National Technical University of Athens
School of Naval Architecture and Marine Engineering
Division of Marine Engineering

Diploma Thesis:

"Development of a computational environment for the technical and financial assessment of vessels with hybrid-electric propulsion"

Michail Dimitrios

January 2019

National Technical University of Athens
School of Naval Architecture and Marine Engineering
Division of Marine Engineering

Diploma Thesis:

"Development of a computational environment for the technical and financial assessment of vessels with hybrid-electric propulsion"

Michail Dimitrios

Supervisor: J. Prousalidis

Associate Professor N.T.U.A. of Marine Electrical Engineering

January 2019

Acknowledgements

I would like to express my very great appreciation to Professor J.M. Prousalidis for his valuable and constructive suggestions during the planning and development of this research work.

Furthermore, I wish to thank Mr. Bakirtzoglou for his valuable help and technical support for the completion of this dissertation. Mr. Bakirtzoglou, whose thesis was the foundation for the creation of my dissertation, supplied in full his knowledge on the subject and provide me with all the computational tools created in his study.

In addition, I would like to thank my family for their unconditional support, which helped me through my studies and especially my mother, who kept me alive providing me with food and internet connection.

Last but not least, I wish to express my sincere regards to Borri Marios, Fouroulis Stavros, Konstantoni Elena, Kontoulis Christos, Liadi Polina, Mourkokosta Vassiliki, Pittara Antonia, Pontikos Simos, Stratis Antonis, Theofanidis Dimitrios, Toki Christina, Tsarouxa Anthi, who supported me, both spiritually and physically, and tolerated with my crampy mood during my studies.

Abstract

The growing interest in echo friendly technologies, created the need of a powerful computational tool that would effortlessly give a first impression of the feasibility of such applications. The consistently increasing petroleum, and fossil fuels in general, rates have led the marine industry to look for and embrace more eco-friendly technologies for the powering of commercial vessels. The most important and more common one is the idea of an all-electric vessel, fully disconnected by the need to use fossil fuels. Moreover, the short distance routes in the Greek marine environment make the use of such applications ideal. Until nowadays there have been plenty studies in the field of all- electric conversion for the commercial vessels. In this thesis the creation of a powerful computational tool has been achieved, providing the user with the capability to make a time cost study to be a matter of minutes.

The development of such a tool have been accomplished with the use of Visual Studio 2017 environment and was written in Visual Basic.net. The methodology used was created by Christos Bakirtzoglou in his thesis and will be shortly presented here. This methodology has been converted to fit the needs of programming writing. It was divided to parts including Basic Calculations, Financial Calculations and Emissions Calculations. These parts have been assigned to individual Classes, so that could be called form the main program at runtime. In the Basic Calculations class, the energy demand is calculated in respect of the time that is available for charging the batteries in each trip of the vessel. Furthermore, the batteries number, volume and total cost is given as a result, respectively to the charging time in each trip. In the Financial Calculations Class, the financial elements of the study are being calculated. These calculations have been divided in the installation cost of the batteries, the operational costs for the batteries and the old system, including the cost of electricity for charging and the cost of currently used fuel and the maintenance costs for the two systems. Resulting in the benefits calculations have as output the benefits of such conversion in respect of the system’s life. Conclusively in the Emissions calculation Class, the total mass and cost of emissions produced by the old system are calculated.

Furthermore, in the last chapter the results of Christos Bakirtzoglou thesis are used to prove this development tool’s capabilities.

Table of Contents

1. Introduction

1.1. Background of the study

The purpose of the present dissertation is to create a powerful computational tool for the financial and technical evaluation of the conversion of a conventional vessel into and an all-electric one, powered by battery cells.

With the first successful attempts of electrical propulsion systems in the late 1830s, way before the use of internal combustion engines, this new kind of technology was proved to be more than useful for the propulsion of small commercial vessels. The first all-electric boats where used in the protected environment of small lakes and canals. This type of environment was ideal as due to the fact of extreme-weather free conditions, which required even less power for the maneuvering of the vessels used. Significant progress was made, especially before and during the Word War I, when plenty of submarines and naval vessels where equipped with electric propulsion systems. The electric motors had the ability to be extremely quiet and easier to maneuver compared to the mechanical based systems. Unfortunately, until the 1970s the use of such technologies did not have big interest, as the alternative of internal combustion engines, that use fossil fuels was a more economical solution. From then and on, big progress has been noted in the field of storing electrical energy and using it to produce effective power.

Nowadays, there is a much bigger interest in the use and evolution of electric propulsion systems. The interest in such systems was revived, especially with the import of more eco-friendly solutions for the production of electrical energy, including solar panels and wind-power generators. These solutions led to the launching of the first all-electric-solar powered vessel Tûranor PlanetSolar, at 2010 in Kiel, Germany. In addition, the electric vessel idea was not restricted only in research vessels but was applied in commercial vessel too. The world's first fully electric battery powered vessel was the MF Ampere, which was of passenger and car ferry type. It is important to note that the fuel savings reached the amount of 60% compared to the traditionally powered ferries. After the launching of MF Ampere, many imitators showed interest in all electric vessels, leading to the launching of several ships in Norway and Sweden.

Every day the supporters of eco-friendly technologies are getting more and more. Even though the marine industry is a field that traditionally fossil fuels where used for the propulsion of the vessels, today there is a rapidly growing interest for alternative types of energy production. This is the result of the continuously growing prices for petroleum and its derivatives. In addition to the cost-efficiency problem, the last few years there a big effort is made by the Classification societies to enforce ship owners to apply eco- friendly technologies to their vessels. More specifically, already there are new regulations about the Water Ballast Treatment, which obligate the treatment of ballasting water before its discharge to different eco-systems. With the Water Ballast Treatment Regulations making the start, now there are new Regulations about emission control for the vessels, which include the explicit use of low Sulphur fuel or the obligatory installation of an emission control system. However, either the use of Low Sulphur fuel or the installation of a Scrubber systems means a noticeable higher cost for the ship owners or the charterers. Concluding to the need of new cheaper technologies that will be also friendly for the environment.

In this case, the option of an all-electric vessel can be found more than useful. With such systems installed onboard, the emission's production almost immediately cuts to 60-70 % of that compared with the fossil fuel use system. Furthermore, the initial cost for such systems can equilibrate the cost of the installation of a scrubber system, or it could amortize the capital spent for one-year use of Low Sulphur fuel. Moreover, the lower cost of such systems is not the only benefit, the significant lower emissions are something to be noted. As far as the operation and maintenance of an all-electric vessel is concerned the emissions produced are almost zero. To be honest the only emissions produced are those at stage of electricity production on land, which can easily be eliminated by the use of renewable energy sources.

The Greek landscape consists of innumerable islands, with a big portion of them being habitable. There is a big need to connect these islands with the rest of Greece and provide them with goods, building materials, agricultural equipment etc. The prementioned connection has been achieved over the years, with over 200 line-routes being active today. The short distances between these islands are ideal for the application of fully battery powered vessels. This is due to the fact that, in order for such an application to be economically feasible, the distance travelled or the radius of autonomy of the vessel needs to be relatively small, so there it will be plausible to power the vessel with a logical amount of batteries and recharge them every few hours. As far as the local power stations that need to be installed accordingly at given ports, for now the national electrical grid's capability is not enough to support such kind of installations. The insufficient grid problem can be easily overcome, with the use of on-point solar panel parks for the production and the supply of big amounts of electrical energy needed to charge the batteries. The prementioned solution would be easily applied, taking in mind the plethora of sunlight in Greece during the whole year.

Considering the benefits of Greece for the application of such technologies, in the past few years several Technological studies have been completed trying to analyze the use and the benefits of an all-electrical vessel. Specifically, the National Technical University of Athens have taken a great part in completing most of these studies and publicize their results. The results showed that an application of an all-electrical vessel in a given Greek line-route would be economically feasible, only with the funding from European Union, but it would be tantalizing with the installation of an on-spot solar panel park. Furthermore, the electricity power installation could be combined with another newborn project known as Cold Ironing. In this field there are also plenty studies and applications around the world. Unfortunately, in Greece, there is not yet any application taken place. But, throughout the work of the National Technical University of Athens, plenty companies have shown interest in such applications and recently an effort has been accomplished to install one Cold ironing facility in the Port of Kilini. Concluding, the combination of the two Systems can certainly lead to profitable installations, both for shipowners and the Greek nation.

With the interest shown from the ship owning companies in the last 2-3 years, the need of quick computational tool for the cost and the feasibility of such installations was born. This tool should have the ability to take as input basic vessel characteristics, as far as the vessel's engines are considered, and give a quick result, which would adequately approach the cost of the conversion of the vessel to an all-electrical one. Consequently, from the result given, it will be decided if further analysis is worth, or the given installation is by far unprofitable.

The creation of such tool has been accomplished and will be presented in the following chapters.

1.2 Problem Statements and Objectives

The growing interest for investing in the all-electric vessel from Greek companies, has led to the need of an easy to use tool, which would quickly compute and answer a rather simple but important issue: Is the installation of all-electric vessel economically feasible, for the given situation, or not? This is the exact purpose for this dissertation · the creation of a powerful computational tool for the technical and financial assessment of vessels with hybrid-electric propulsion.

In addition to this, the desirable result could be easily obtained from experienced engineers to someone with no technical knowledge at all. This could be achieved by creating a simple but compendious User Interface, that could easily guide the user through the essential steps for the data insertion and the calculation of the result. Furthermore, additional information and tips would help the unexperienced user through every step.

Another objective of this thesis is to create and connect a database with the same computational tool, that would store every calculated vessel's information for future use. Also, vessel's information and characteristics could be inserted skipping the step of the calculation, in order to store this info for future calculations. In the meantime, the Vessel's Database can be used as representative sample of the commercial vessel's sailing in Greek territory.

The idea of a study for the calculation of the economic feasibility of the conversion of commercial vessels to all electrical is known before this thesis. Specifically, an analytical study on this subject has been accomplished by Christos Bakirtzoglou and is presented thoroughly in his dissertation "***Techno-economical feasibility study on the retrofit of double-ended Ro/Pax ferries into battery-powered ones***". The main objective of this thesis is to transfer Mr. Bakirtzoglou's study into an easy to use program written for Windows Base environments. In this thesis, no changes have been made to the study of Mr. Bakirtzoglou, as its objective was not to question or evaluate his method of calculations.

In addition to the basic computations, a comparison tool will be added to the program, with which the user will effortlessly compare different vessels or even the same vessel with different economical or technical characteristics. This tool will give a quick visual result of the vessels chosen to compare, so the user could easily end up to the best fitted solution for each case.

1.3 Structure of the study

In the second chapter the methodology of retrofit study is presented, as it was evaluated in Mr. Bakirtzoglou thesis. The basic data that are essential for the calculations of the study will be presented, followed by their basic use. Furthermore, the formulas and mathematical expressions that were used for the computation of the study will be explained, in both technical and mathematical terms. The given case studied by Mr. Bakirtzoglou will be introduced, by means of Vessel type and in general the conditions of the environment it took place. Finally, the results of the study will be analyzed and discussed briefly, in the means of financial and ecological impact.

The 3rd chapter will include the presentation of the computational tool that was designed in this thesis. The structure of the program will be analyzed with respect to Retrofit Study and methodology, that has been discussed in the first second chapter. At first, the data used for the calculations will be analyzed and explained. Additionally, the formulas and the mathematical expressions behind the calculations will be presented and the exact steps of the calculations will be discussed. At the end of this chapter the results of the program are going to be identified and compared with those from Mr. Bakirtzoglou study. Furthermore, each of the above steps is followed by the adequate example of code written inside the program.

A comparison between two vessels, will be used to check the program's validity of calculations. These projects have been calculated in other dissertations, but without the use of the program created in this thesis. The main object is to test this development environment, through these already studied applications. Furthermore, a presentation of the, newly added, comparison tool will be analyzed, ending with the projection of comparison tool's benefits in the given projects.

Finally, a discussion is made regarding the potential of the created program. Especially, ideas for further investigation and development will be expressed and useful tools that can be added in future versions will be recommended.

2. Retrofit Methodology

The methodology created in Mr. Bakirtzoglou thesis was based in the study of specific type vessels and generally short travel distances. A brief presentation of this methodology will follow:

2.1 Battery storage systems, some general knowledge and types of batteries

In order to fully understand the methodology used for the transformation of a commercial vessel to an all-electric, powered by batteries, it essential to gain some basic knowledge in the field of batteries.

Battery energy storage systems are well known and commonly used for the last 100 years. The energy produced can be successfully stored in a battery system with the form of chemical energy. For this reason, the battery can be defined as an electrochemical storage device that stores electrical energy in the form of chemical potential between its positive and negative electrodes.

The key components of an elementary battery cell are the anode and the cathode, which form the electrodes, the electrolyte and the separator, as shown in ***Figure 2.1***. Upon discharge, chemical reactions initiate a flow of electrons from the anode to the cathode, which produces an electric current in the external circuit. The separator allows for positive charges to migrate from the anode to the cathode in the electrolyte without the passage of other molecules.

Based on the desired output voltage and capacity a battery consists of one or more cells connected in series, parallel or both. Each cell consists of:

- The anode or negative electrode; the chemical reaction at the anode (oxidation) releases electrons that flow to the cathode through an external circuit. The anode material is selected based on its efficiency, high specific capacity, conductivity, stability, ease of fabrication and low cost.

- The cathode or positive electrode; the chemical reaction at the cathode (reduction) accepts electrons. The cathode is selected based on its voltage and chemical stability over time.

- The electrolyte completes the cell circuit by transporting the ions between the anode and the cathode. The electrolyte can be liquid, like water, acids, alkalis or solvents with dissolved salts. The electrolyte can be selected based on its high conductivity, non-reactivity with the electrode materials, and stability in properties in various temperatures, safety and cost.

