the developer (or designer) to describe the behavior and integration of components without the use of procedural programming. Although it is rare that an entire application will be built completely in XAML, the introduction of XAML allows application designers to more effectively contribute to the application development cycle.
Using XAML to develop user interfaces also allows for separation of model and view, which is considered a good architectural principle. In XAML, elements and attributes map to classes and properties in the underlying APIs.

As in web development, both layouts and specific themes are well suited to markup, but XAML is not required for either. Indeed, all elements of WPF may be coded in a .NET language (C#, VB.NET). The XAML code can ultimately be compiled into a managed assembly in the same way all .NET languages are.

2.4 DevExpress :

Developer Express Inc. (DevExpress) is a software development company founded in 1998 with headquarters in Glendale, California. DevExpress initially started producing UI Controls for Borland Delphi/C++Builder and ActiveX Controls for Microsoft Visual Studio. Presently, DevExpress has products targeting developers that use Delphi/C++Builder, Visual Studio and HTML5/JavaScript technologies.[19]

2.5 SQL Server 2019 :

2.5.1 Presentation:

Microsoft SQL Server is a database management system (DBMS) in SQL language incorporating among other things a BDMS (relational DBMS) developed and marketed by Microsoft.

It runs on Windows and Linux OS (since March 2016), but it is possible to launch it on MacOs via Docker, because there is a download version on the Microsoft website.

SQL Server Express is a free entry-level version of the database, ideal for learning, as well as creating desktop applications and small servers up to 10GB of data. [20]

2.5.2 Strengths of SQL Server:

- SQL Server Management Studio to easily manage databases (creation of tables, queries, etc..)
- Integration with other Microsoft products
- Very good performance in general on Windows
- Advanced Security Options
- Data and backup compression
- Free Express Version

2.6 Material Design

Material Design (codenamed Quantum Paper) [21] is a design language developed by Google in 2014. Expanding on the "cards" that debuted in Google Now, Material Design uses more grid-based layouts, responsive animations and transitions, padding, and depth effects such as lighting and shadows.

Google announced Material Design on June 25, 2014, at the 2014 Google I/O conference.

3 Test:

Load the data before entering the program

Figure 4.1: Splash Screen

3.1 Home :

It carries all the operations and shortcuts of the program

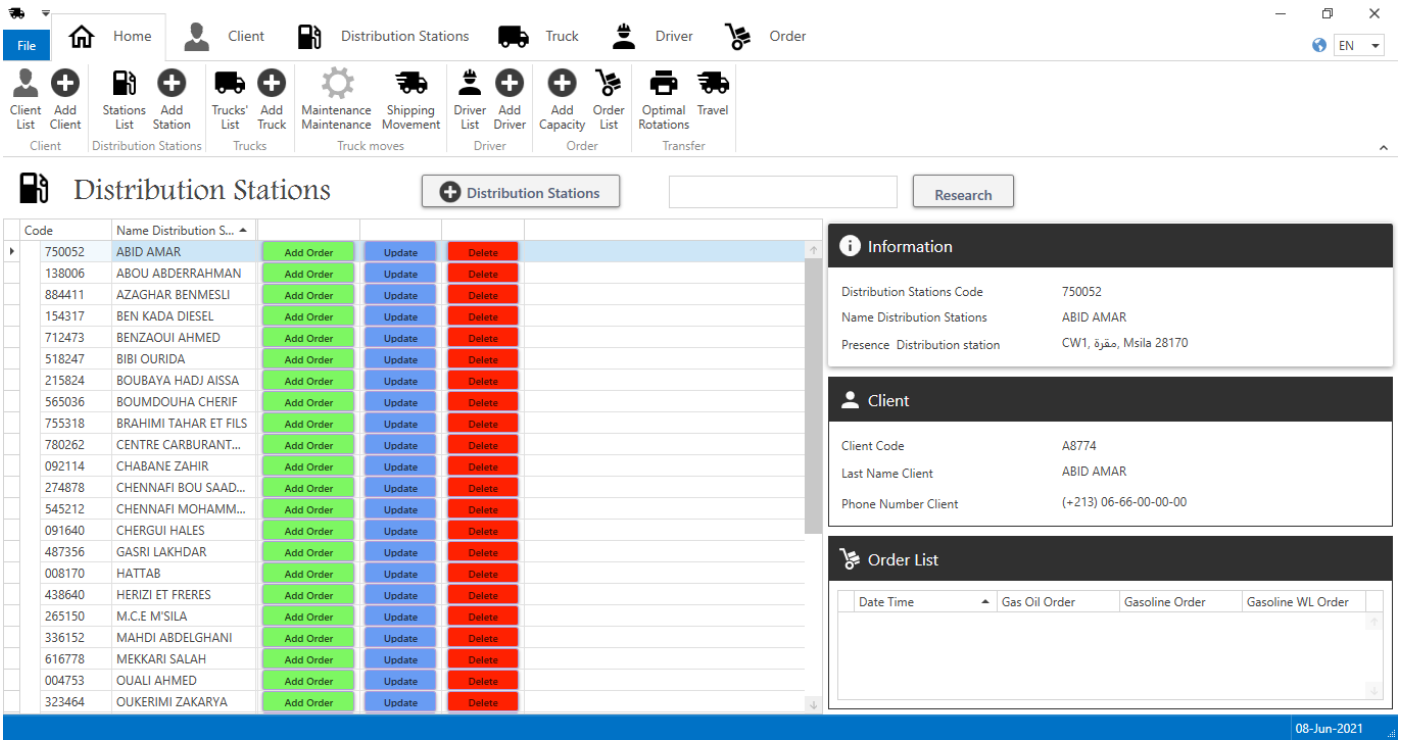


Figure 4.2: Homepage

3.2 Distribution Stations :

3.2.1 List Distribution Stations :

List of distribution stations and also add requests for each distribution station

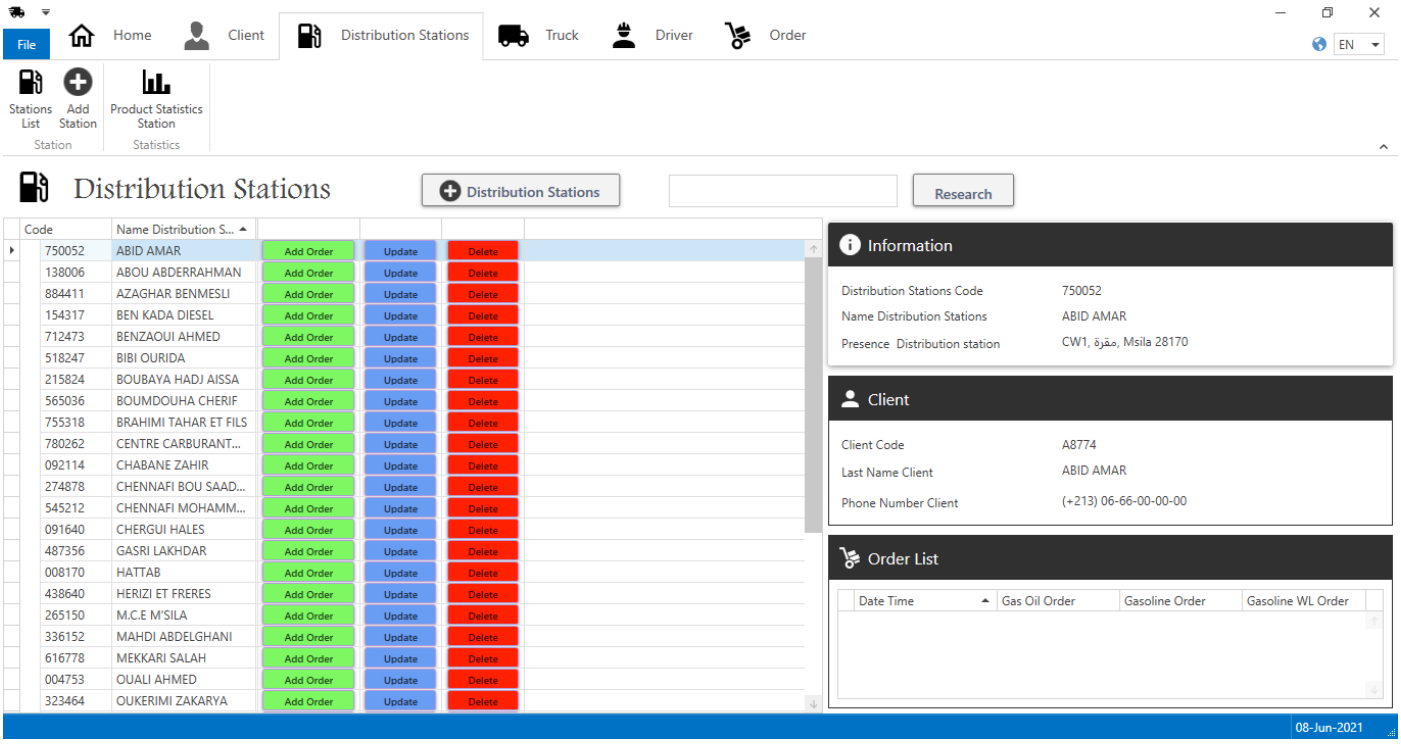


Figure 4.3: List Distribution Stations

3.2.2 In addition Distribution Stations :

Here we can add the distribution station by adding the client, location and other information.

A) Add Client :

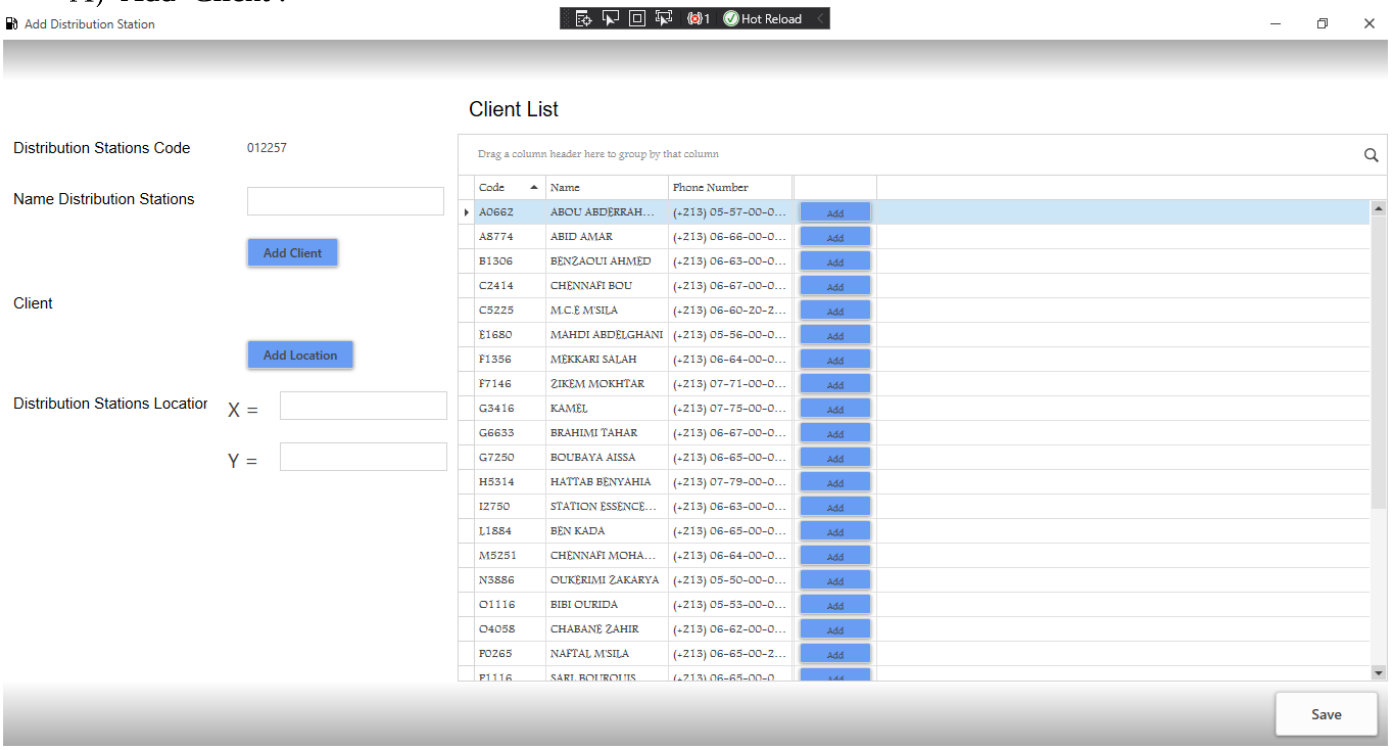


Figure 4.4: List Client on Distribution Stations

B) Add Location :

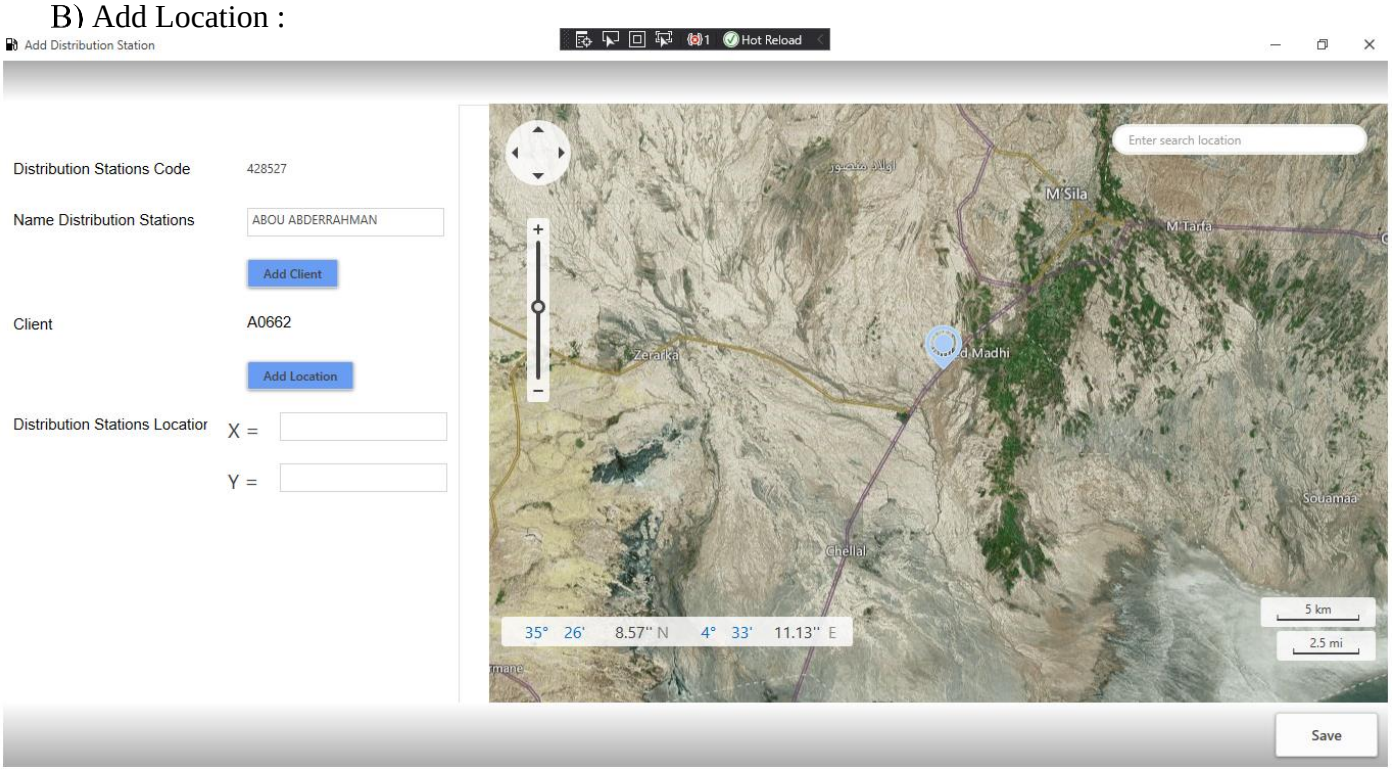


Figure 4.5: Add Location on Distribution Stations

3.2.3 in Updated Distribution Stations :

Here we can modify the distribution station's data.