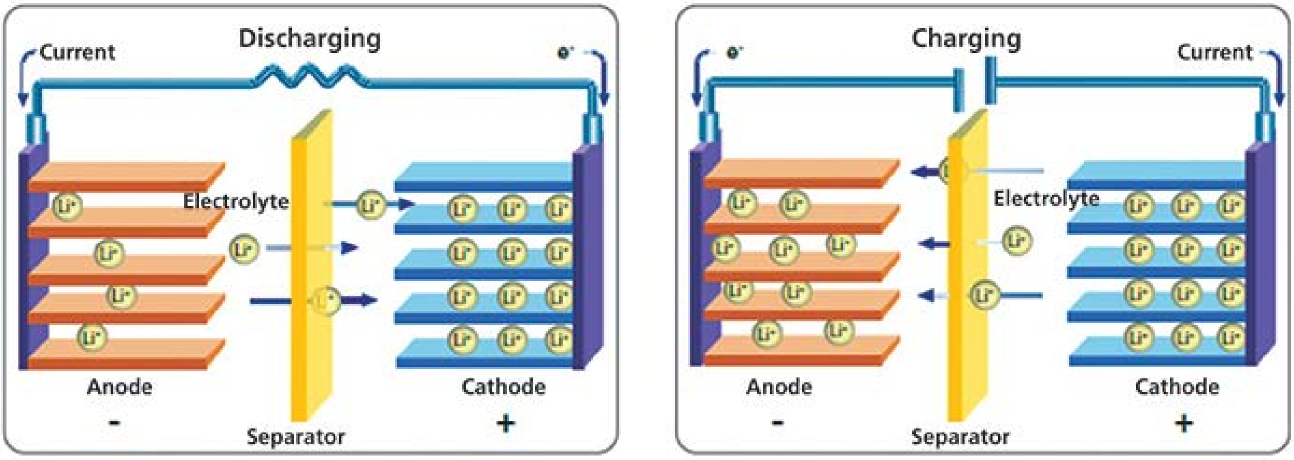

Figure 2.1: Battery Cell Function

Physically, the electrodes are electronically isolated preventing internal short-circuit situations, however, they are surrounded by the electrolyte. In a practical cell design separator which are permeable to the electrolyte are used to provide mechanical separation between the electrodes. The most beneficial combination of all these elements are those which result in a cell with light weight, high voltage and high capacity

When chemical reactions happen in a cell, chemical energy of the system decreases due to its transformation in electrical energy. The theoretical cell voltage is a function of the property of materials and is the sum of the anode and cathode potentials. The theoretical capacity is a function of the amount of active materials used in the cell.

Batteries are classified into two broad categories, primary batteries which host irreversible reactions and can thus be used only within a single cycle, and secondary batteries whose reactions are reversible and can be charged and discharged numerous times. Secondary batteries are charged by applying an external electric current. The current triggers the chemical reactions to operate in reverse, bringing the battery back to a state of high energy. Given the cyclical nature of marine applications, batteries used in marine industry are of course rechargeable.

The battery cells are made in different form factors, such as cylindrical, prismatic and pouch, and come in all sizes, from the small cells primarily made for consumer electronics to large sizes targeting heavy commercial applications. The maritime-focused systems have mainly been based on Li-ion cells with NMC (Nickel Manganese Cobalt Oxide) cathodes and graphite anodes. Systems based on iron-phosphate cathodes have also been used. Both the NMC and iron-phosphate chemistries represent a good compromise between the most important parameters of safety, energy, power density, cycle life and cost.

2.2. Legal & Regulatory framework

Due to the batteries cell design, a safety frame has to be set and used in industry applications. The safety framework has been defined by classifications societies, for the application in marine industry.

As far as the ROPAX vessels are concerned the following rules are in power:

According to I.M.O.'s rules (EUROSOLAS-directive 98/18/EC), to which Greece as a member state and as flag-state authority complies, state the following for double-ended Ro/Pax vessels under retrofit:

- Belong to class D, for passenger ships engaged only on domestic voyages in sea areas where the probability of significant wave heights exceeding 1.5 meters is less than 10% over a one-year period for all year-round operation.
- Serve routes of categories: VI. Regional routes (≤ 6 nm), VII. Protected zones
- Vessels serving routes VI and VII are allowed to be of "open-type"

For this study, the framework and set of regulations of DNV-GL is being followed:

- DNV-GL: Rules for classification, Part 6, Additional Class Notations (Oct.2015)
- DNV-GL: Guideline for Large Maritime Battery Systems (Mar. 2014)
- Lloyd's: Battery installations, Key hazards to consider and Lloyd's Register's approach to approval (Jan. 2016)
- DNV-GL: Tentative Rules for Battery Power (Jan. 2012)
- IEC61508: Functional Safety

- SOLAS: ChII-1: Electrical installation
- SOLAS: ChII-2: fire protection
- IEC 62619 9.2.3
- IEC 62620
- IEC 61508: Functional Safety
- IEC 62619
- IEC/ISO/IEEE 80005: Utility Connections Reports (– Shore Connection High Voltage)
- IEC/ISO/IEEE 80005-1: The onshore power supply standard high voltage
- IEC/ISO/IEEE 80005-2: Communication protocol

2.3. Battery System

Due to the battery system's high complexity, only some of its main features will be presented here, with the most alerting hazards following accordingly. The battery installation procedure is divided into two parts. The first one includes the installation that will take place onboard the vessel and it will be shown as Vessel-side topologies. The other side will include the installation and features that will take place on shore for the support of the vessel battery system and will be called as Shore-side topologies.

2.3.1. Vessel side topologies

Battery system

The battery system is the main installation, which will replace the common fossil fuel reservoirs energy wise. In other means it is the heart of the electrical vessel, as it contains and manages the total energy available for consumption on board. The main components of the battery system are the following:

- Battery cells
- Battery modules arranging hardware
- Battery pack and subpacks
- Thermal management components
- Safety features as fuses, contactors etc.

- Bus-bars and high voltage cabling
- Electronics
- Voltage and temperature sensors
- Low voltage cabling and connectors

In order that the reader will be in position to understand this methodology some of the above components need to be explained.

A **cell** is the smallest electro chemical unit. An assembly of cells including some level of electronic control forms the **module**.

The **modules** are connected into series and parallel to form a **sub-pack.** Sub-Pack is the smallest unit that can be electrically isolated. Depending on the system architecture, each sub-pack can have internal relays/contactors which can interrupt main power connection.

A battery **pack** consists of several parallel **sub-packs**. The battery system may consist of several battery packs. The electrical connections between the different aggregate levels of the battery system may be connected using cables, bus bars or a combination of these.

The **battery system** consists of one or more **battery packs** including all required systems that can work for the intended purpose as a standalone unit.
All the components of the battery system need to be carefully placed and surveilled because many dangers which can lead to hazardous situations may arise in all aggregated levels. The most important ones are referenced here:

CELL'S DANGERS:

- High Impedance
- Internal short circuit
- Insulation fault
- Electrolyte leakage

MODULE'S DANGERS:

- Short circuits
- Control Failure
- Temperature Sensor failure, Voltage sensor failure
- Internal open circuit, high impedance

- Internal Short Circuit
- Insulation fault
- Cooling system leakage
- Loss of Cooling

SUBPACKS' DANGERS:

- Contactor does not open/close when required
- Current sensor measurement error
- Connector high impedance
- Leakage of cooling connector
- Sub-pack enclosure leakage/damage
- Mishandling of battery system.

PACKS' DANGERS:

- High level sensor failure
- Voltage and temperature imbalance
- Battery life too short
- Contactor does not open/close when required
- Reverse polarity protection
- Emergency shutdown

Most of the above hazards are most probable to lead in a fire situation. Moreover, the system needs to be redundant, so that in case of failure of one part the rest system could provide energy for the vital procedures of the vessel. For that reason, we conclude that the battery system needs to be protected by fire and also have redundancy in case of failure. Henceforth, certain solutions will be discussed in the following paragraph:

Firstly, the battery casing, covering modules and cells, shall be made of a flame-retardant material. The batteries must be placed in dedicated rooms, that must be rated with A-0 integrity and A-60 integrity towards:

- Machinery spaces of category A as defined in SOLAS Reg.II-2/3
- Enclosed cargo areas for carriage of dangerous goods
- Muster stations and evacuation stations

In addition, battery space shall demonstrate robustness for long term exposure in a marine environment, such as temperature, moisture, list, trim, roll, etc., and shall provide

protection against external hazards (e.g. fire, mechanical impact, water ingress, pipes leakage). Furthermore, the battery system shall not be located without adequate protection from heat, ignition sources, dust, oil.
pollution or other potential harmful environmental influence to the system and its components.

Given these points, it is essential for the batteries to be stored in an independent, well insulated space. This space should not be the same with the machinery space, but should be used only for the storage of the battery system. Equally important, is the division of the battery system in at least two parts, so that the required redundancy is satisfied.

Except the battery system, the conversion for the given vessel is composed also by the replacement of the internal combustion engines with electric motors.

Although there are plenty alternatives for the design of the propulsion system, for the better understanding of the conversion an example for a specific vessel will be presented.

The vessel is a Roll-on Roll-off Passenger double ended vessel. It is equipped with 4 main engines, 4 azipod thrusters. The battery system is divided in 4 different arrays for redundancy and it includes an A/C distribution system.

The complete system is visualized with the following figure:

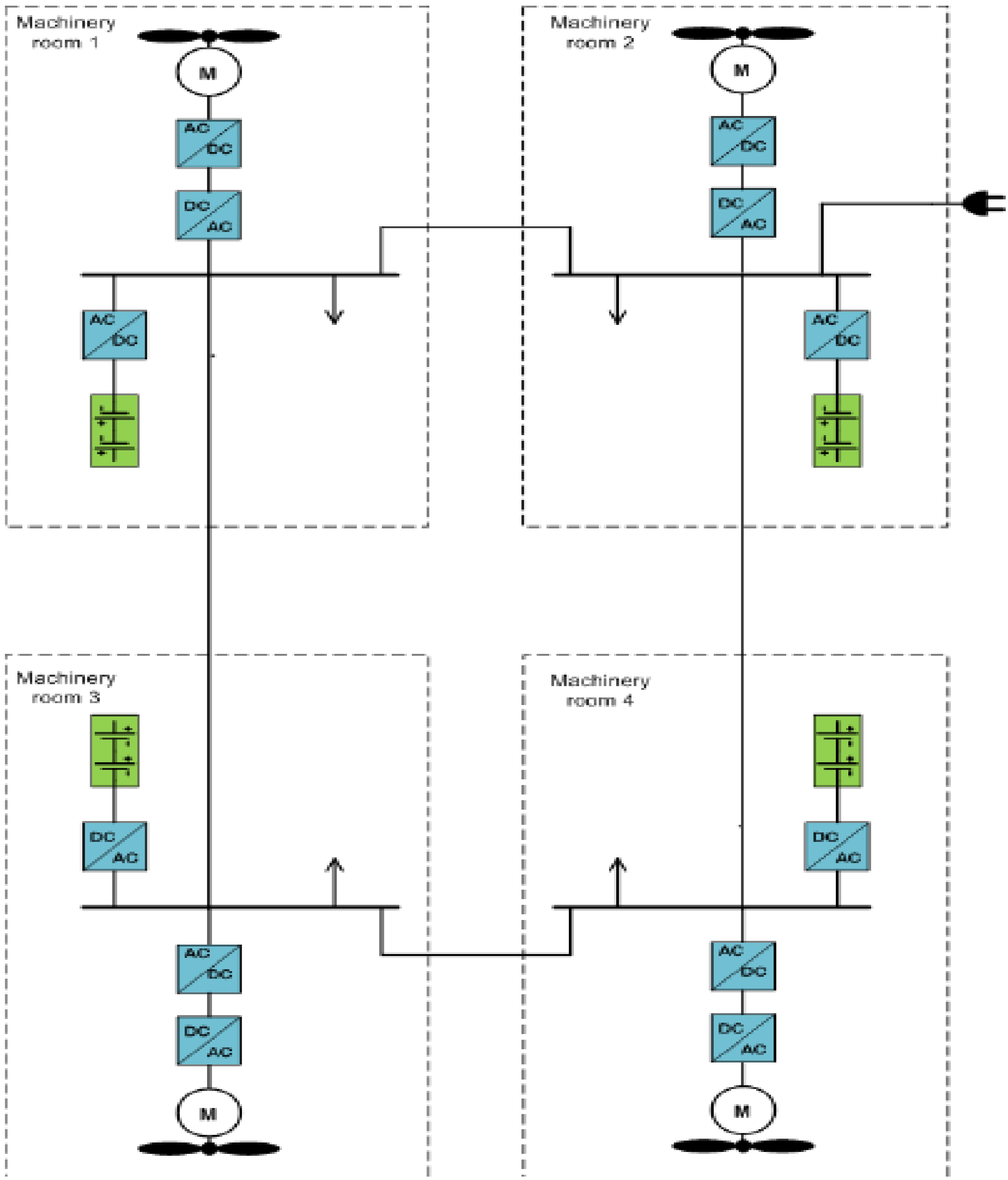

Figure 2.2: All-electric ship with an AC distribution network

The main components of this configuration are:

- Four inverters (AC/ DC), one after every battery array, for the conversion of the DC voltage of the batteries to AC.
- Four back-to-back converters (AC/DC/AC) for the control of the induction propulsion motors of nominal power equal or greater to the nominal power of the motor.
- 4 three-phase AC cables for the interconnection of the machinery rooms.

2.3.2. Battery shore side system

Although, in this thesis the financial cost of the shore side installation will not be analyzed, a short presentation will be given in the following paragraphs.