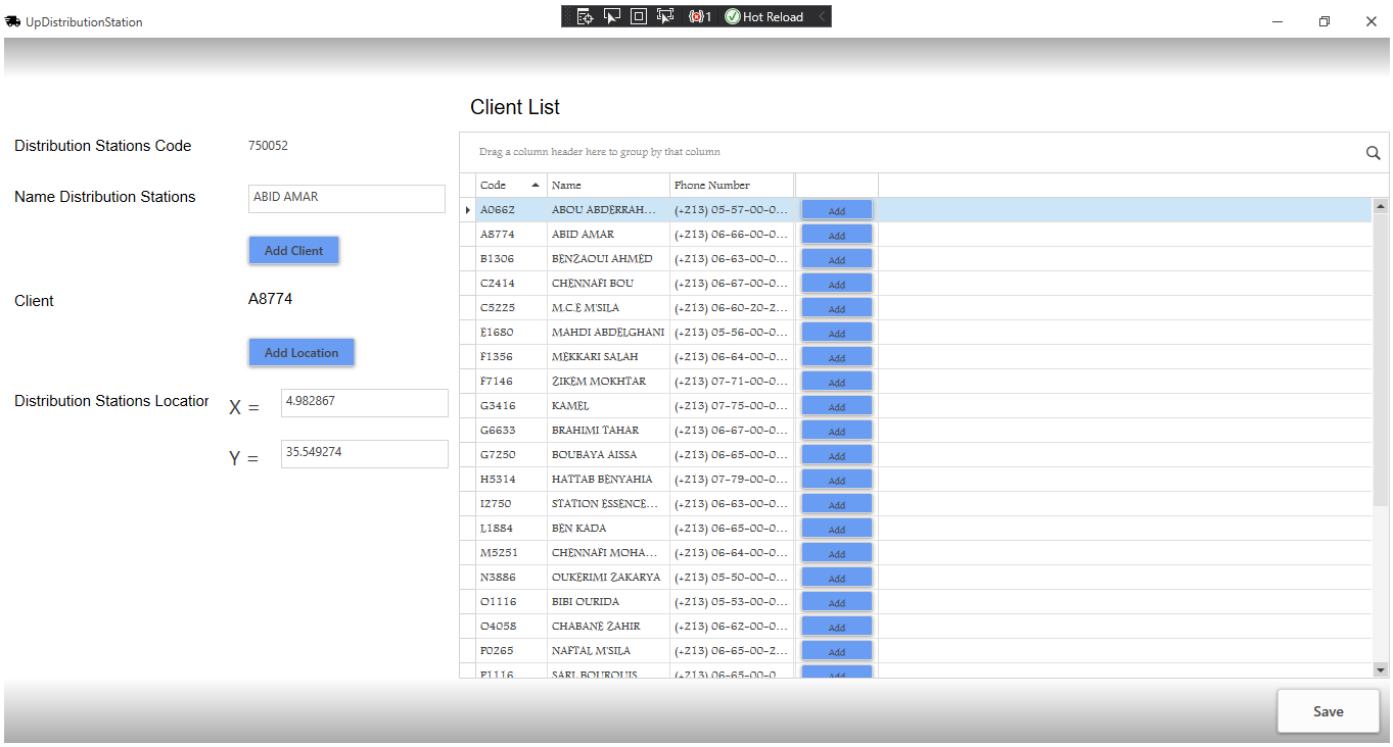


Figure 4.6: Modify Distribution Stations

3.2.4 Order Statistics:

View the statistics for the orders of the distribution station

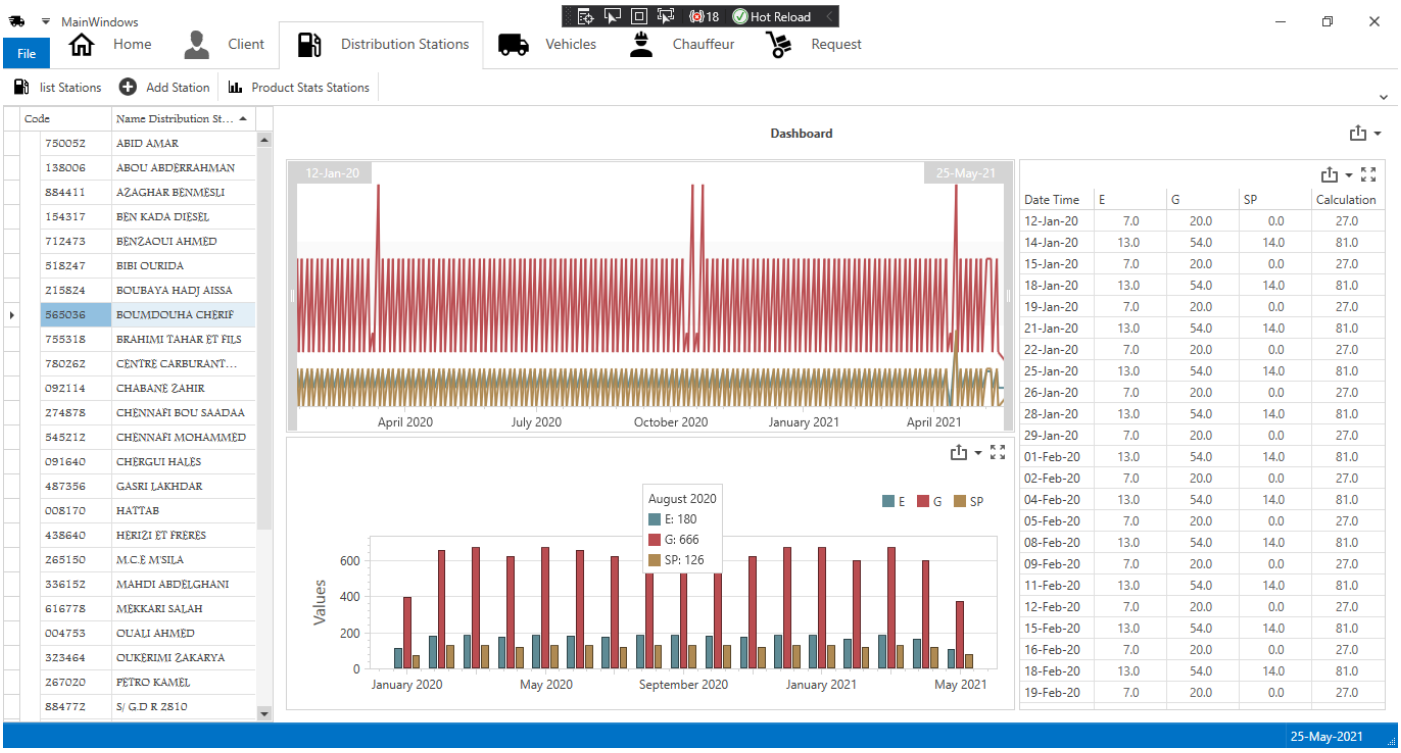


Figure 4.7: Distribution station order statistics

3.3 Client :

3.3.1 Client List :

View the list of clients and information about each client

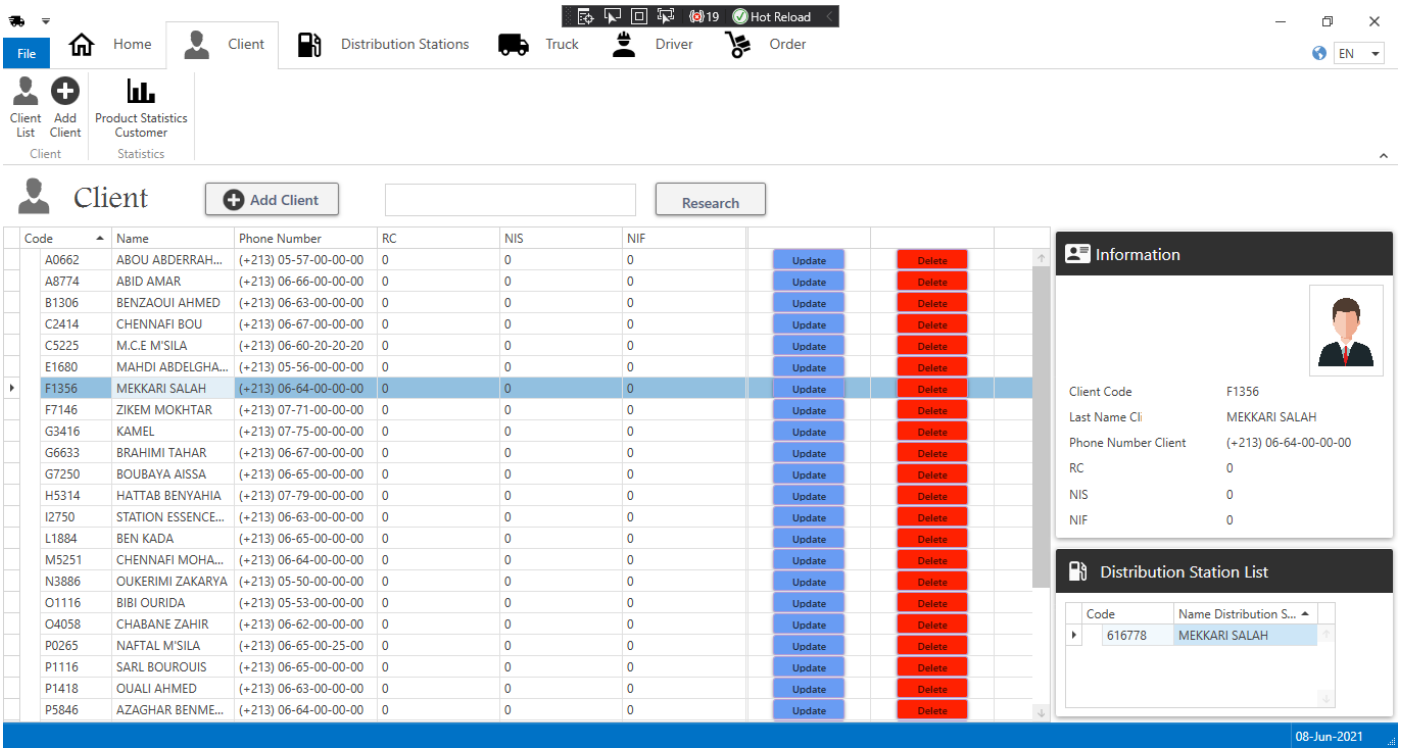


Figure 4.8: Distribution station order statistics

3.3.2 In addition client :

Add and save client information

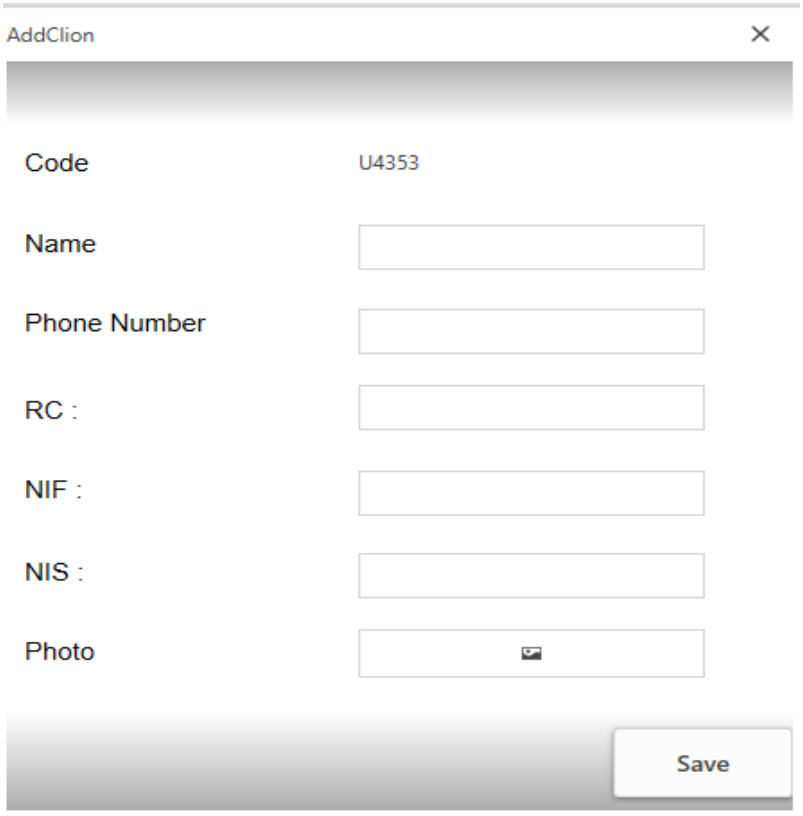


Figure 4.9: Add client information

3.3.3 in updated Client :

Modification and save of client information

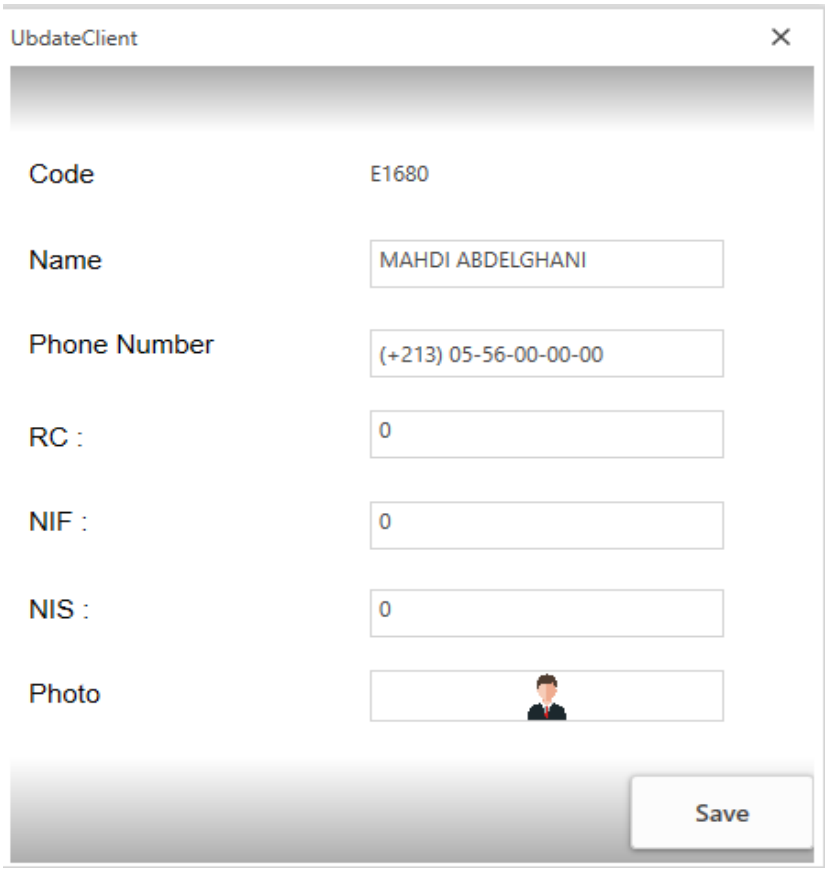


Figure 4.10: Modification client information

3.3.4 Order Statistics:

View the statistics for the orders of the client

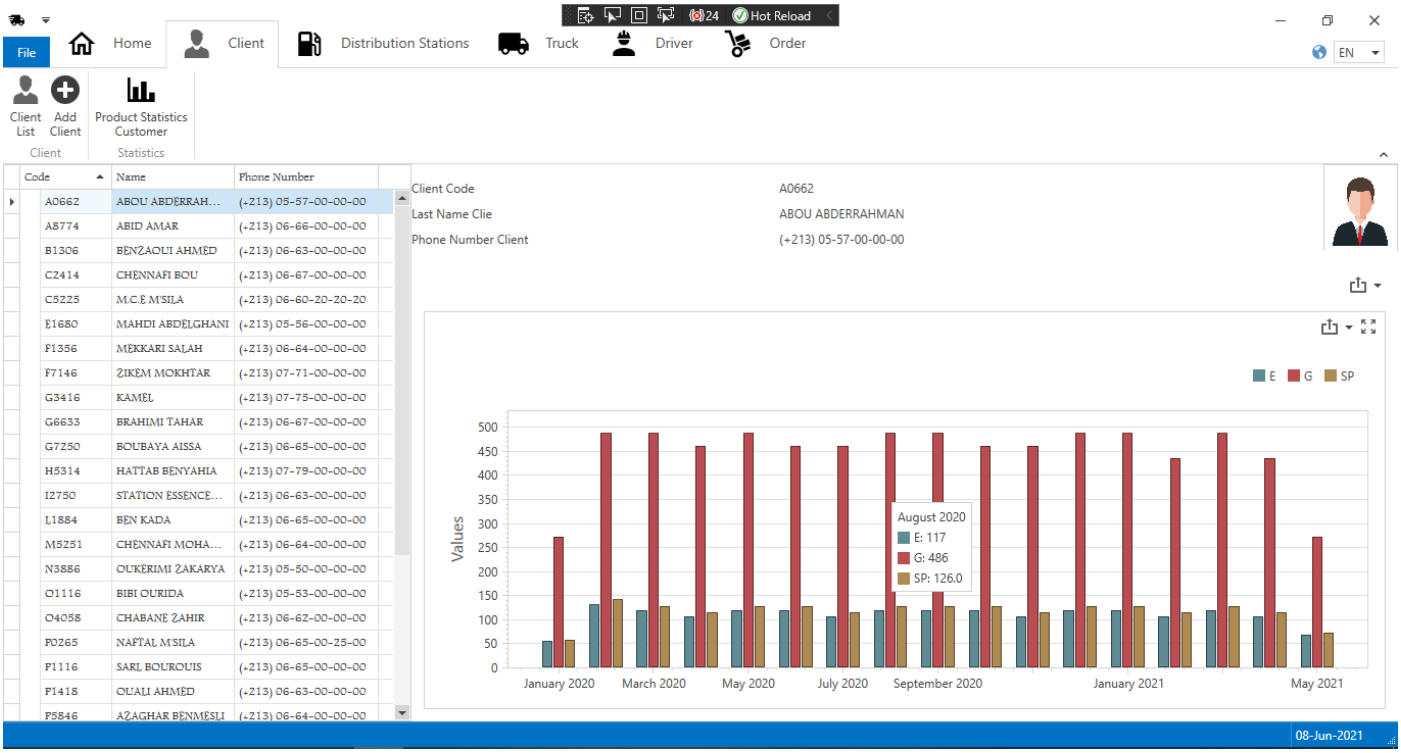


Figure 4.11: client order statistics

3.4 Trucks :

3.4.1 Trucks‘ List :

View all registered trucks and their data, and trailer

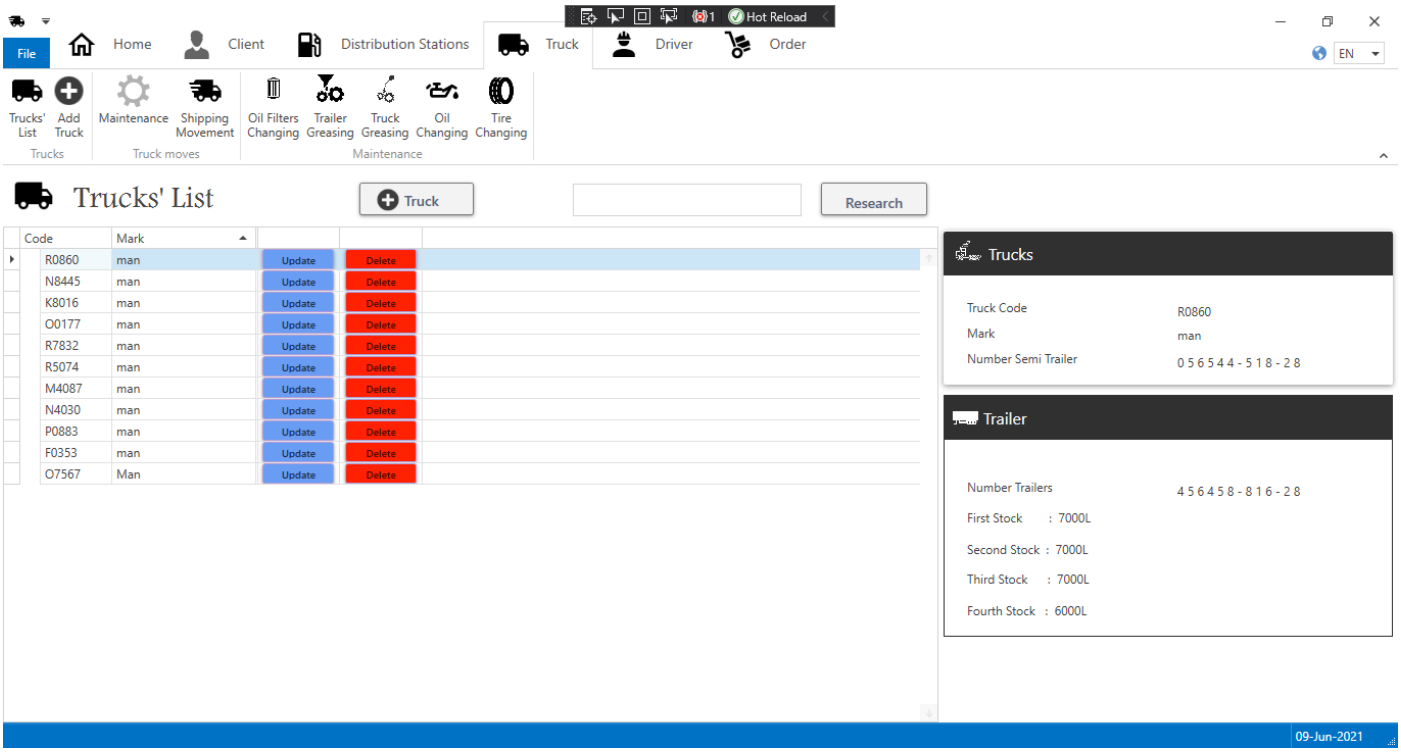


Figure 4.12: Trucks ‘list

3.4.2 In addition Trucks :

Add Information Truck and Trailer

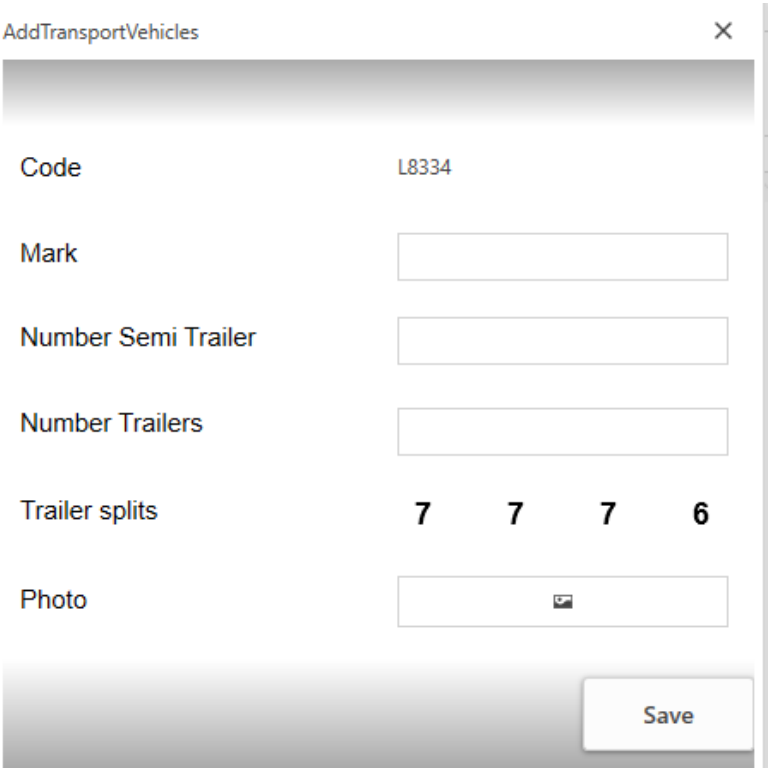


Figure 4.13: Add Truck and Trailer

3.4.3 in updated Trucks :

Modify Information Truck and Trailer

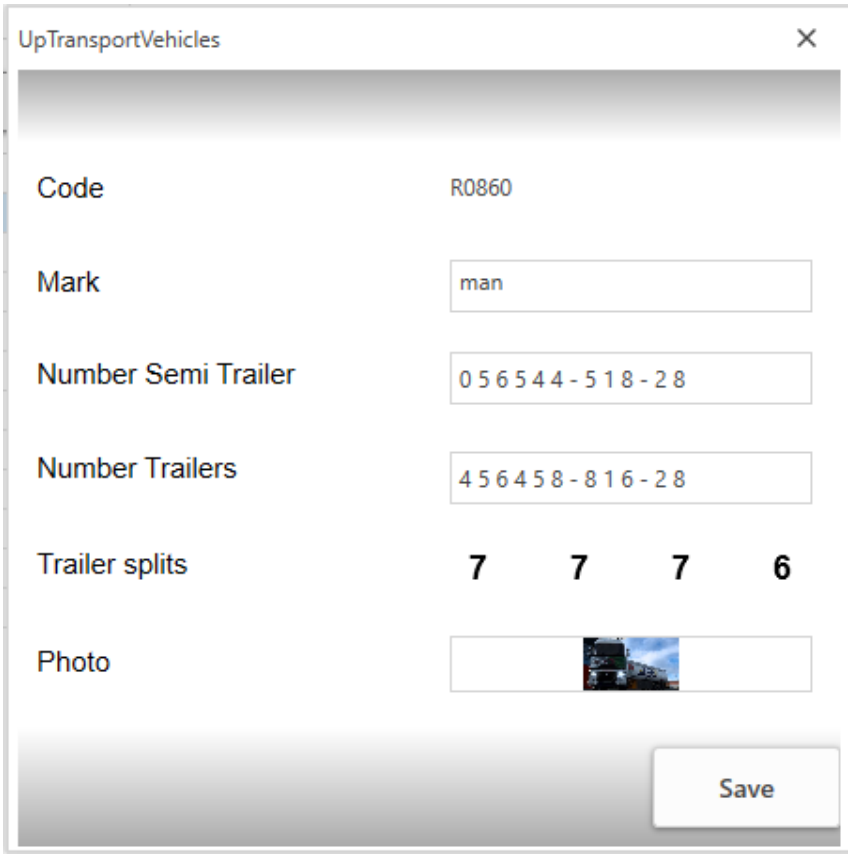


Figure 4.14: Modify Truck and Trailer

3.4.4 Maintenance :

List Control Truck repair

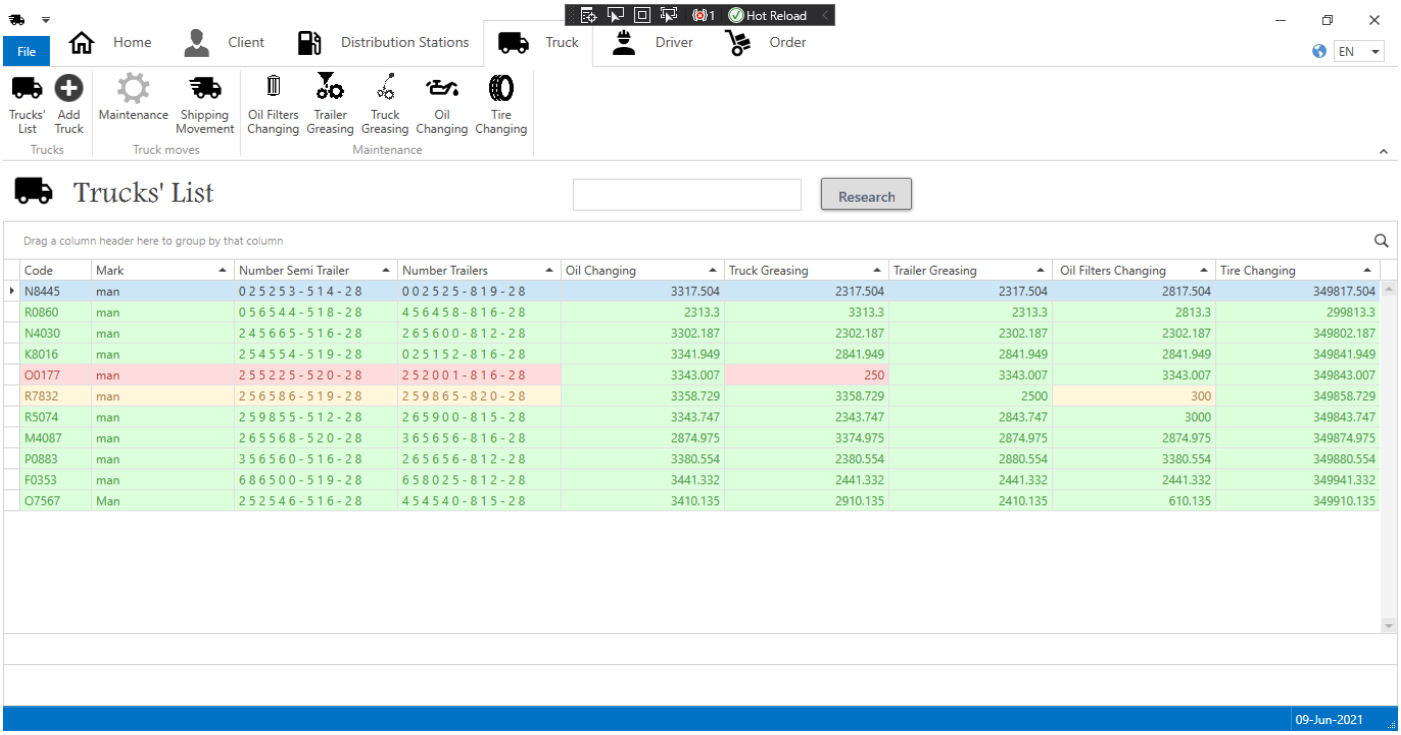


Figure 4.15: List Control maintenance truck

A) Oil Filter Changing :

List to control and changing oil filters for trucks

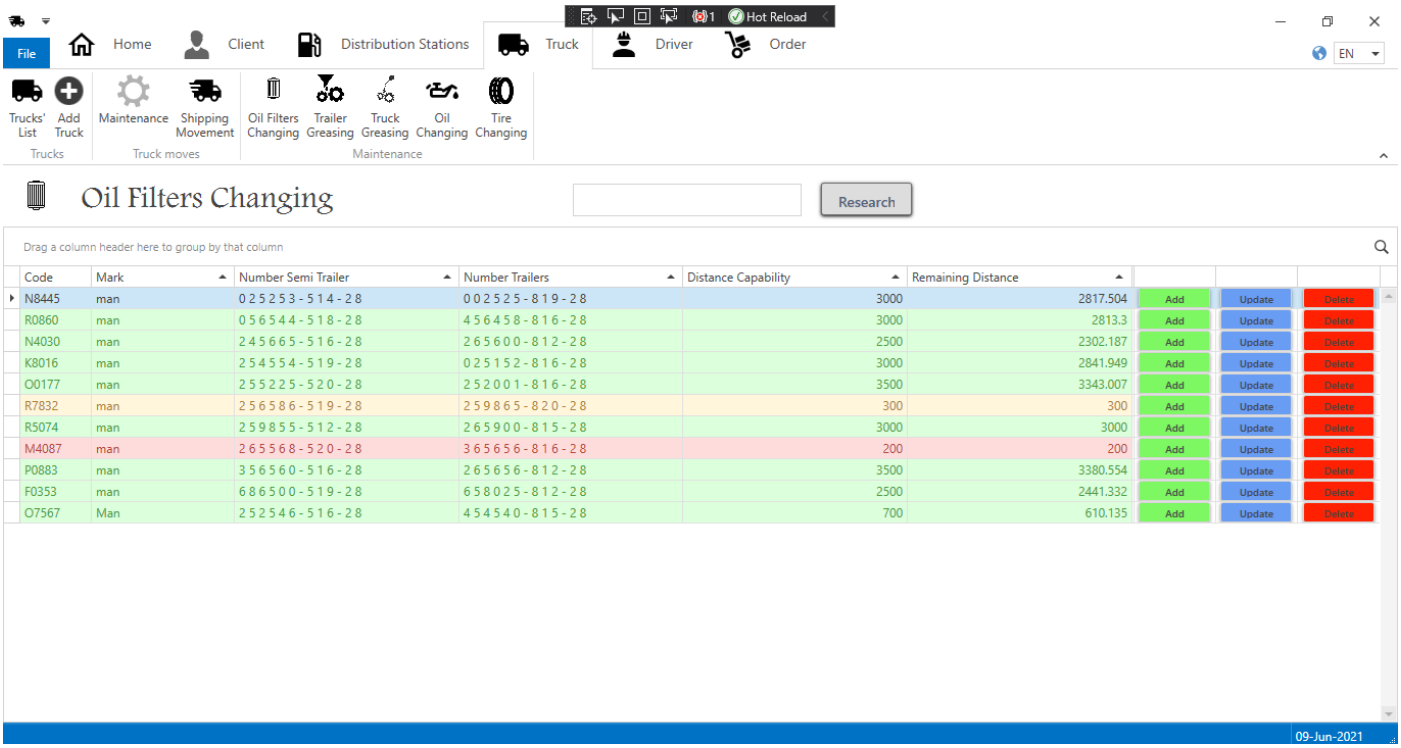


Figure 4.16: List Control oil Filter Changing truck

B) Trailer greasing :