Figure 5-*4* below depicts the shore side configuration for an AC shore connection. It includes:

- A main substation equipped with a MV switchboard supplying the shore side substations
- Shore side substations supplying the connection points between the vessels and the port, equipped with:
  - An isolation transformer of Dynamic configuration for adapting the utility grid MV to the connection voltage, with the neutral point grounded (possibly through a grounding resistance)
  - The outgoing switchboard supplying the plugs of the point of connection between the port and the vessel

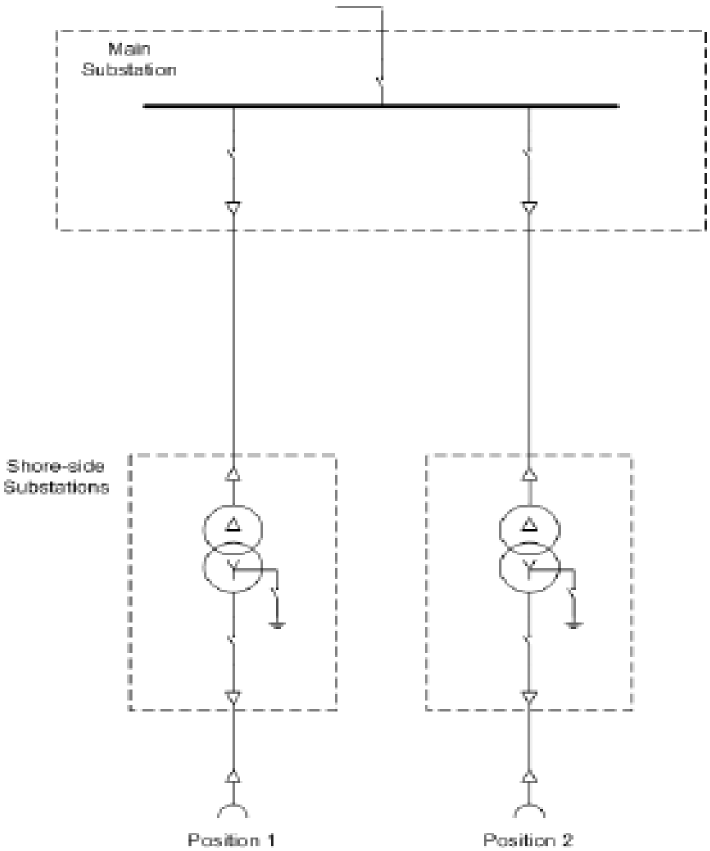

Figure 2.3: Shore side configuration for an AC LV shore connection

Similarly, the ship-shore connection system is not an object of this dissertation, so it will be presented briefly bellow.

A ship-to-shore connection cable installation should be arranged to provide adequate movement
compensation, cable guidance and anchoring/positioning of the cable during normal planned ship-to shore connection and operating conditions.

There are, mainly, two alternative CMS solutions on the market:

- the shore-based system
- the ship-based system

The option of fixed or mobile system is plausible for both shore-based and ship-based systems.
As far as the shore-based system is concerned, it can be installed firmly on the quay without the possibility to move it. On the contrary the mobile system could have the ability

to move along the dock, while be controlled by a user, or even be installed on a barge to supply ships moored at distance from the quayside.

The ship-based system could be installed on a fixed place along the main deck of the vessel or, to add mobility, it could be installed inside a specially arranged TEU.

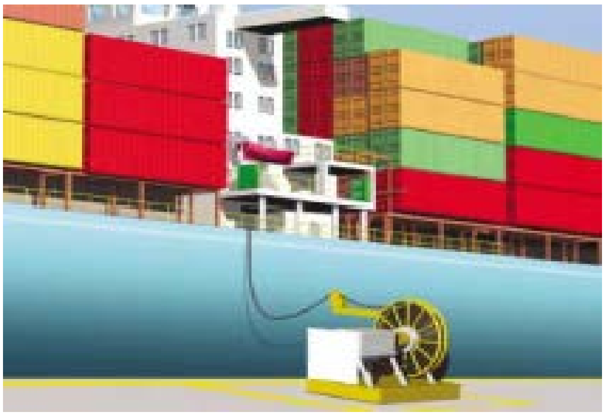

Figure 2.4: Fixed shore-based cable management system

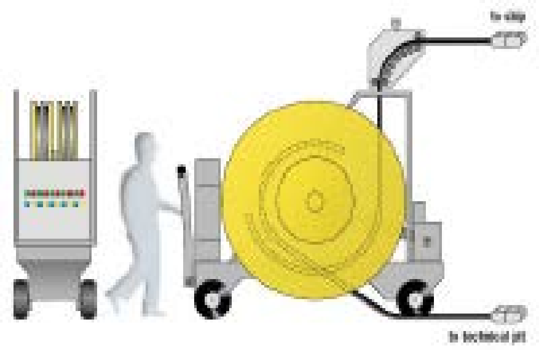

Figure 2.5: Mobile shore-based cable management system

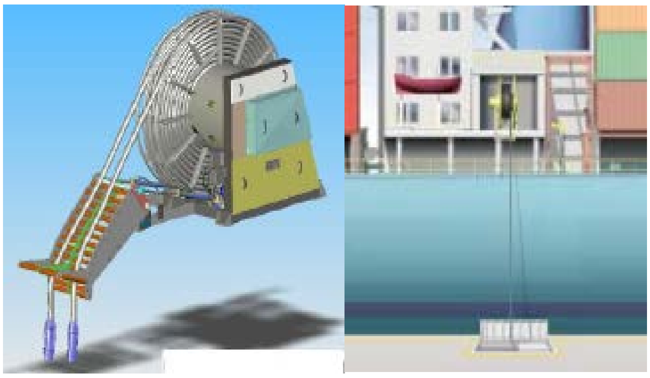

Figure 2.6: Fixed ship-based cable management system

Figure 2.7: Mobile ship-based cable management system

2.4. Input Data

Now, that the main parts of an electrical vessel transformation were presented, the data used as inputs for the calculations will be discussed. The data will be divided in sections, according to their nature (technical, economical etc.)

Model inputs for all stages of calculations

2.4.1. Engine related Data

The data related with the existing main engines and electrical generators are:

- No of Main Engines for propulsion and their nominal output
- No of Operating Main Engines for propulsion
- Main Engine Load Factor
- No of Electric Generators and their nominal output
- Electric Generators Load Factors
- Electrical Load Balance at Sea
- Electrical Load Balance at Port
- Electric Motors Diversity factor
- Electric Motors Efficiency number
- System's DC Voltage (V)

2.4.2. Route related Data

The data used to describe the route of the vessel are the following:

- Cruising distance (nm)
- Time Cruising (min)
- Time at Berth (min)
- Required(max) no. of trips per shift

2.4.3. Battery system related Data

As far as the batteries installed, the essential data for the calculations are:

- V nominal (V)
- Dimensions (m)
- Capacity (Ah)
- Volume (m3)
- Weight (kg)

- Nominal Charging/Discharging current for max lifecycles (A)
- C-Rate
- Nominal D.O.D.

2.4.4. Financial costs related Data

Finally, data regarding the cost of installed equipment and other costs need to be inserter. These data are:

- Batteries initial cost
- Batteries Inverter
- Motor Drivers
- Electric Motors
- Used Price for the Main Engines
- Used Price for the Electric Generators
- Electricity Price and Growing Rate
- Fuel Price and Growing Rate

2.5. Calculation Procedure

After, the above data have been gathered the required calculations can be done. Again, the calculations done in this methodology will be divided in sections, accordingly to the desired calculated result.

2.5.1. Calculations sections

Battery Values Calculations

In this section, the main goal is the calculation of the total batteries number needed to replace the existing engines and cover the energy needs of the vessel.

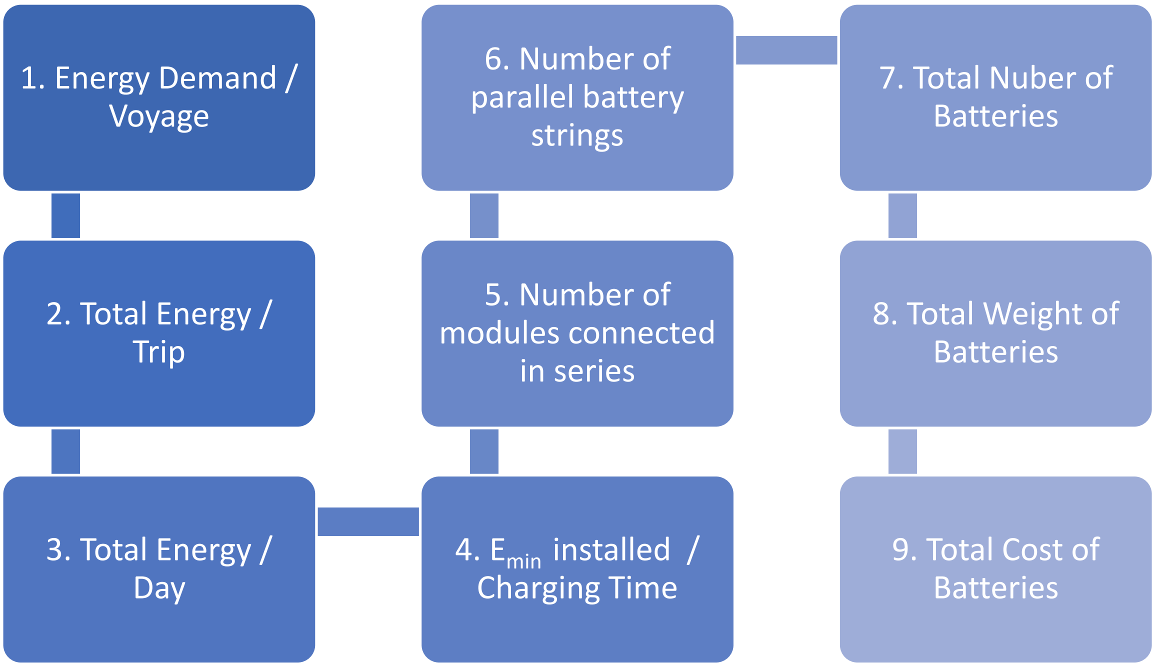

Figure 2.8: Battery Values Calculation Procedure

In Mr. Bakirtzoglou's thesis, two methods were presented for the calculation of the number of batteries needed. The first one was with respect to the number of trips /days, so that there would be no need for intermediate charging. The second one, take into consideration the capability of charging the batteries between each trip. Above, the second method is presented as this is the method used in this study.

As shown in the above flowchart, the first step is to calculate the energy demand of the vessel for one trip. It has to be mentioned, that one trip includes a two-way route between the starting port and the destination port. After, the minimum installed energy is calculated, according to the available time for charging in each port. Then, the number of modules connected in series is calculated, so that the voltage will be same with the desired one. The number of battery strings is calculated with respect to the systems desired amperage. The total number of batteries is calculated with a simple multiplication giving the desired result.

Operational and Maintenance Costs Calculations

This section is dedicated to the calculation of the total cost for the installation of a complete battery system, including the electric motors, a/c to d/c converters etc. Also, a cost benefit analysis takes place, respectively the years of the investment.

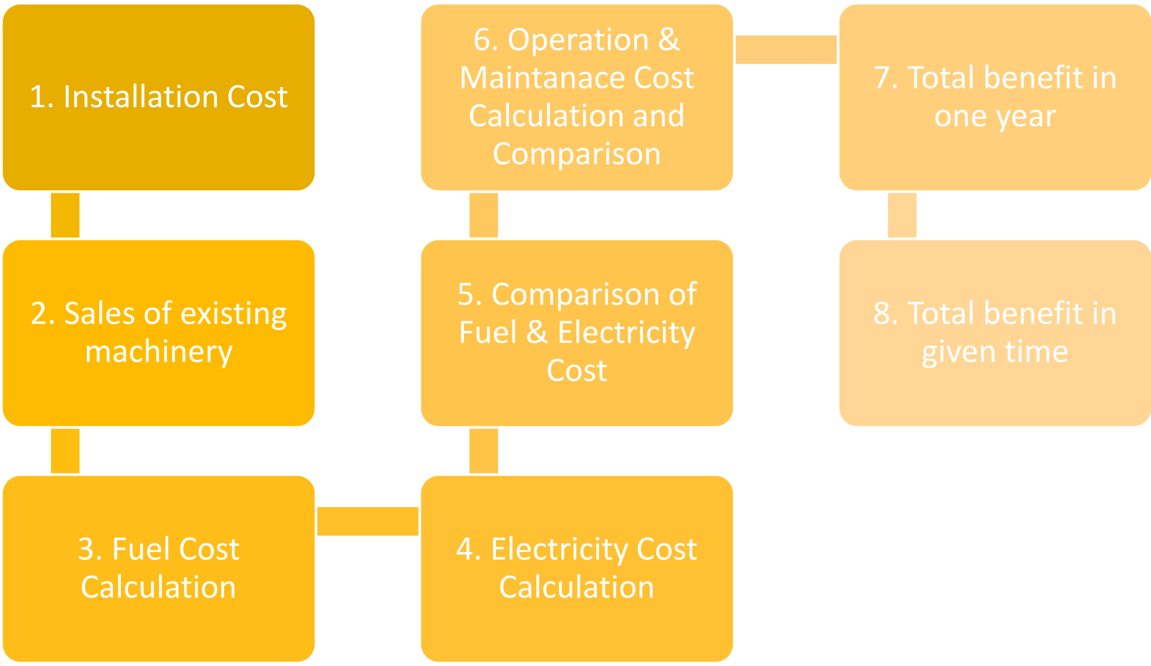

Figure 2.9: O&M Costs Calculation Procedure

First, the installation cost is calculated, which includes the cost of the batteries and the electrical components needed for propulsion, charging etc. Similarly, the earnings from used machinery items are calculated. After step 3, the most important calculation is done, which includes the cost of fuel and electricity for a year of operation. The cost of maintenance for a regular mechanical propulsion system and for an electrical one is computed, likewise. Forthwith, a comparison is made between the two systems in the period of one year and for multiple years accordingly.

Emissions produced calculations

Finally, the total mass of emissions produced from the existing machinery is calculated and an approximate cost analysis is made.