List to control and add greasing trailer for trucks

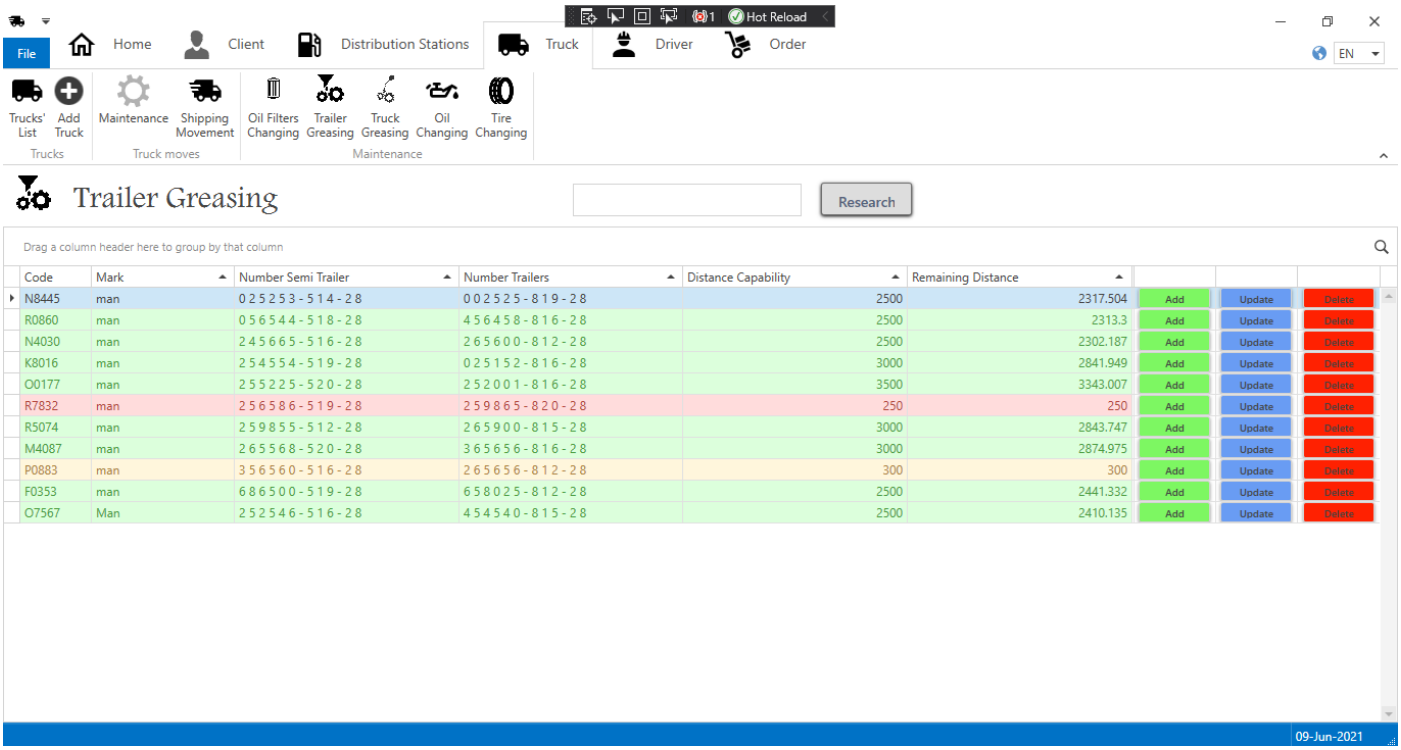


Figure 4.17: List Control trailer greasing truck

C) Truck greasing :

List to control and add greasing for trucks

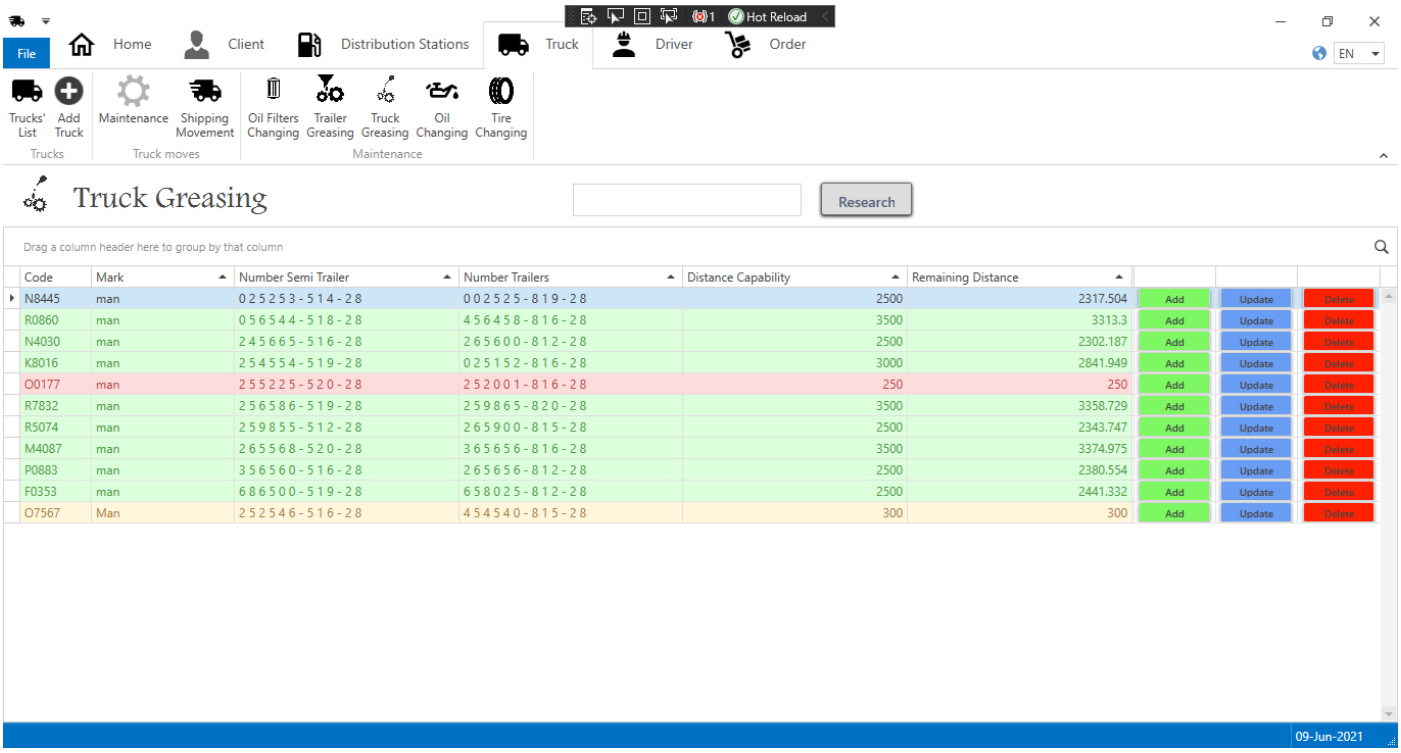


Figure 4.18: List Control greasing truck

D) Oil Changing :

List to control and changing oil for trucks

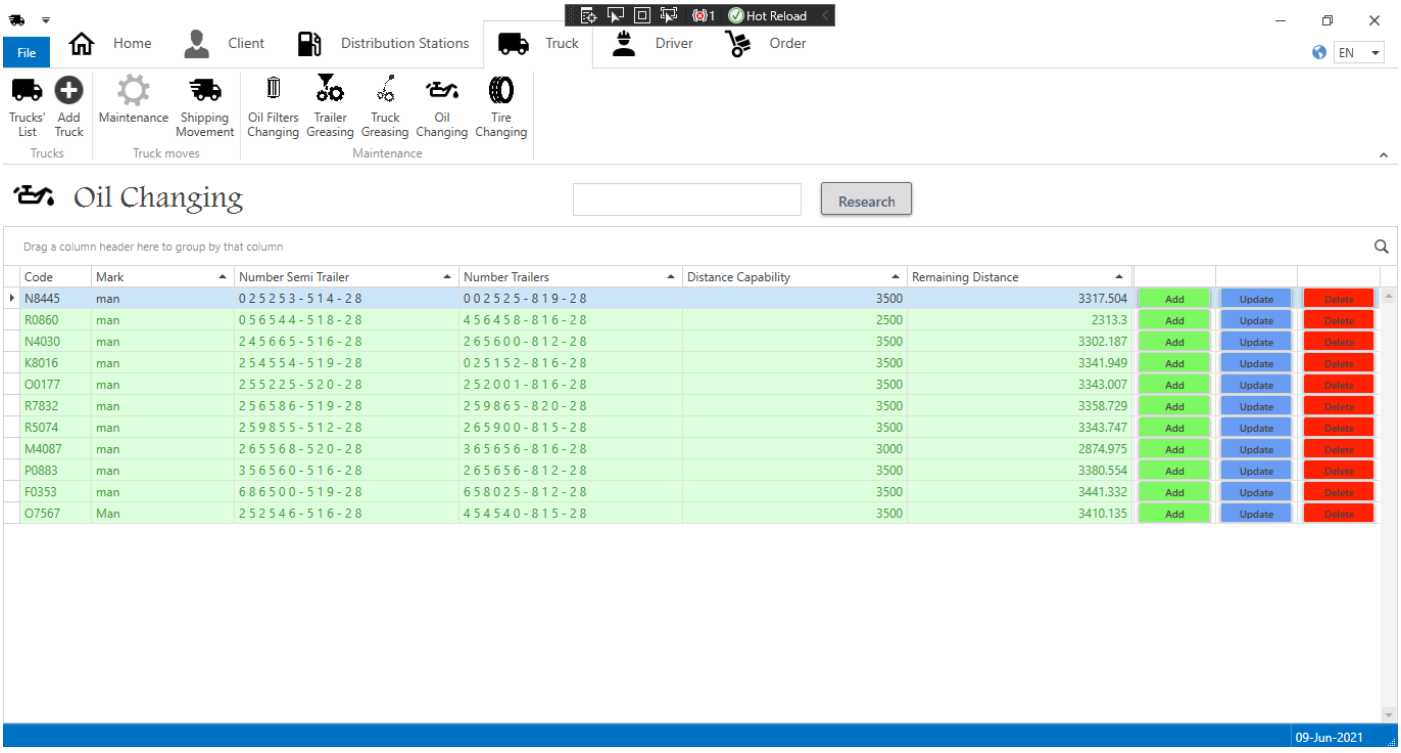


Figure 4.19: List Control oil changing truck

E) Tire Changing :

List to control and changing tire for trucks

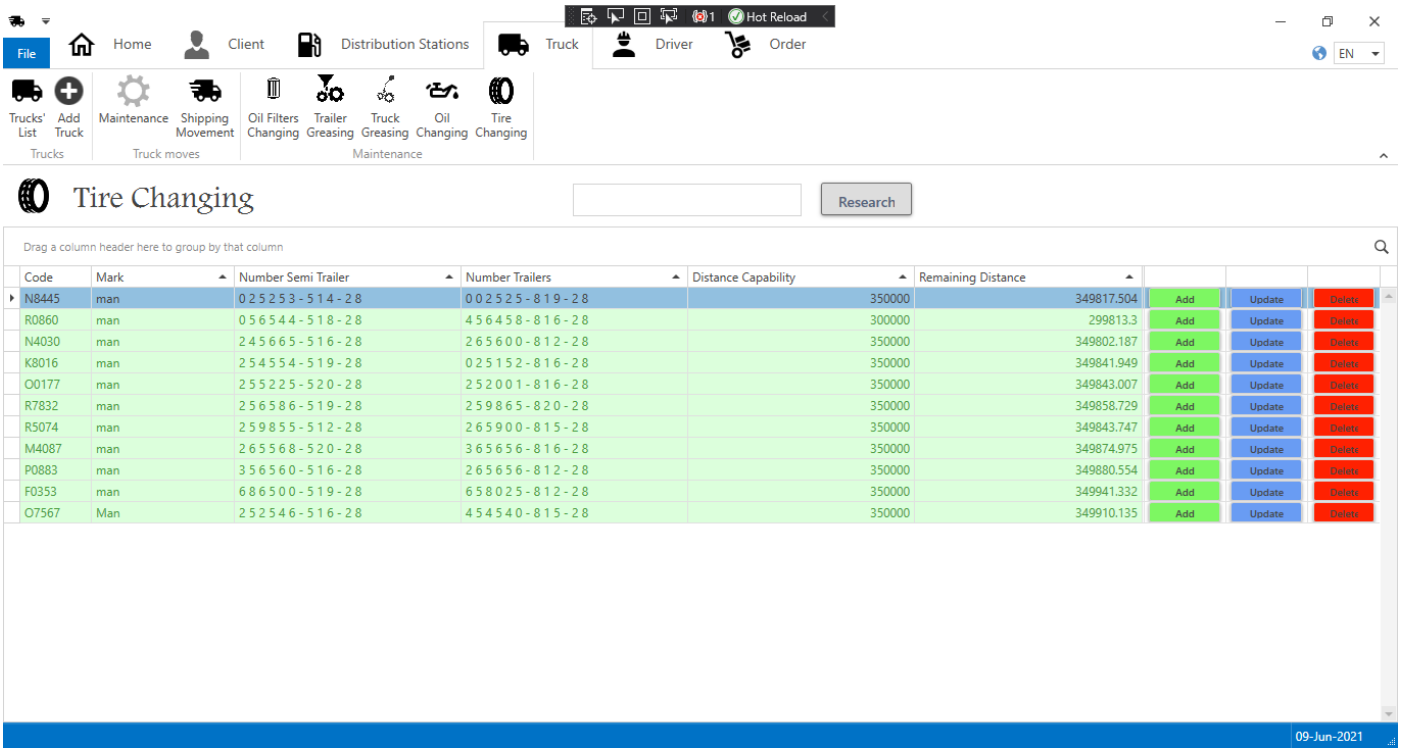


Figure 4.20: List Control tire changing truck

3.4.5 Shipping Movement:

List Control shipping movement

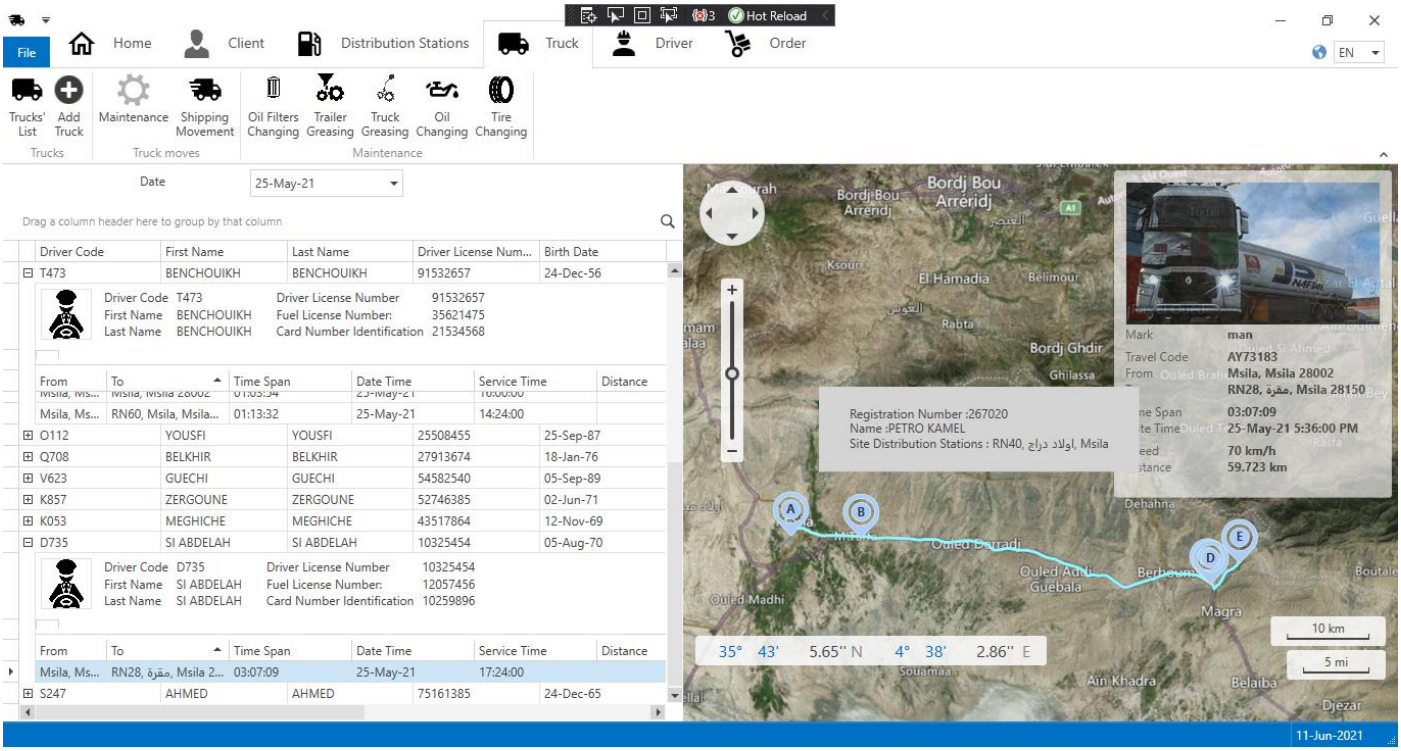


Figure 4.21: List Control shipping movement

3.5 Driver :

3.5.1 Driver List :

Driver list, driver information and associated truck

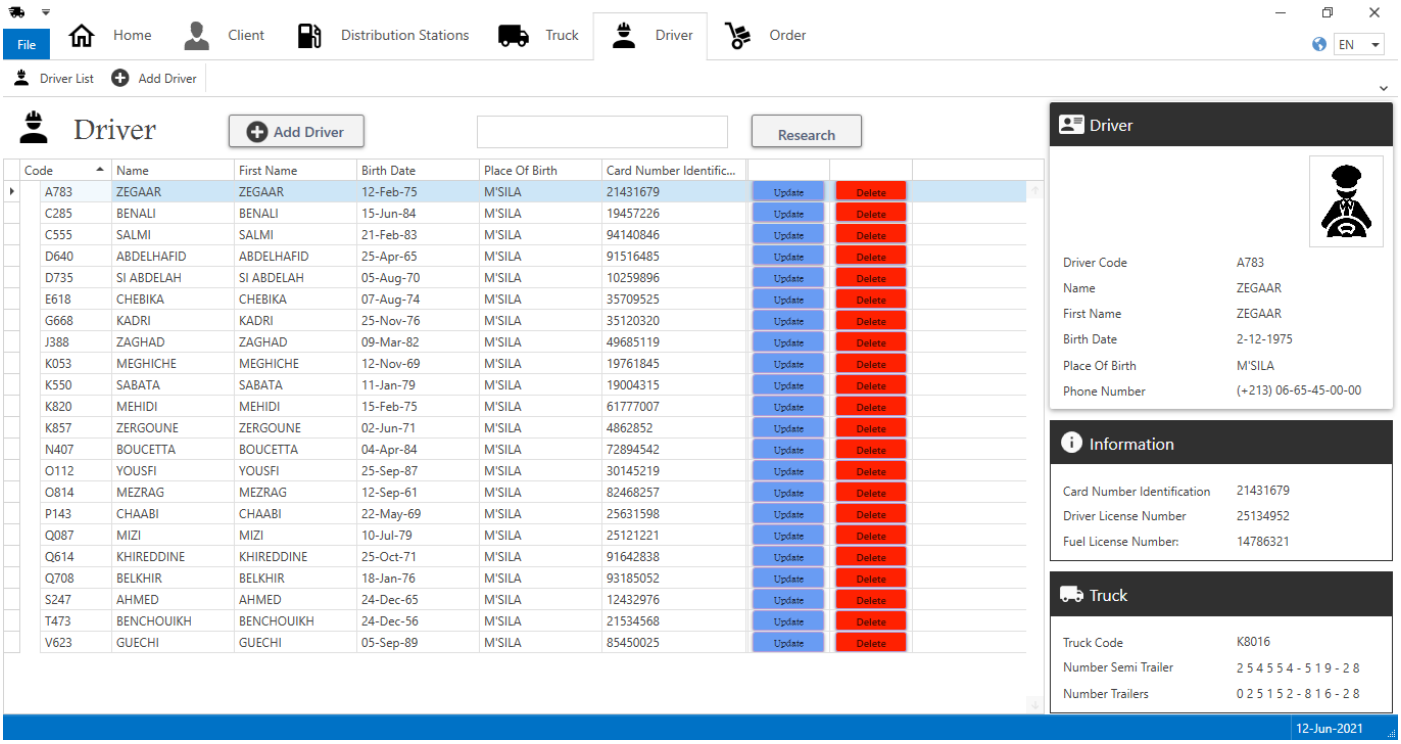


Figure 4.22: Driver List

3.5.2 in addition driver

Add driver and information

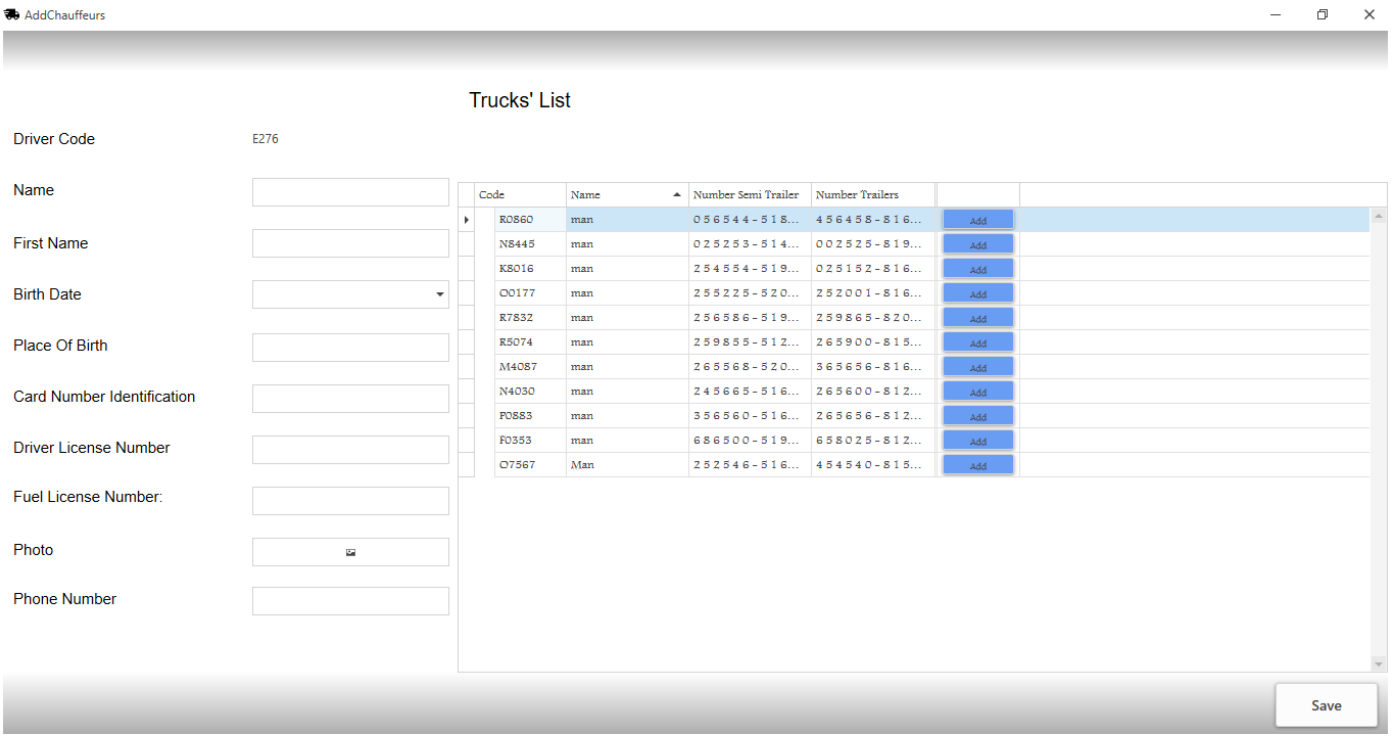


Figure 4.23: Add driver

3.5.3 in updated Driver:

Modification to driver information

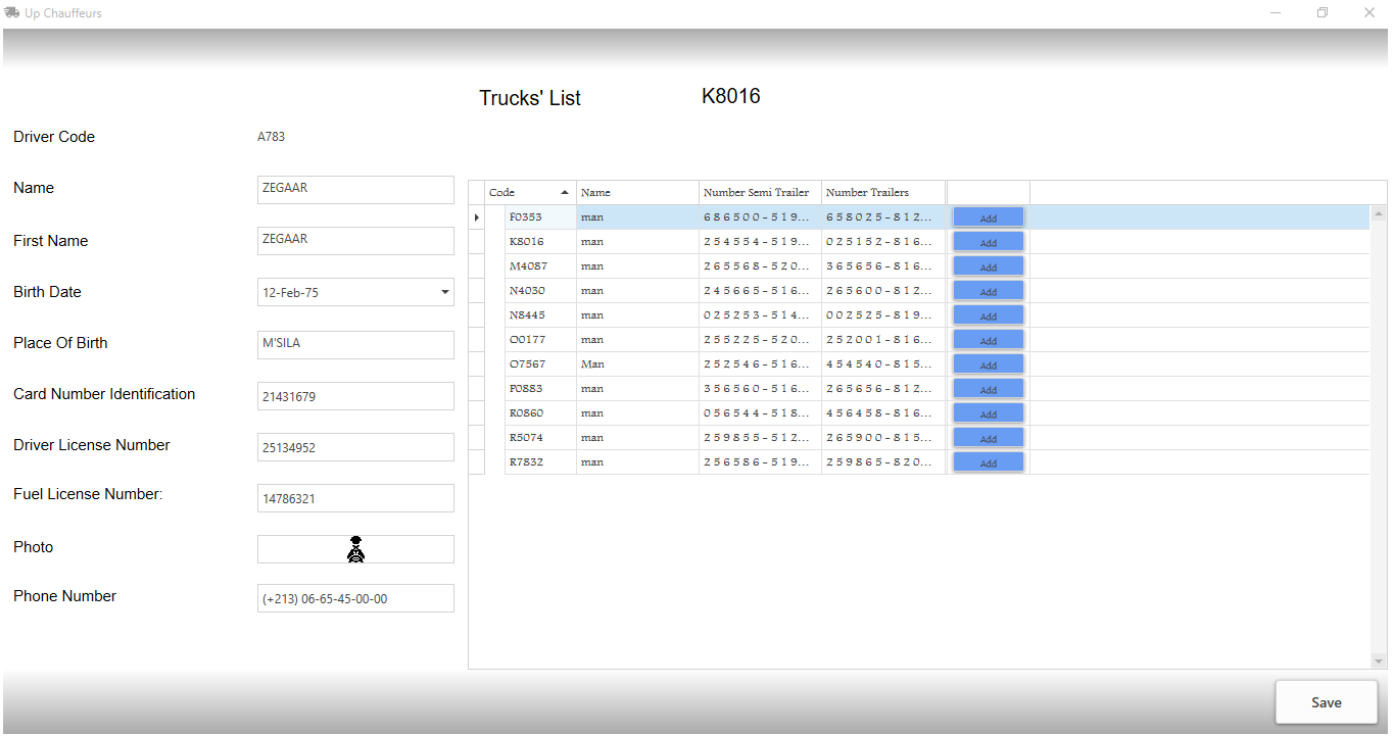


Figure 4.24: Update driver

3.6 : Order

3.6.1 Add Capacity :

Scheduled capacity for tomorrow

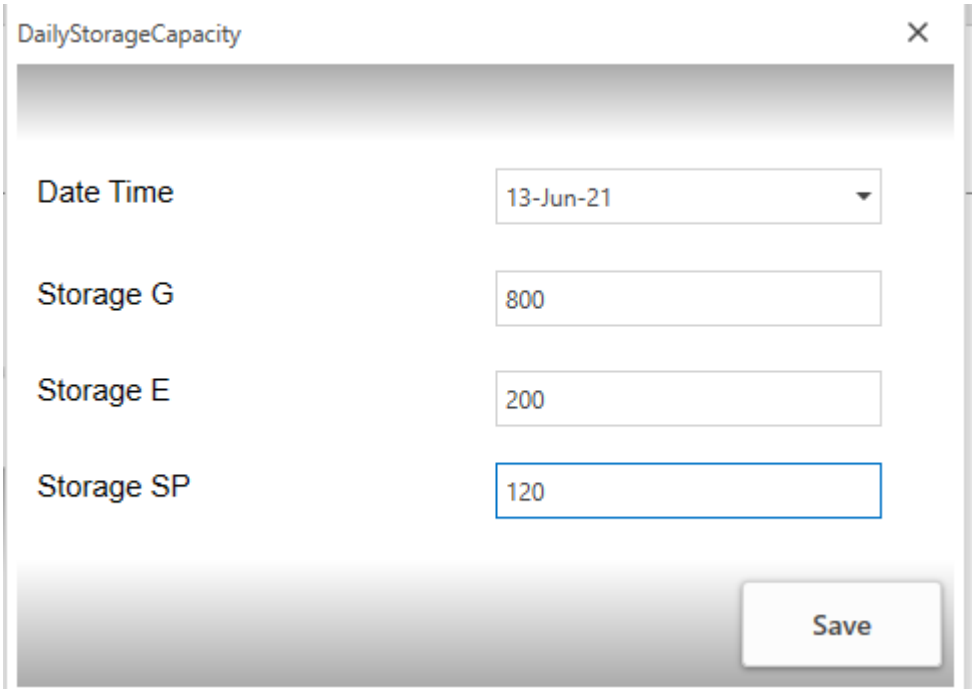


Figure 4.25: Add capacity

3.6.2 Order List :

List of orders for a specific day

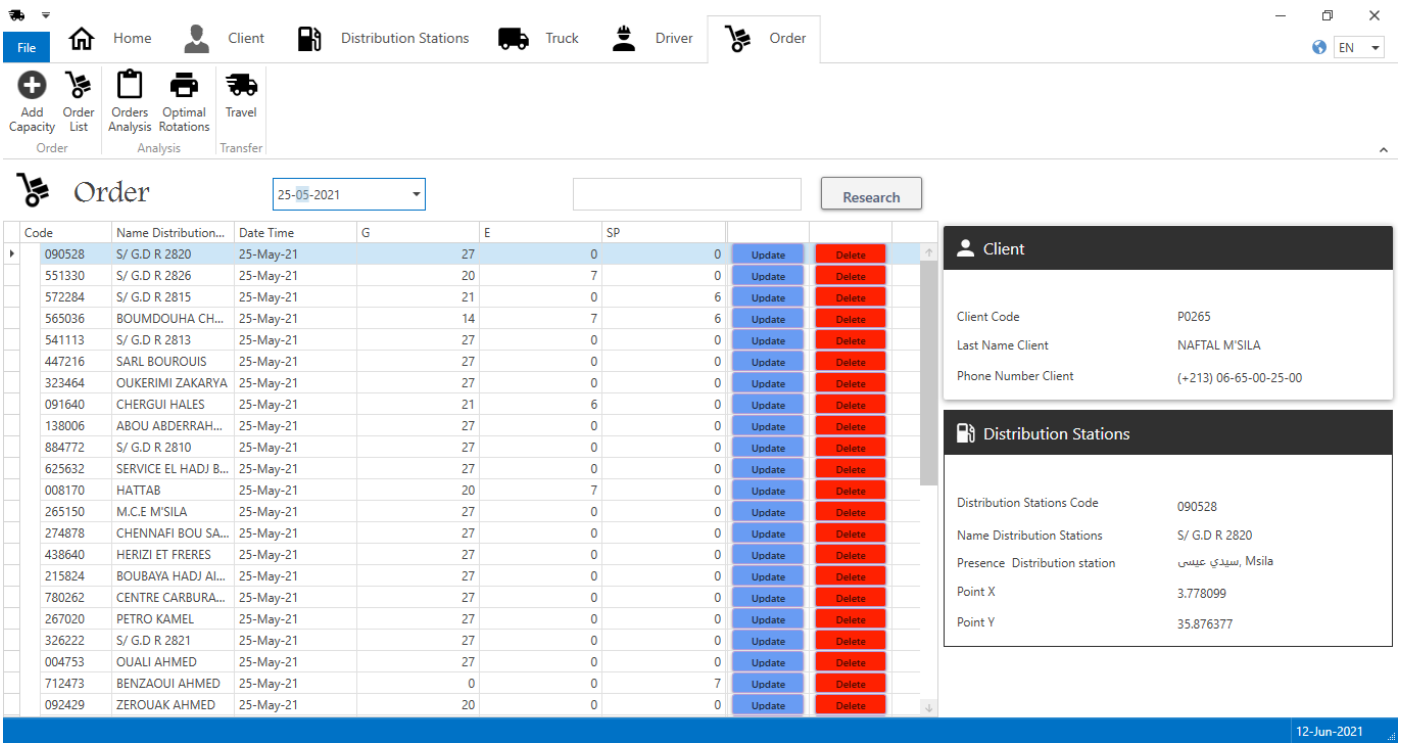


Figure 4.26: Order List

3.6.3 Orders Analysis :

Order analysis and travel by driver and truck

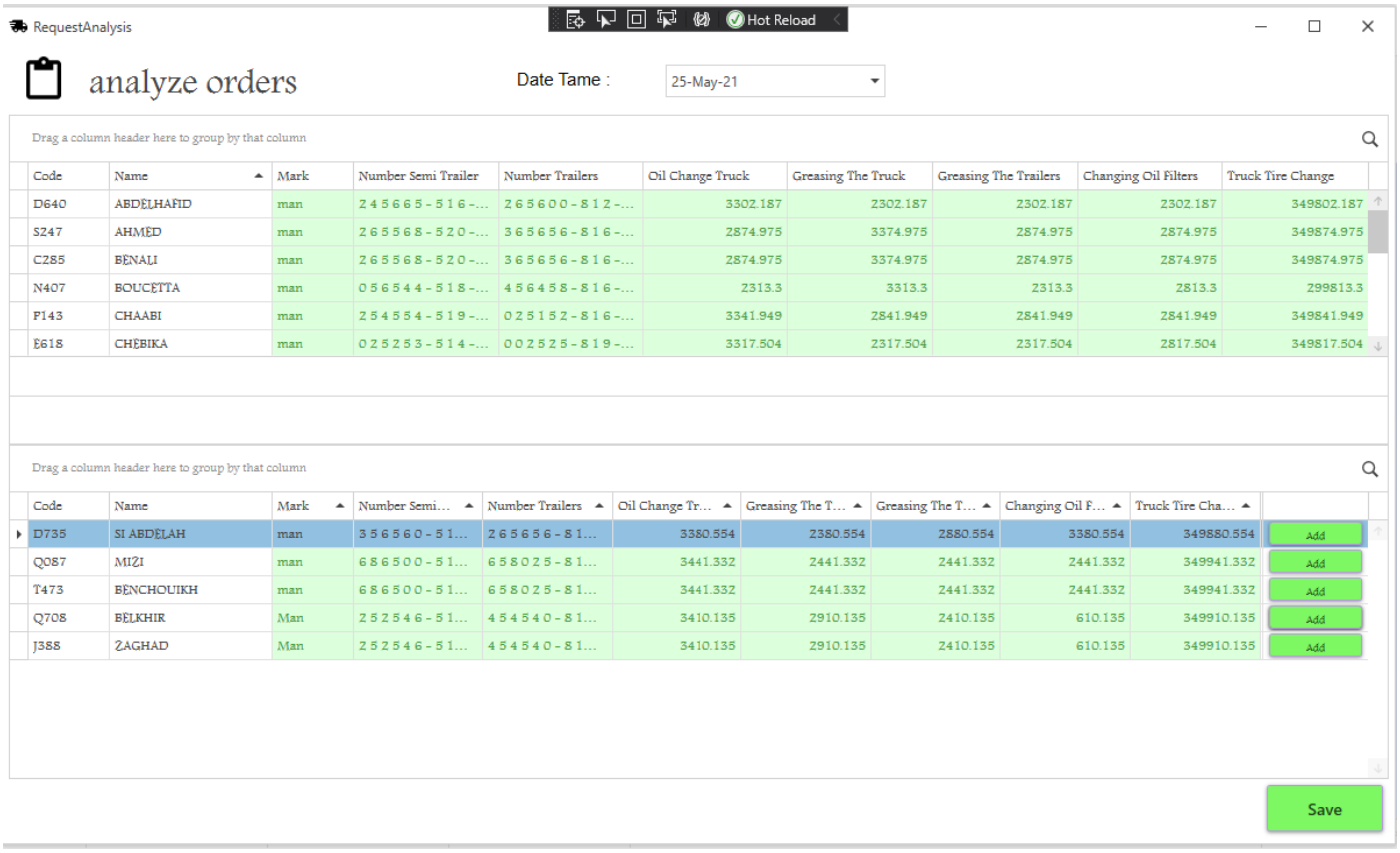


Figure 4.27: Orders Analysis

3.6.4 Travel :

Travel lists for each truck and driver

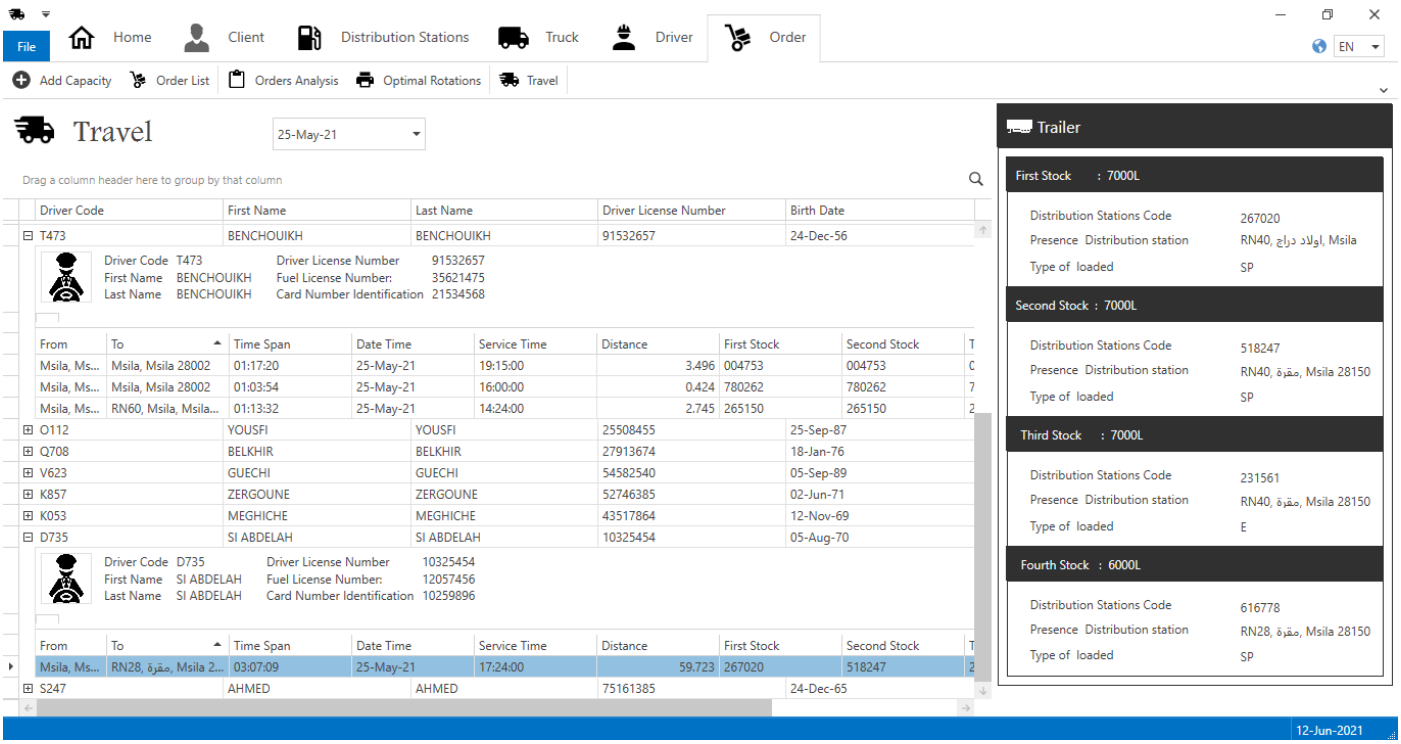


Figure 4.28: Travel list

4 Conclusion

In this chapter we have presented the technical part of our work. We have presented the technical environment and the necessary tools used in our development process. Some interfaces for the main functionalities are also presented

General Conclusion

General Conclusion :

We have proposed in this work an information system based optimization for fuel delivery management in NAFTAL company. The proposed work shows the importance of the delivery management in optimal and intelligent way. It improves and facilitates the handling and the analysis of orders by quantity in storage as well as the number of trailers.

The proposed solution can bring many advantages to the targeted company such as:

- Optimization of the time delivery
- Optimization on the number of the shipments
- Quickest response time after receiving orders from fuel stations.

The proposed solution has been designed under the UML language and implemented using essentially the C# language under Visual Studio IDE and SQL server as data base manager.

In the future and as perspectives, we propose to continue to work on multilateral trading systems and develop new ways to solve problems, particularly in the area of artificial intelligence. Other sides of the proposed system can be handled using optimization strategies, notably by the exploration of data mining and big data technics.

References

[1] Lawler, E. L. (1976). Combinatorial Optimization: Networks and Matroids. Holt.

[2] Hoos, H. H., & Stützle, T. (2005). Stochastic Local Search: Foundations and Applications. Morgan Kaufmann.

[3] Retrieved . Apache Tomcat/9.0.2. https://dspace.univ-msila.dz:8080/xmlui/bitstream/handle/123456789/2204/TAIBI%20Salah%20Eddine.pdf?sequence=1&isAllowed=y April 16 , 2021

[4] Weise, T. (n.d.). Home. Retrieved, from http://iao.hfuu.edu.cn/blogs/25-what-is-optimization April 16, 2021

[5] Retrieved. Department of Information and Computing Sciences - Universiteit Utrecht. https://www.cs.uu.nl/docs/vakken/mads/lecturecomplexity1.pdf April 16,2021

[6] Rothlauf, F. (2011). Design of modern heuristics: Principles and application. Berlin: Springer.

[7] Allaoua Hemmak (n.d.). Retrieved. Department of Information and Computing Sciences - Universiteit M'sila. https://elearning.univ-msila.dz/moodle/pluginfile.php/94716/mod_resource/content/1/support%20de%20cours.pdf May 09, 2021

[8] Https://themeforest.net/user/dan_fisher. (n.d.). MDA specifications. OMG | Object Management Group. https://www.omg.org/mda/specs.htm May 25, 2021

[9] UML - Use case diagrams. (n.d.). College of Science and Engineering | University of Houston-Clear Lake. https://sceweb.sce.uhcl.edu/helm/WEB-TOC-UML/Tutorial/uml__use_case_diagrams.html?fbclid=IwAR1Ouz1v_iT8qh7C8mQ846mSnZiEgvWNjUEPOGLvnpGrQHSXSWqKYztV-Hk May 23, 2021

[10] *UML tutorial*. (2021, May 21). RxJS, ggplot2, Python Data Persistence, Caffe2, PyBrain, Python Data Access, H2O, Colab, Theano, Flutter, KNime, Mean.js, Weka, Solidity. https://www.tutorialspoint.com/uml/index.htm May 24, 2021

[11] 99 library support in visual studio 2013. (2019, February 18). C++ Team Blog. https://devblogs.microsoft.com/cppblog/c99-library-support-in-visual-studio-2013/ May 25, 2021

[12] F# at Microsoft research. (2017, June 9). Microsoft Research. https://www.microsoft.com/en-us/research/project/f-at-microsoft-research/ May 25, 2021

[13] Best Python IDE for Python programming – Pythonic quest. (2017, December 17). Pythonic Quest – A Pythonic Quest. https://www.pythonicquest.com/article/best-python-ide/ May 25, 2021

[14] Chrisnb. (n.d.). Visual studio product lifecycle and servicing. Developer tools, technical documentation and coding examples | Microsoft Docs. https://docs.microsoft.com/en-us/visualstudio/releases/2019/servicing#support-for-older-versions-of-visual-studio May 25, 2021

[15]Gewarren. (n.d.). .NET framework & Windows OS versions. Developer tools, technical documentation and coding examples | Microsoft Docs. https://docs.microsoft.com/en-us/dotnet/framework/migration-guide/versions-and-dependencies May 25, 2021