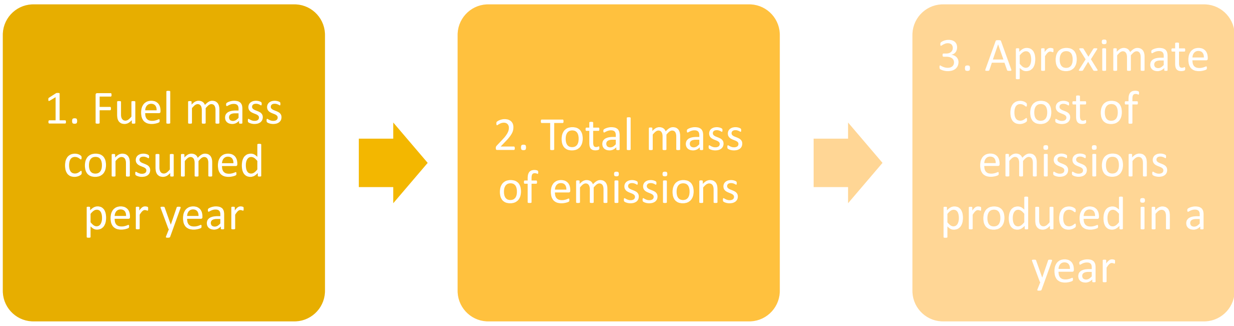

Figure 2.11: Emissions Values Calculation Procedure

The amount of emissions produced by current machinery systems for propulsion is significant. It is of high importance to reduce this amount to benefit social health and environmental prosperity. This goal can be achieved with the use of electrical power vessels. Given that, a method to calculated the benefits from pollution reduction is used in this study.

Firstly, the total fuel mass that is used in a period of one year is calculated. Secondly, the total mass of emissions is computed using the stoichiometry of marine diesel fuel, according to TIER III instructions. Given the above calculations and an emission-cost method it is possible to reach a result for the total cost benefit from the reduction of the emissions produced with the conventional propulsion systems.

3. Program presentation

3.1. Purpose of Creation

The current computation tool was developed to satisfy the need for a powerful way to calculate the cost of retrofit conversion for a current vessel, using conventional propulsion systems. The main objectives that need to be fulfilled are speed, accuracy and easiness to use. In order to fulfill the objective of accuracy, an already proven for accuracy study was chosen, that was created and presented by Mr. Bakirtzoglou. Despite its accuracy, this method is aimed at individuals with engineering and electrical engineering knowledge. For this reason, an easy to use program should be created, so any individual with basic understanding of engineering could use it to end up to the same result of Mr. Bakirtzoglou's thesis. Furthermore, programming a tool to proceed in background with all the calculations needed, could effectively speed up the process.

3.2. Creation Process

For the creation of the program Mr. Bakirtzoglou's thesis was used as basic methodology. The program was written in Visual Basic.NET programming language, and it was developed in Visual Studio programming environment. It is written in Visual Basic .NET, because it seemed to be the most convenient language for use at the time, given the short time schedule. Visual studio is an integrated development environment created by Microsoft. It was chosen, due to its simplicity and ease of design the user interface. For this reason, the developing of the program started by creating the User Interface, with the actual programming following after the basic idea for the interface was completed. In particular, the creation of the program was divided in sections, trying to mimic the already written study of Mr. Bakirtzoglou. First, the main form was designed, in which all the main controls are located. Second, a database was created inside a local server, which contains basic data for vessels calculated in the program, and can be accessed from whichever personal computer. Third, a simple insert form was design, providing the user with an easy way to insert data into database. Fourth, the main calculator process was designed and created, in which the user can enter the required data for each vessel and result in an approximation for the cost of the conversion to an electrical ship. Last, a report form was designed, through which the user can visualize the results of each analysis and additionally can compare different scenarios of studies.

In the following paragraphs the developed calculation tool will be presented, with respect to the creation procedure and the methodology, which was explained in the second chapter.

3.2.1. Design of the Main Form

The main form is the first template that the user will come across when he opens the program. It is designed in such way, so it will be simple but in the same time can provide the user with the required information and tools to use the program. The form can be seen in the following image.

Figure 3.1: Main Form

A sliding menu was installed at the left side of the form. From this menu the User can easily navigate to the desired part of the program he wishes to use. As it can be seen, the menu has three tabs. The Home tab, the Vessel Database tab, and the Retrofit Calculator tab.

3.2.2. Home tab

When the user clicks the Home tab the following template becomes visible.

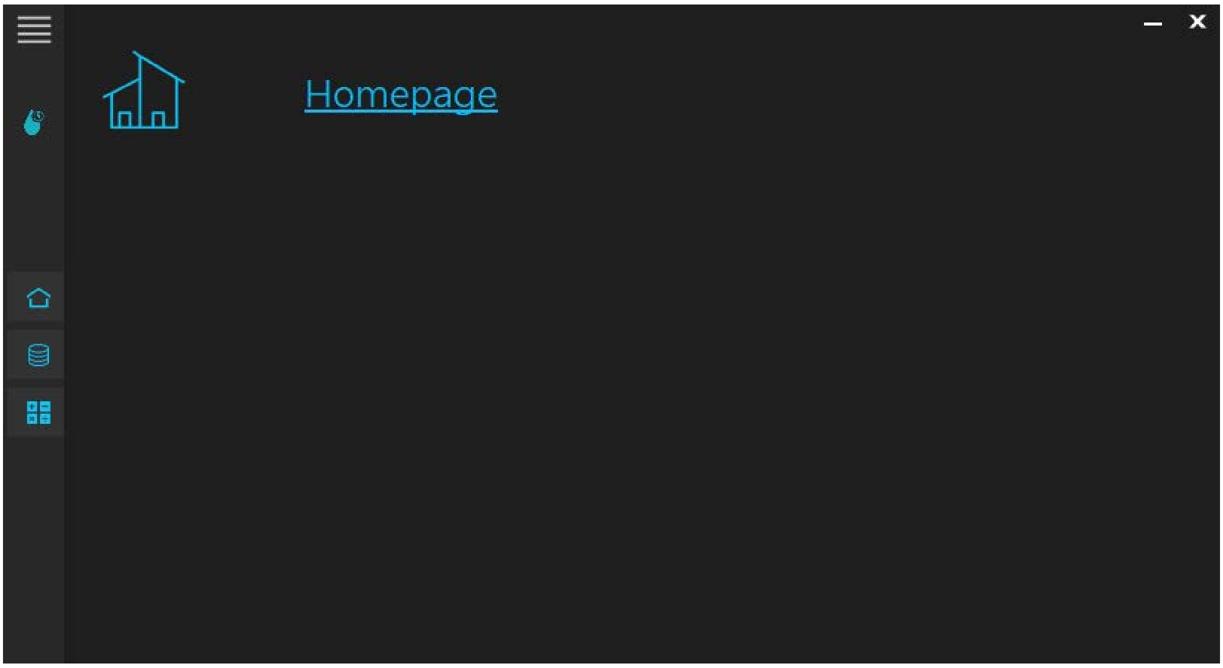

Figure 3.2: Home Tab

Here the user will see, basic info about the program and quick start tutorials, that will help him use the program successfully.

3.2.3. Vessel Database tab

When the user clicks the Home tab the following template becomes visible.

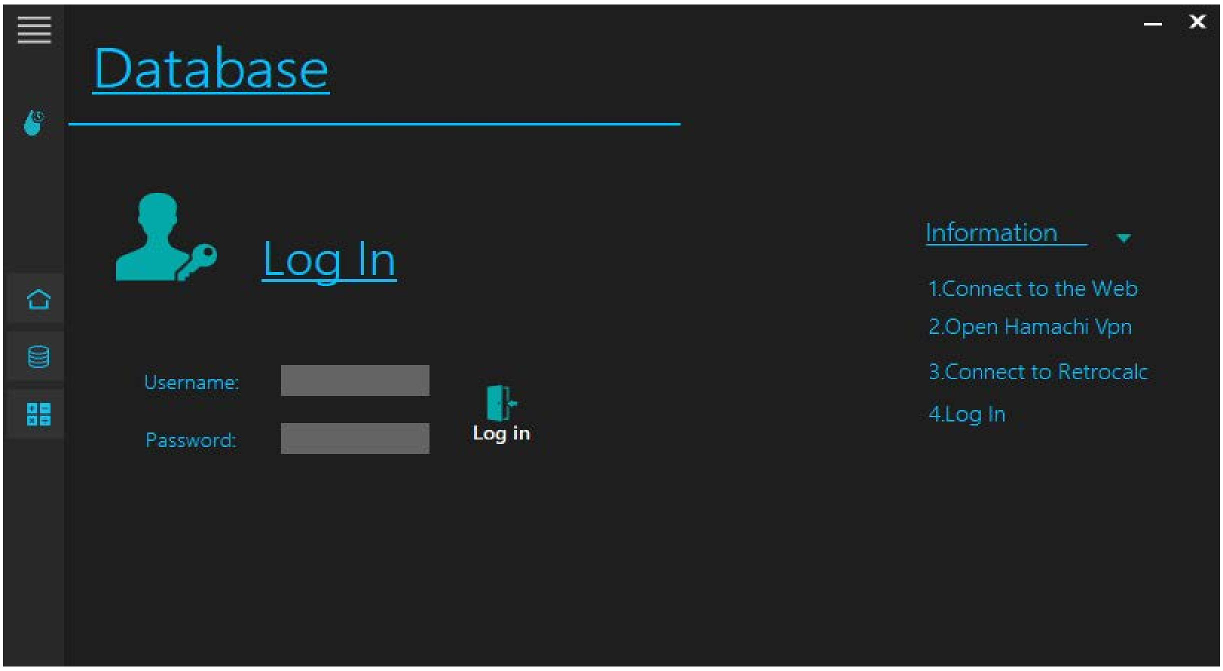

Figure 3.3: Database Tab

In this panel the user can log in to a database, which contains basic info about vessels traveling in Greek seas. The user has the option to access the database as admin, through an already registered username and password. When accessing the database as admin, the capability of deleting data from the database becomes available. On the other hand, if there is no such user registered, it is possible to enter the database with simple user privileges, with the difference that now the user cannot delete data.

The panel after the successful log in will look like the following.

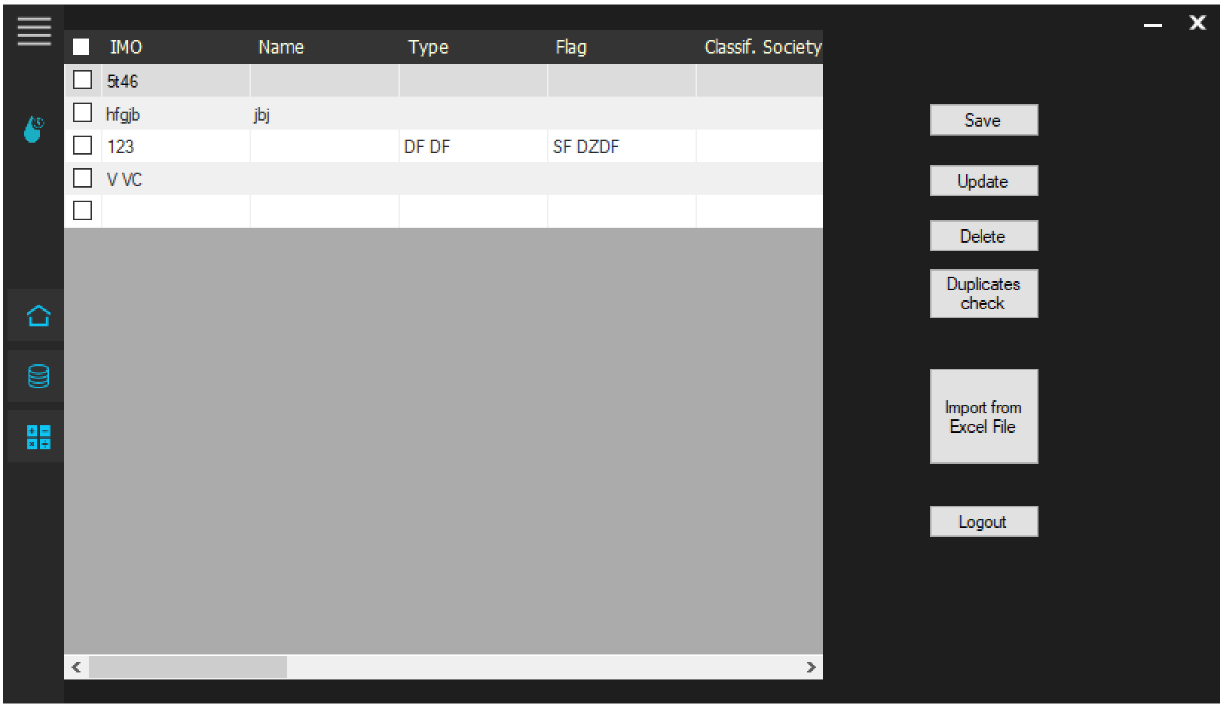

Figure 3.4: Database Manipulation Panel

From this panel the user can easily:

- Add new data
- Update existing data
- Delete data (admin privilege)
- Import bulk data from excel file
- Check for duplicate data

Database connection:

In order to connect successfully to the database, the server that the database was created in has to be online. Specifically, the connection to the server can be achieved through VPN connection. For the VPN connection the **LogMeIn Hamachi** Application is used.

LogMeIn Hamachi is a virtual private network (VPN) application that is capable of establishing direct links between computers that are behind Network address translation("NAT") firewalls without requiring reconfiguration. Prior to login the user will have to download and install **LogMeIn Hamachi** and connect to the **Retrocalcsql** server, which has been created in the application's platform. After this the connection will be successful come through in short time, depending from the online connections to the server.

3.2.4. Retrofit Calculator tab

When the user clicks the Home tab the following template becomes visible.

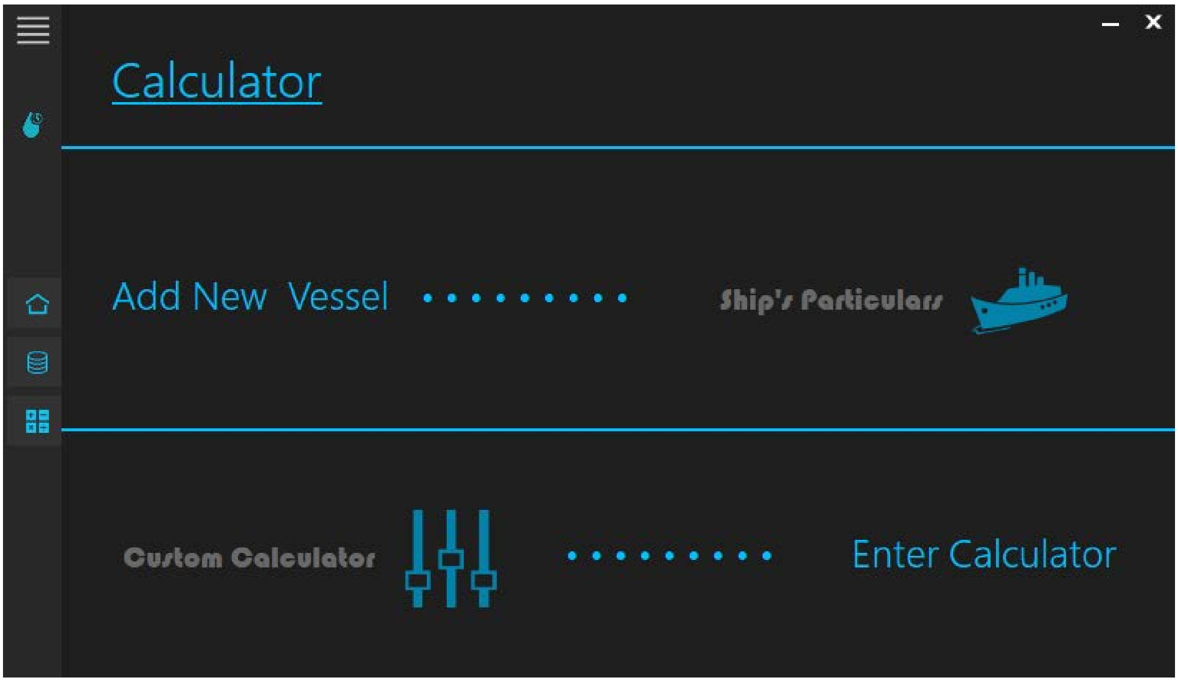

Figure 3.5: Calculator Tab

From this panel the user has access to two tools. With the first one he can easily Add New Vessel to the database. This feature is provided also by the, previously discussed, database panel, with the difference that now an individual visual tool makes the import of data easier, as can be seen below.

3.2.5. Add New Vessel Tool

Figure 3.6: Add New Vessel Form

With this tool the user is able to add data for a New Vessel to the Database. This way of import is recommended, if the user wants to enter a single vessel to the Database. In contrast, if more than one vessel needs to be imported in the database **the Import from Excel file Method** can be used.

The data to be added for each vessel are divided into three categories, the General, the Machinery and the Dimensions associated data.

The General data are data that are not required for any calculations, but are essential for the Vessel Archive. These data are:

- Name of the vessel
- IMO number of the vessel
- Type of the vessel (e.g. Bulk Carrier, Tanker etc.)
- Flag registry of the vessel (e.g. Greek flag, Panama flag)
- Class registry of the vessel (e.g. ABS, BV, NKK)

The Machinery related data contain the info that will be used for the calculation of the retrofit conversion of the vessel. This data are the only data used in the calculation process. These data are:

- Main Engine's Manufacturer

- Main engine’s Power
- Number of Main Engines
- Generator Engine’s Manufacturer
- Generator engine’s Power
- Number of generator engines

Again, here the Names of Manufacturers are inserter for filing purposes.

The Vessel’s Dimensions related data, consist of the Vessel’s principal dimensions and quantitative info. Similarly, these data are only imported for archive purposes and are:

- Length Over All
- Length Between Perpendiculars
- Breadth
- Depth
- Draft
- Deadweight
- Gross Tonnage

The user can insert data in the database by simply selecting the Save Button. If the user tries to insert a vessel that already exists in the database, an error will occur, informing the user. The check is done by comparing the given vessel’s IMO number, with each and every one of the already stored vessels in the database.

3.3. Custom Calculator Tool

This is the main tool of the program. Through this tool the user can accomplish a full analysis in a retrofit conversion of an existing vessel. Bellow the form of the Calculator tool is shown and will be analyzed thoroughly.

The Custom Calculator form is divided in five subcategories:

1. Machinery
2. Battery
3. Trip Info
4. Costs
5. Scenarios

Which will be presented in the following paragraphs.

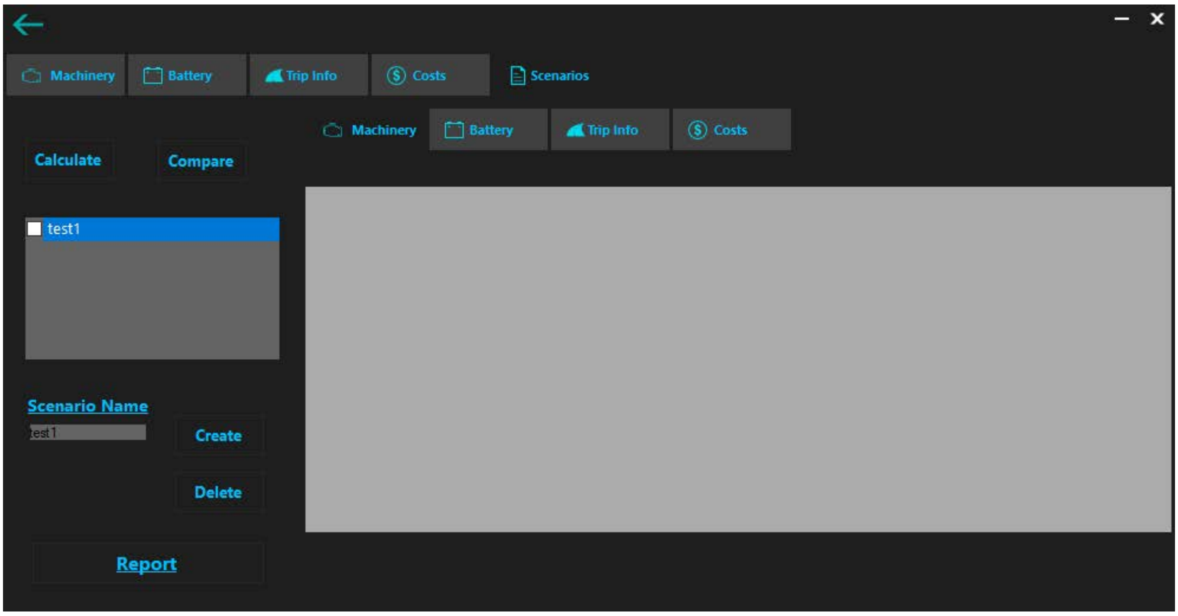

Figure 3.7: Scenarios Panel

From this panel the user can create a desirable scenario. The scenario has to be created, before the user starts adding the required data. The name of the scenario can be written in any language, but cannot consists only space characters. The user is able to create five different scenarios.

After the user has created the desirable scenarios, it is time to insert the required data for the calculation. This is done by selecting each tab from the tab menu and filing it with requested data.

3.3.1. Machinery

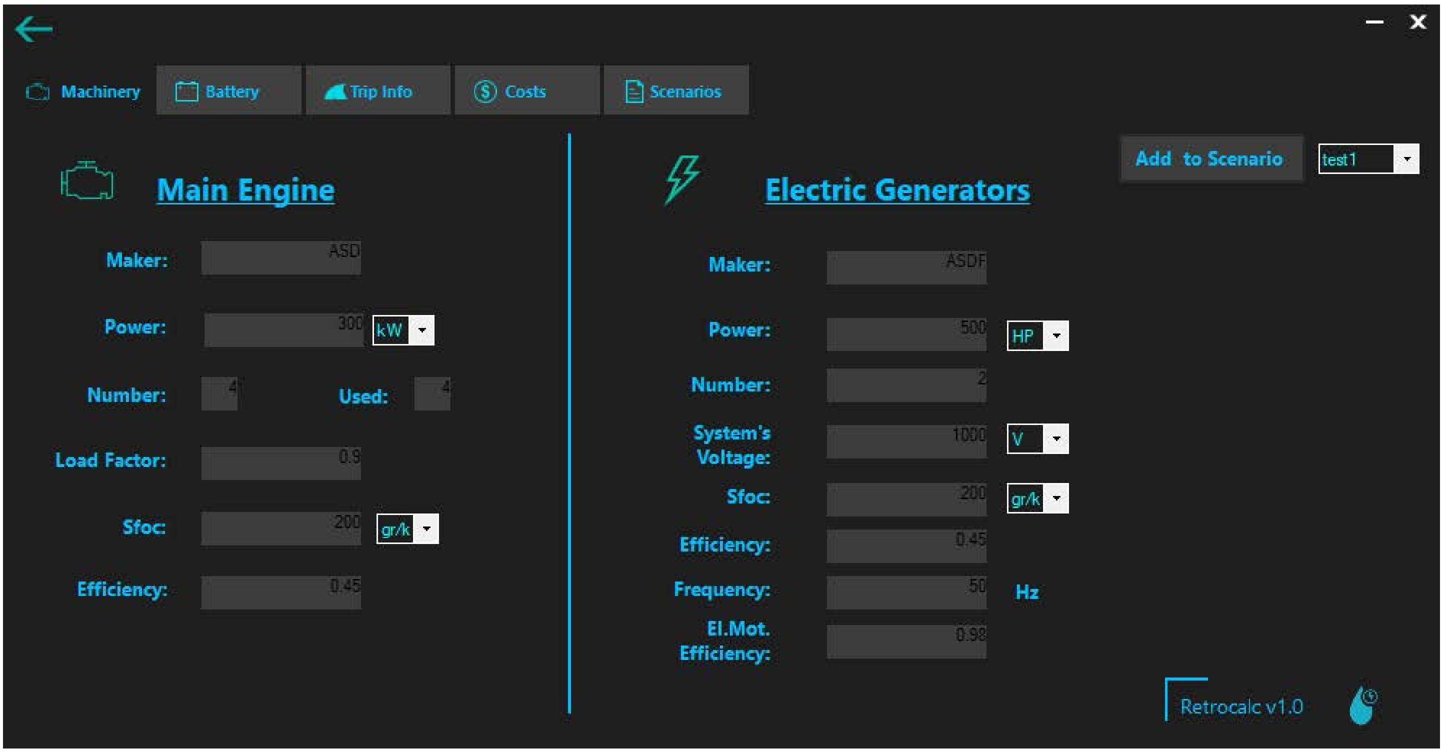

Figure 3.8: Machinery Panel

In this panel the user can enter data that are related with the machinery of the vessel. Specifically, he must complete the form by entering:

Main Engine:

- M/E Maker
- M/E Power and units
- M/E Number (total number, number of used engines)
- M/E Load Factor
- M/E Specific Fuel Oil Consumption and units
- M/E Efficiency

Generator Engine:

- G/E Maker
- G/E Power and units
- G/E Number (total number)
- System's Voltage Magnitude

- G/E Specific Fuel Oil Consumption and units
- G/E Efficiency
- Generated Current's Frequency
- El. Motor's Efficiency

After, the user has completed the form, can easily import the data by selecting the scenario from the upper right corner of the form and press the **Add to Scenario** button.

3.3.2. Battery

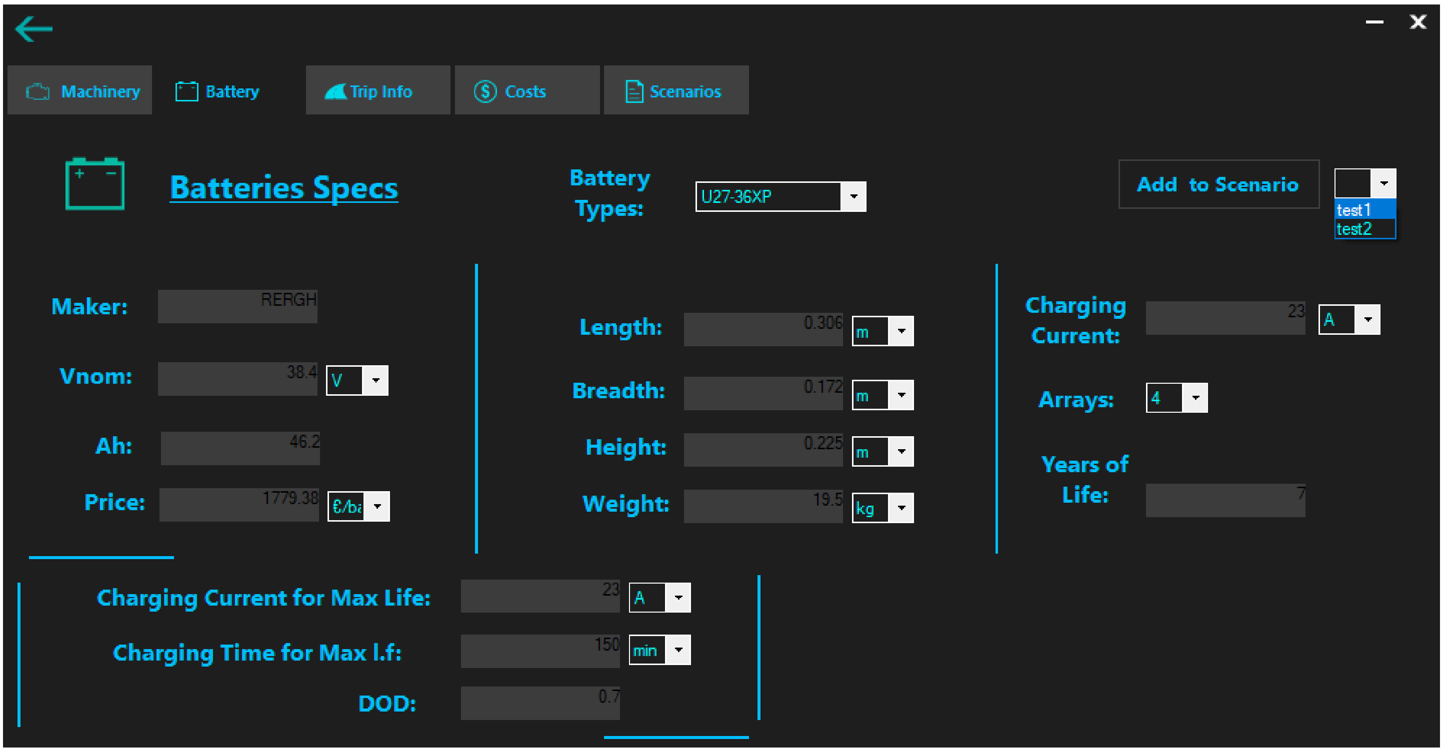

Figure 3.9: Battery Panel

Similarly, with the Machinery form, here the user can enter the required data, related to the battery system, which will be installed. The required data are:

- Maker of the Batteries
- Nominal Voltage of the Batteries
- Capacity of the Batteries, expressed in Ampere Hours
- Cost of the Batteries (can be entered by unit or by kWh)
- Battery's Dimensions and Weight
- Charging Current of the Batteries (recommended value)
- Number of Arrays, in which the batteries will be divided
- Years of Life for the batteries
- Charging Current, recommended for Maximum life

- Charging Time, nominal for full charge
- Depth of Discharge Allowed

After, the user has completed the form, can easily import the data by selecting the scenario from the upper right corner of the form and press the Add to Scenario button.