[16] ECMA-334. (2020, November 27). Ecma International. https://www.ecma-international.org/publications-and-standards/standards/ecma-334 May27, 2021

[17] Nathan, A. (2006). Windows presentation Foundation unleashed. Pearson Education.

[18] MacDonald, M. (2010). Pro WPF in VB 2010: Windows presentation Foundation in .NET 4. Apress.

[19] Inc., D. E. (n.d.). About us: Our mission and contacts | DevExpress. .NET UI Controls for Developers of Mobile, Desktop, Web, Reporting & BI Apps. https://www.devexpress.com/aboutus/ May 25, 2021

[20] Présentation du logiciel de base de donnees Microsoft SQL server. (2021, May 17). Développement web et mobile. https://consultant-webdesigner.fr/logiciel-base-de-donnees-ms-sql-server/?fbclid=IwAR2-HDlEUezwzfQEFDASXCBlBnofV-hBW5dwIaMU0zCxGbgiGdOzBJD1UIU May 25, 2021

[21] Exclusive: Quantum paper and Google's upcoming effort to make consistent UI simple. (2014, June 11). Android Police - Android news, reviews, apps, games, phones, tablets. https://www.androidpolice.com/2014/06/11/exclusive-quantum-paper-and-googles-upcoming-effort-to-make-consistent-ui-simple/ May 25, 2021

Annexes

الجمهوريـــة الجزائريـــة الديمقراطيـــة الشعبيـــة
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التعليم العالي و البحث العلمي
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

جامعة المسيلة
كلية الرياضيات والإعلام الآلي
قسم الإعلام الآلي

Université de M'SILA
Faculté Mathématiques et Informatique
Département d'informatique

رقم :21/ق.إ.آ/2021

المسيلة في:2021/01/05

إلـــى السيد : مسؤول مؤسسة نفطال فرع مسيلة

الموضوع : مساعدة الطلبة في إجراء تربص ميداني

في إطار ربط الصلة بين الجامعة والمحيط الاقتصادي يشرفنا أن نلتمس من سيادتكم اتخاذ الإجراءات اللازمة لتمكين الطالب المذكور أدناه من إجراء تربص ميداني بمؤسستكم :

الرقم	الاسم و اللقب	تاريخ ومكـــان الازديـــاد	رقم بطاقة الطالب
01	ديقش السعيد	1996/12/18 بعين الملح	161635094754

مدة التربص : 60 يوم

رئيس القسم

المؤسسة المستقبلة

A.BENZIANE
Chef d'Antenne

ملاحظة: الطلبة المتربصين بمؤسستكم لهم كل الحقوق البيداغوجية و الحماية الاجتماعية بالجامعة بما فيها التأمينات .

الملخص

إن الهدف من هذة المذكرة هو انشاء نظام يساعد على تخطيط عملية النقل في إحدى المؤسسات الوطنية للإنتاج ونقل المواد البترولية و مشتقاتها (NAFTAL), لذا قمنا بعمل دراسة لهيكل ومخطط توزيع المواد للشركة المتعلقة بهدف التسيير الأمثل لهذه الوظيفة من خلال نقل أكبر كمية من منتج (وقود) من أماكن العرض إلى أماكن الطلب بسرعة وفعالية, قمنا بعمل دراسة تحليلية ومقارنة بين اشهر انظمة البحث ومن ثم استخالص افضل الطرق لتطبيقها فى النظام, بعد ذلك حصلنا على النتائج المرجوة من ناحية الفعالية وهذا لخفض التكاليف الإجمالية التي تتحملها المؤسسة ، ليتم في الأخير وضع خطة نقل مثلى استنادا إلى ما هو متاح لدى المؤسسة.

كلمات داللييه: النقل,تحسين ,توزيع,الوقود ,تسيير.

Abstract

The objective of this note is to establish a system that will assist in the planning of the transfer process in one of the national institutions for the production and transport of petroleum materials and their derivatives. (NAFTAL), so we did a study of the structure and material distribution scheme of the company with a view to optimizing this function through the transfer of the largest quantity of product. From supply to demand places quickly and effectively, we did an analytical study and a comparison of the most popular search systems and then we took the best way to apply them to the system, and then we got the desired results in terms of efficiency and this to reduce the overall costs to the enterprise, so that an optimal transport plan is finally developed based on what is available to the enterprise.

Keys words: Transportation, Optimization, distribution, fuel, operation.

Résumé

L’objectif de cette note est d’établir un système qui facilitera la planification du processus de transfert dans l’une des institutions nationales pour la production et le transport de matières pétrolières et de leurs dérivés. (NAFTAL) , nous avons donc fait une étude de la structure et du schéma de distribution des matériaux de l’entreprise en vue d’optimiser cette fonction par le transfert de la plus grande quantité de produit. De l’offre à la demande rapidement et efficacement, nous avons fait une étude analytique et une comparaison des systèmes de recherche les plus populaires, puis nous avons pris la meilleure façon de les appliquer au système, et puis nous avons obtenu les résultats souhaités en termes d’efficacité et cela pour réduire les coûts globaux pour l’entreprise, de sorte qu’un plan de transport optimal est finalement développé sur la base de ce qui est disponible à l’entreprise.

Mottes clés: Transport, Optimisation, distribution, carburant, exploitation.