3.3.3. Trip Info

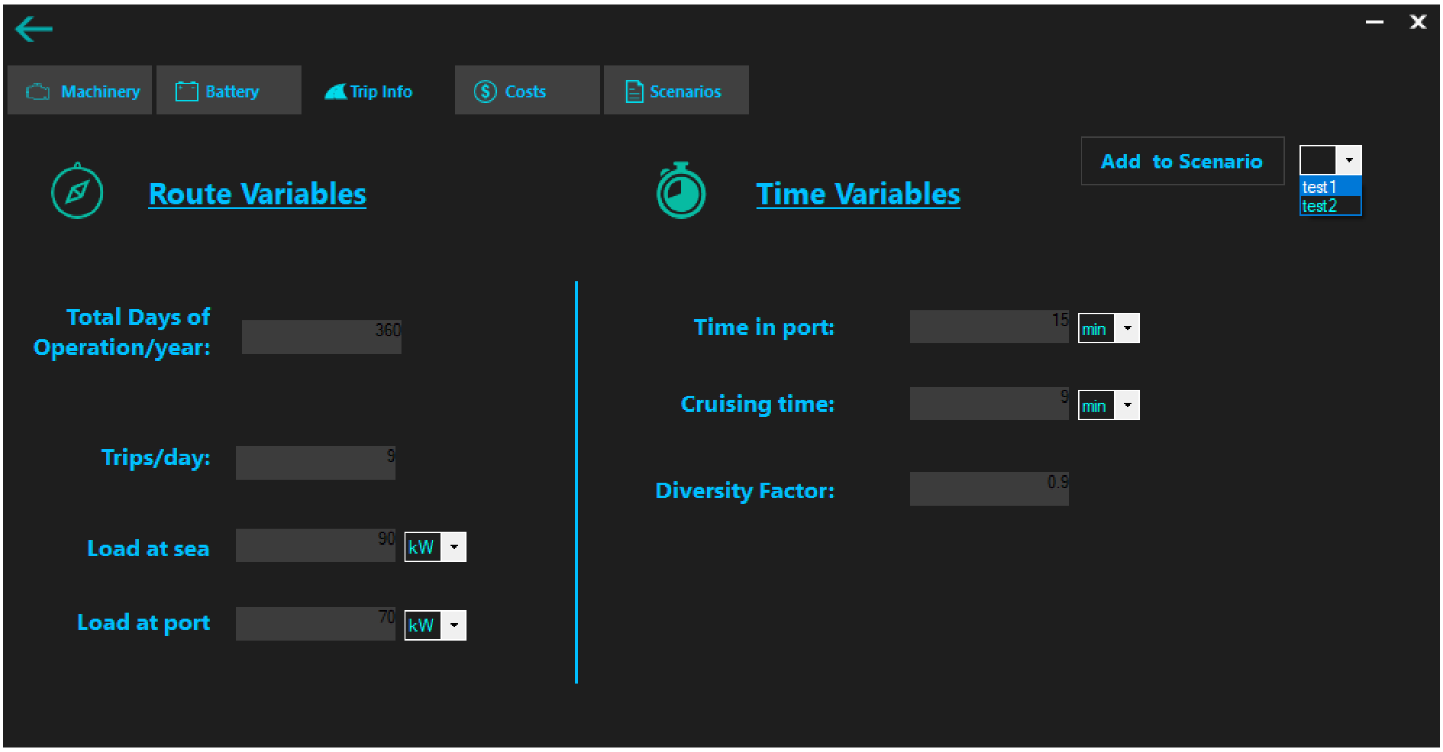

Figure 3.10: Trip Info Panel

Through this panel, the data related with the trip of the vessel can be inserted. The data are divided between Route Variables and Time Variables.

Route Variables:

- Total Days of Operation in one year of operation
- Total Trips completed in one day of operation
- Load required during sailing
- Load required during birth

Time Variables:

- Time spent in birth
- Sailing Time

- Diversity Factor

It should be noted, that the available **time at birth** is the main factor, which affects the results of the analysis.

After, the user has completed the form, can easily import the data by selecting the scenario from the upper right corner of the form and press the Add to Scenario button.

3.3.4. Costs related Info

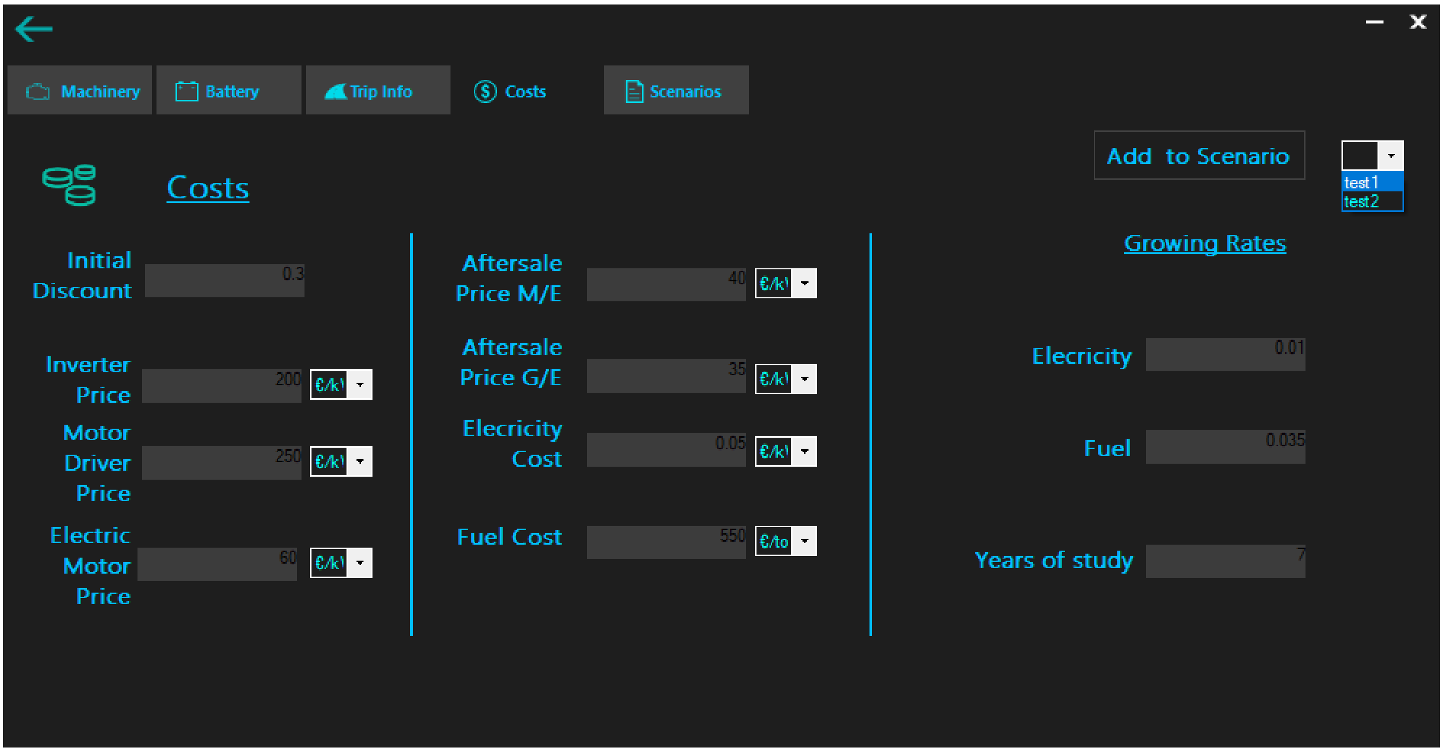

Figure 3.11: Costs Panel

Completing this form, the cost variables for the required equipment are added. Furthermore, the price of sale for the used machinery is asked for both main engine and generator engines. Also, the user is asked to add the cost per kWh and the cost per ton of fuel. Last, if a long-term analysis is needed, growing rates of electricity and fuel costs should be added for the total years.

After, the user has completed the form, can easily import the data by selecting the scenario from the upper right corner of the form and press the Add to Scenario button.

3.3.5. Scenarios Tab

When a scenario has been created and filled with the required data, the user can select it by the listbox seen in the left side of the panel. By selecting the desired scenario, the inserted data are shown in the table at the right side as shown below:

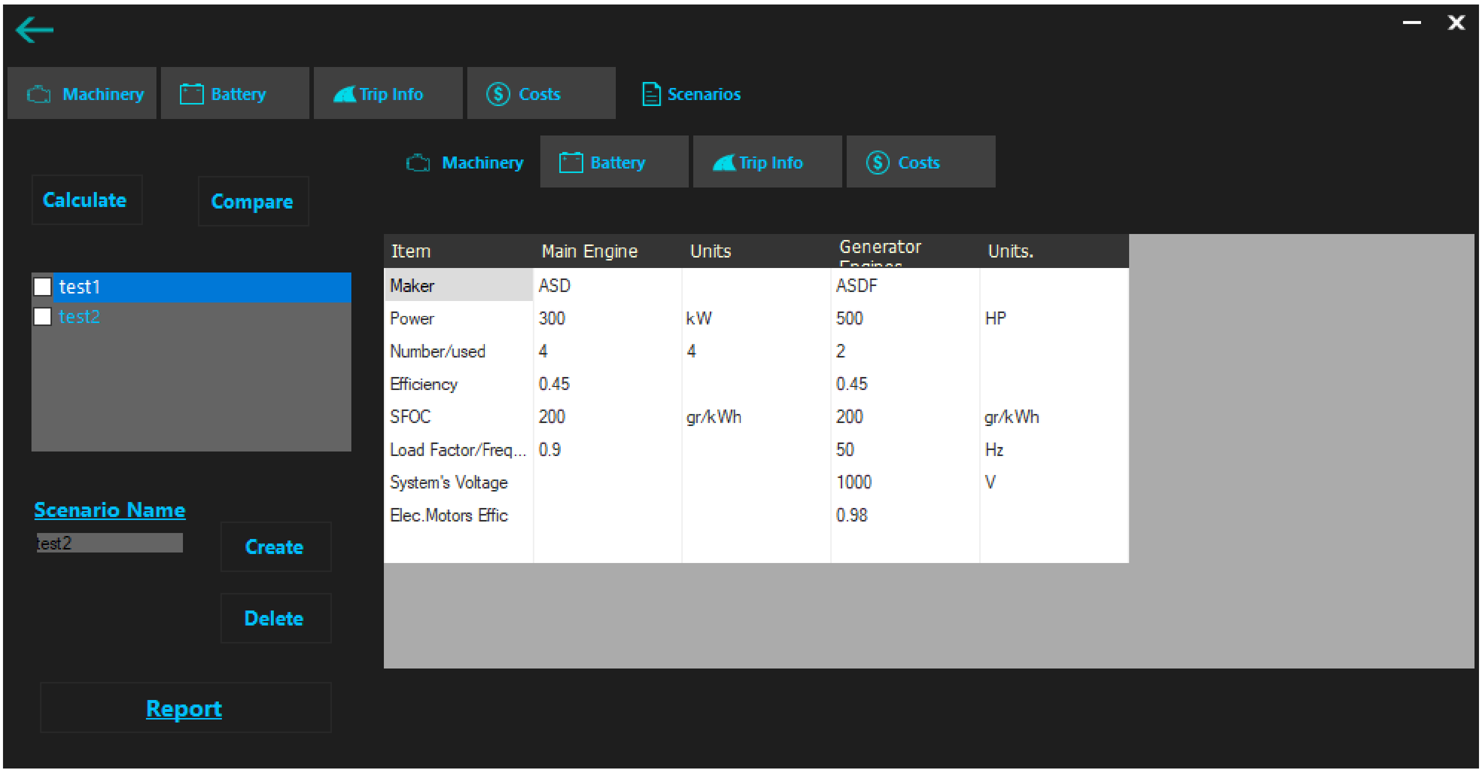

Figure 3.12: Scenarios Panel (after data input)

By using the tab menu above the table, the user can navigate through the inserted data, to complete a last check before proceeding with the calculation. After, the check is completed, the user can proceed to the calculation by pressing selecting the desired scenario and pressing the calculate button. When the calculate button is pressed, the program stores the calculated data in a table, in order to use it where it is requested.

3.4. Calculation Process

The Calculation Process initializes when the user presses the Calculate button. The analysis of the process is shown in the following flow chart.

Figure 3.13: Calculation Procedure

3.4.1. Calculator class

This class will be presented, with respect to the different functions that it contains.

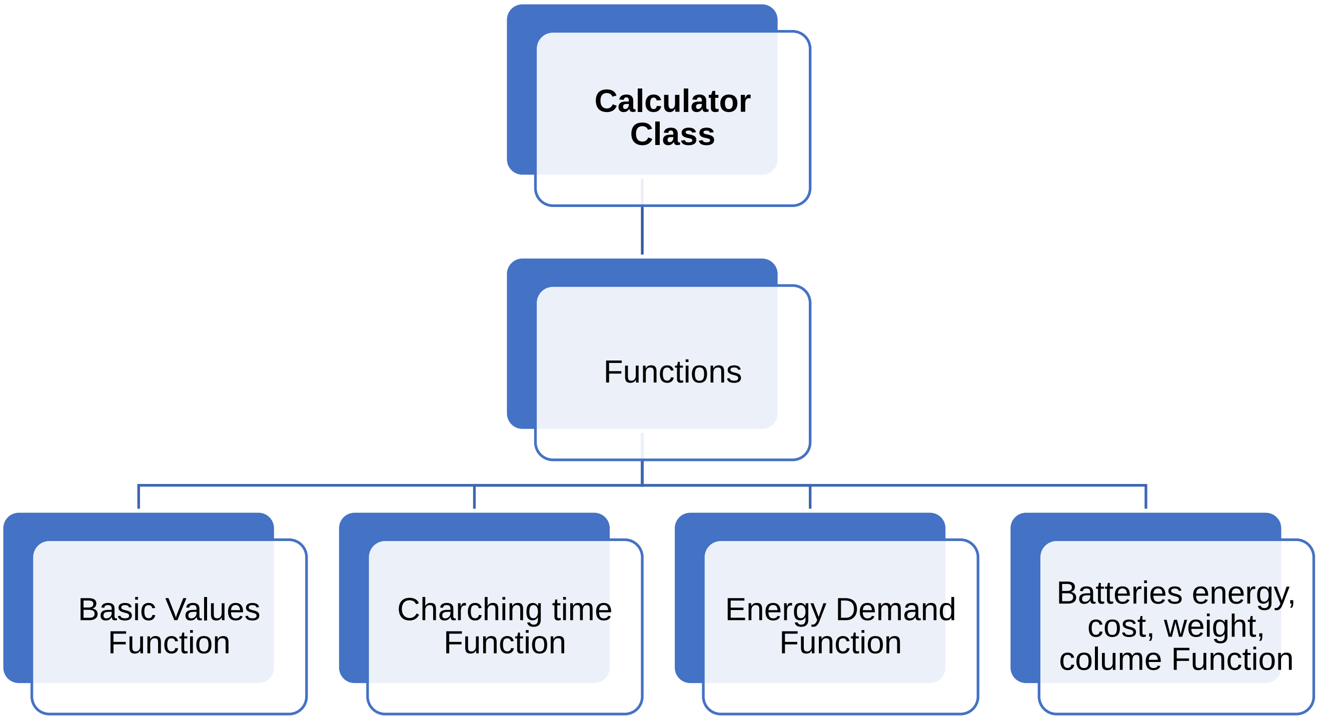

Figure 3.14: Calculation Class Hierarchy Tree

Basic Values Function:

With this function, basic values that are not influenced by the charging time are calculated.

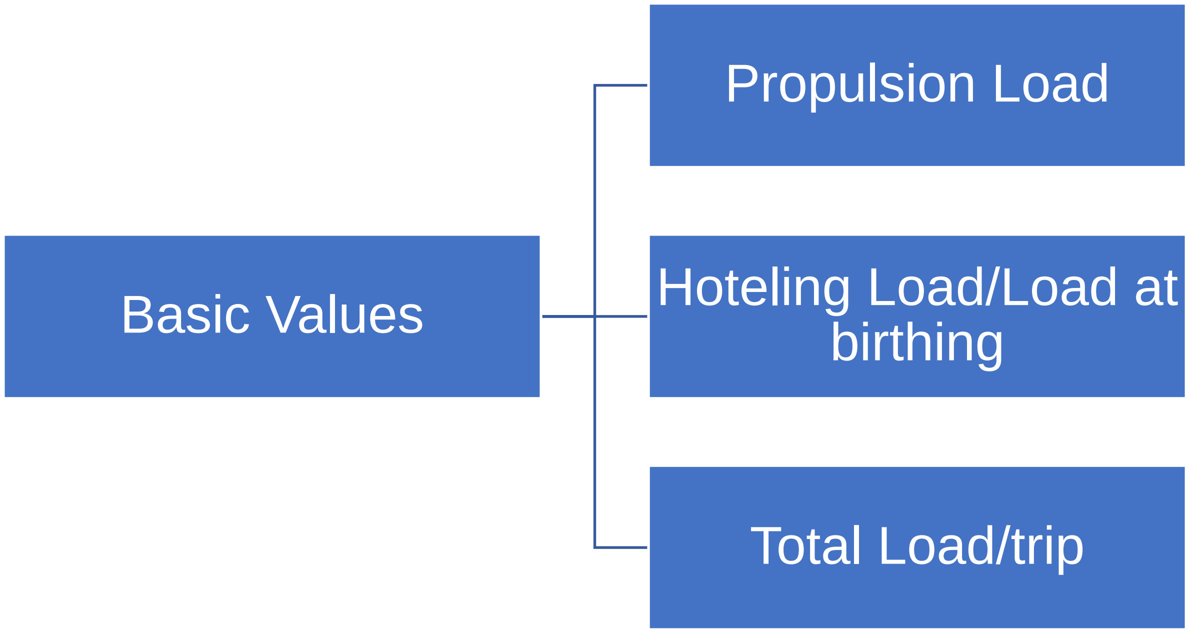

Figure 3.15: Basic Values Function Return Values

Charging time available Function:

This function is the heart of the program, as it calculates the charging time available in each port, with respect to the remaining capacity of the battery system.

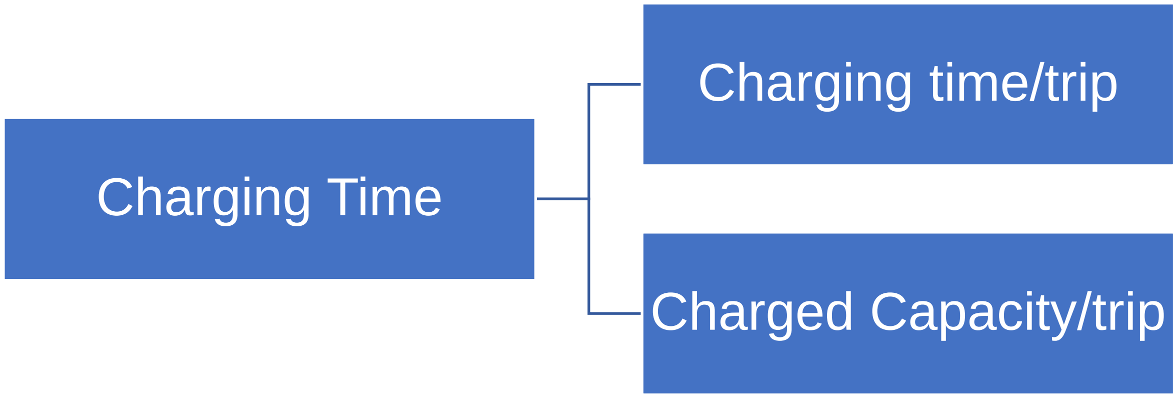

Figure 3.16: Charging Time Available Function Return Values

Energy Demand Function:

The Energy Demand Function, calculates and returns the energy demand.

Figure 3.17: Energy Demand Function Return Values

Batteries energy, cost, weight, volume Function:

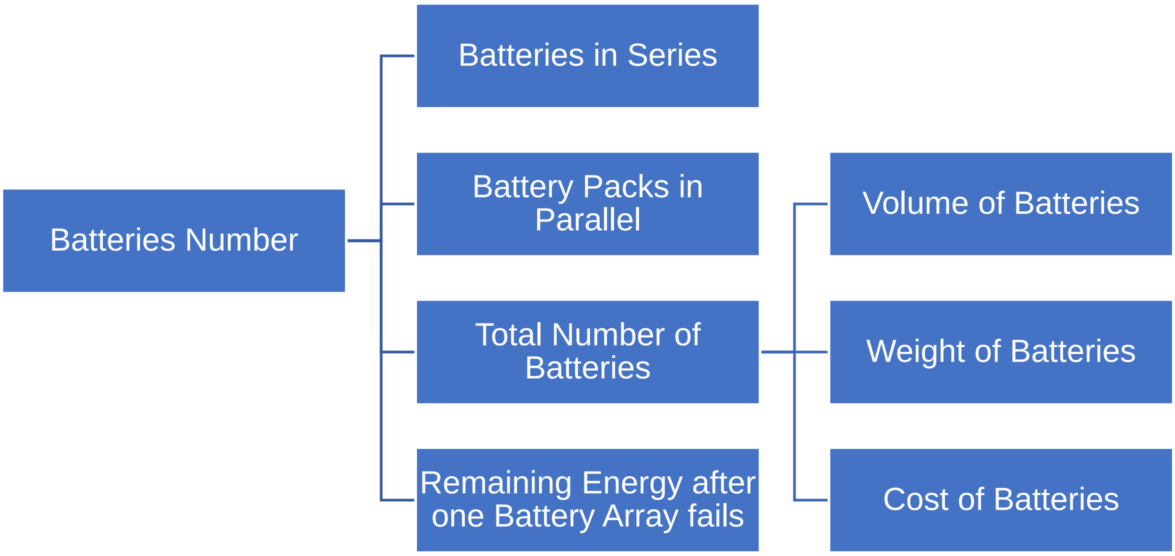

Figure 3.18: Batteries Values Function Return Values

Within these functions, the total number of batteries required is calculated. First, the number of batteries, in series and the number of packs connected in parallel, are calculated. Next, the total number of batteries installed is calculated. After, the battery system's volume, weight and cost are calculated and returned in the main program. Then the remaining energy, after failure of one array, is calculated.

3.4.2. Operation and Maintenance costs Class

The next step is the calculation of **The Operational and Maintenance costs**. This procedure is done by the **Opermaintcosts Class**.

This class will be presented, with respect to the different functions that it contains. These are:

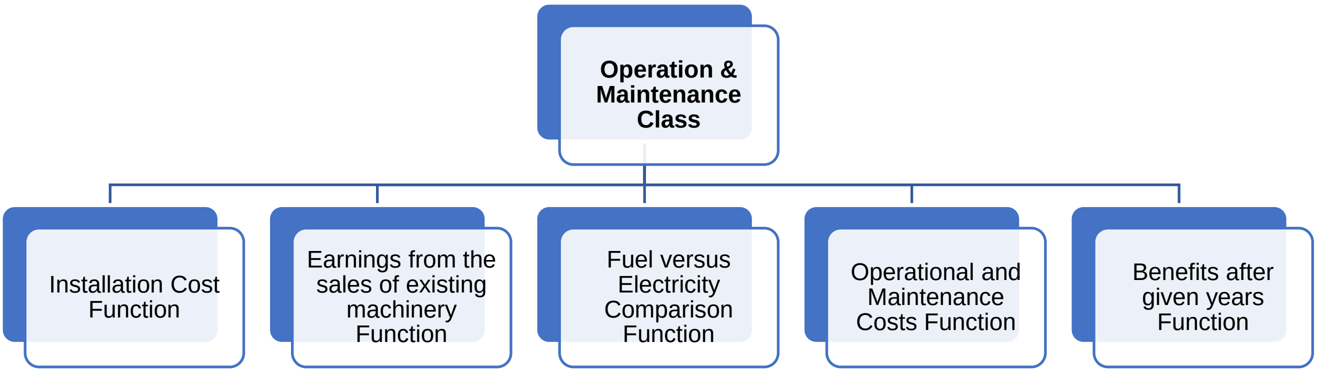

Figure 3.19: O&M Class Hierarchy Tree

Installation Cost Function:
In this part, the initial cost for the battery system is calculated. This includes, the cost of the batteries and the cost of the electrical components required for the conversion.

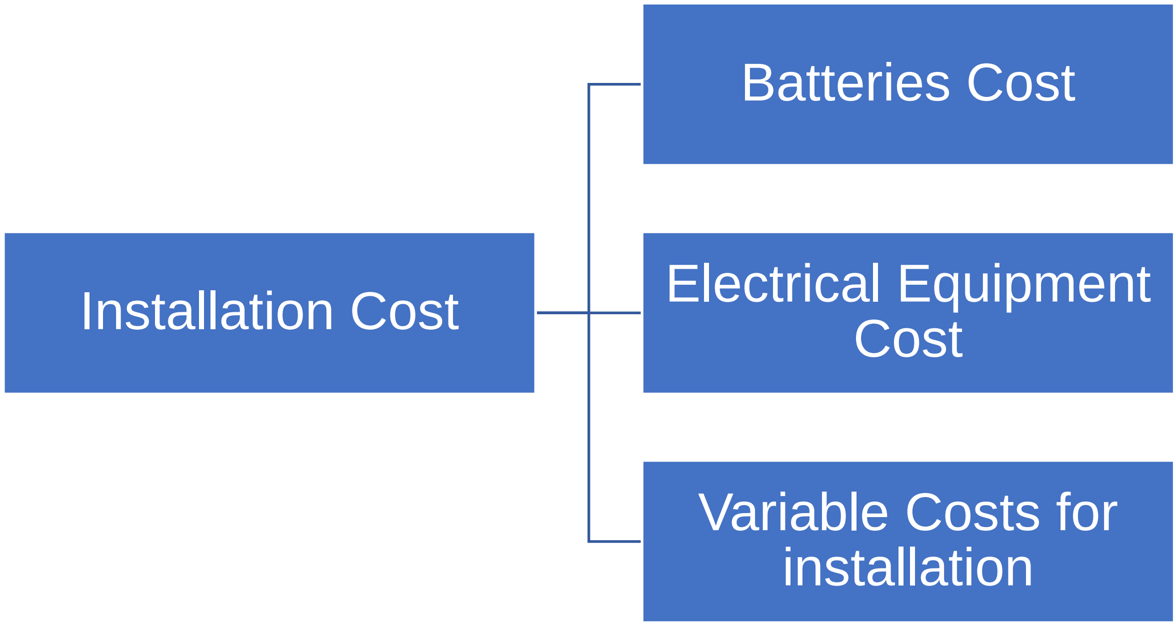

Figure 3.20: Installation Cost Function Return Values

Earnings from the sales of existing machinery Function:
This function calculates the earnings from the sales of the installed machinery. The installed machinery includes the main engines and the generator engines.

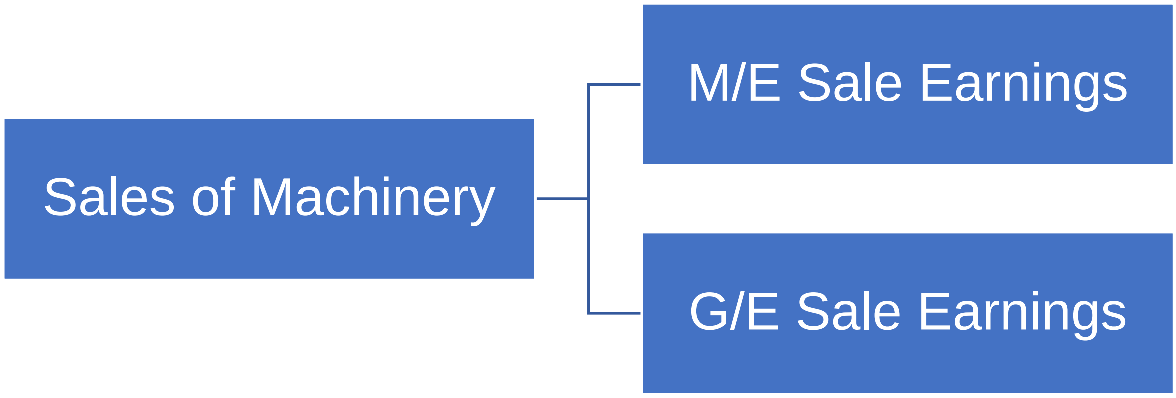

Figure 3.21: Sales of Machinery Function Return Values

Fuel versus Electricity Comparison Function

Through this procedure, the total costs of fuel and electricity, needed for the powering of each system correspondingly, are calculated for the length of one year. The results will be used for the calculation of the total benefits.

Figure 3.22: Fuel vs Electricity Function Return Values

Operational and Maintenance Costs Function:

Here the cost for the maintenance of each system is calculated. The results will be used for the calculation of the total benefits.

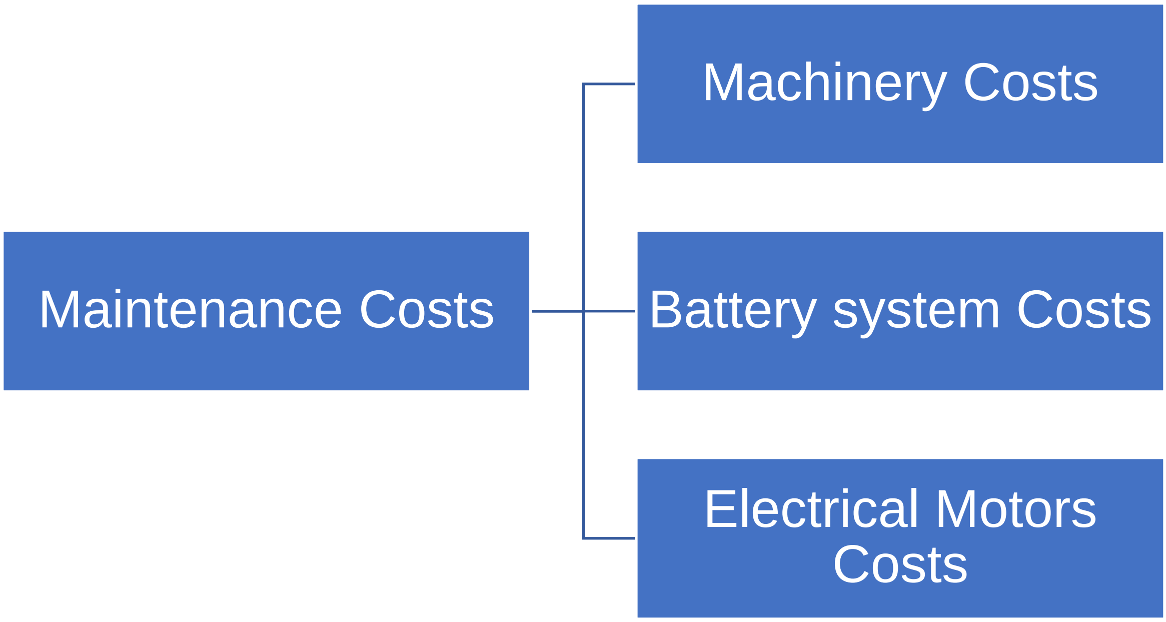

Figure 3.23: Maintenance Costs Function Return Values

Benefits after given years Function:

With the use of this function, the program calculates the fuel, maintenance and total benefits in a given period of time. The depth of time, for the analysis, is given by the user. It may be noticed a significant decrease in the benefits, at the years when the battery's life expires and needs to be replaced.

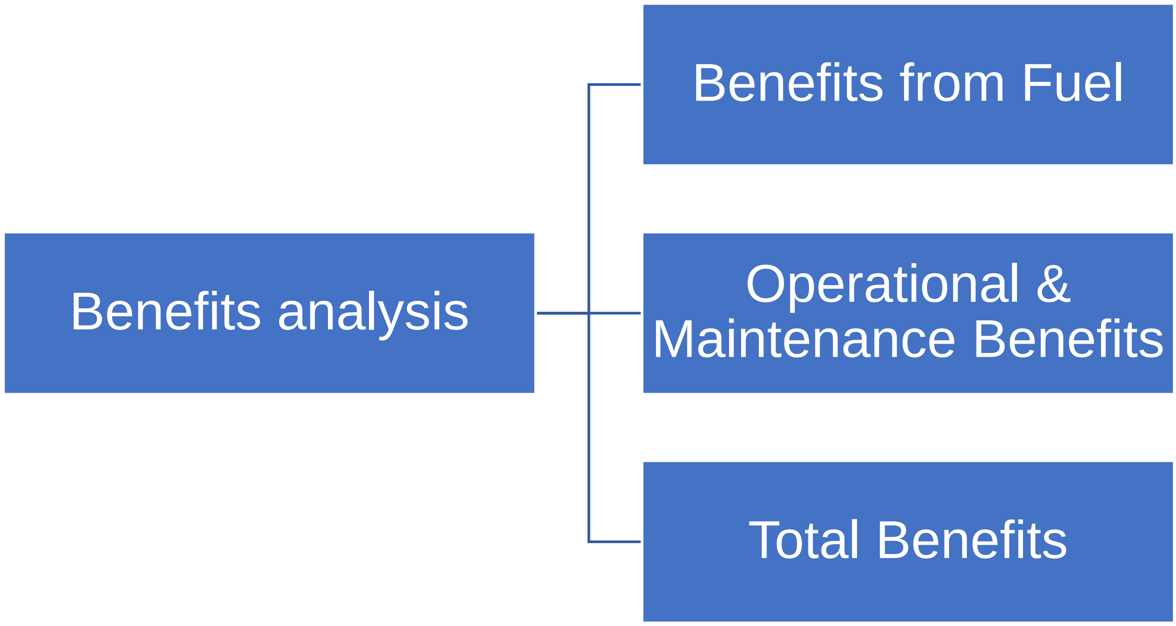

Figure 3.24: Benefits Analysis Function Return Values

3.4.3. Emissionscalc Class

The final step is the calculation of **The Externalities costs**. This procedure is done by the **Emissionscalc Class**.

This class will be presented. Only one function is included in this class.

Emission costs Function:

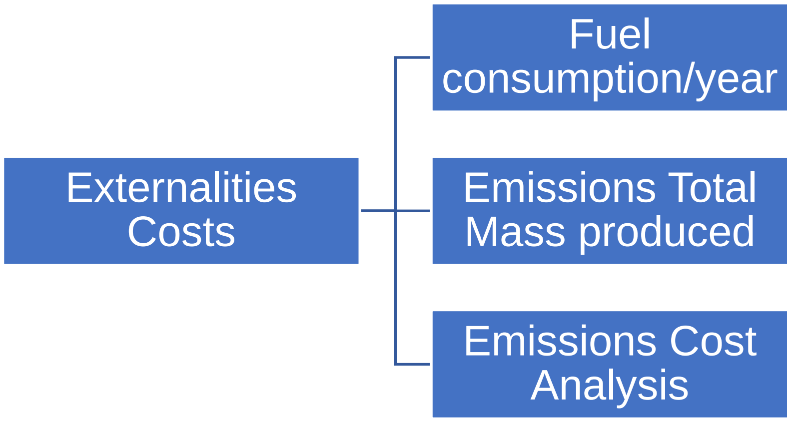

Figure 3.25: Externalities Costs Function Return Values

This function receives as input the loads and mass of fuel consumed by the main engine and the generator engines during birthing and sailing. With these data proceeds to the calculation of the mass of the produced emissions and later to the calculation of the total cost of the emissions saved.

3.5. Report form

After, the desired scenarios have been created and filled with the required data, the user can enter the **Report Form.** In this form, a visualization of the analysis results is presented. The form is shown below

Figure 3.26: Report Form Preview

The form is divided into three basic sections: **Individual Graph Section, Comparison Graph Section and Values Section.**

3.5.1. Individual Graph Section

This section contains two separate graphs, one for the visual presentation of batteries related info and the other with financial info.

In the first diagram the number of batteries with respect to time of charging available in port is presented. In addition, the total cost of batteries is related with the charging time. Through this graph the user can understand the effect of the charging time available, to the total number of batteries needed, hence the initial cost of the installation. **The time available in port** is expressed in **minutes** and **the cost of the batteries** is expressed in **Euro (€)** currency.

The second graph visualize the benefits of the conversion, against the conventional propulsion system, with respect to the years count of the study. **The total benefits** are expressed in **Euro (€)** currency.

The user can select which scenario he wishes to display. This option is provided by the drop-down menu tool in the top part of the form.

3.5.2 Comparison Graph Section

In this section, two graphs are presented. The first one displays the total benefits next to the fuel saving benefits for each scenario. The graph type chosen is a Column Graph, in which the user can easily distinguish the benefits between different scenarios. Accordingly, in the Y-Axis the amount of money is displayed in **Euro (€)** currency and in the X-Axis the name of each calculated scenario is shown.

The second graph displays the total benefits form each scenario, with respect to the charging time available in port. The graph type chosen is a Line Graph. From these diagrams the user is able to compare the benefits of the calculated scenarios and the effect of available charging time on them. In fact, this chart combines all the individual charts displayed in the bottom left area.

3.5.3. Values Section

At the bottom side of the form, the calculated values are displayed. These values correspond to each calculated scenario and are refreshed every time the user selects another scenario from the drop-down menu.

The Values Section is divided into 4 Sub-Sections:

- Batteries related section
- Trip related section
- Emissions related section
- Benefits related section

Batteries related section

Here the calculated values related with the batteries installed are shown. These Values are:

- Total Number of Batteries
- Total Weight of Batteries
- Total Volume of Batteries
- Total Cost for the Batteries

These values represent the most important data of the analysis results. **The total number**, **the weight** and **the volume** of batteries are shown, so that an approximation of the size needed for the installation becomes known to the user. Furthermore, **the initial cost** of the batteries is displayed, as one of the main info that determine the viability of the conversion to an all-electrical vessel.

Trip related section

In this part, the charging time available in each port is displayed. As previously mentioned, the value of time is the most crucial value in this analysis, due to the methodology used. Furthermore, the user is able to select a time value and preview the related calculated values for Batteries, Emissions, Benefits.

Emissions related section

Furthermore, to other benefits, the conversion to an all-electrical vessel contributes in the reduction of the produced emissions, during operation. The two related values are:

- Emissions Total Mass
- Emissions Total Cost

The Mass produced, refers to the conventional system, that uses fossil fuels. **The Mass** value is expressed in **metric tons**. Furthermore, through a cost-analysis the benefits form the reduction of the emissions was calculated and is presented here. The cost benefit is expressed in **Euro (€)** currency. As one of the main benefits of the all-electrical vessels, the benefits from the reduction of the produced emissions should be displayed.

Benefits related section
Here the main benefits, created by the use of batteries, are presented. These are:

- Benefits from not using fossil fuel
- Total Benefit related to operational and maintenance costs
- Both of the Above Benefits Combined

The above benefits, that are presented, are calculated for the lifespan of the project. In other words, only the total benefits produced until the last year of the study are shown. All the values, displayed in this section, are displayed in **Euro (€)** currency. The user can easily compare and retrieve a state for the size of the different benefits.

3.6. Additional tools and features

Due to the nature of this program, additional tool and utilities were added compared with the basic methodology used in Mr. Bakirtzoglou’s thesis. These are:

1. Vessel Database
2. Batteries Default Values per Type
3. Comparison Tool
4. Default values, for quick analysis

3.6.1. Vessel Database

The Database utility was shortly presented in previous chapter. Here an extended presentation will be given.

The Database was created with the use of Microsoft SQL Server Management Studio 2017© application. The database inside the application environment is shown in the following figure.

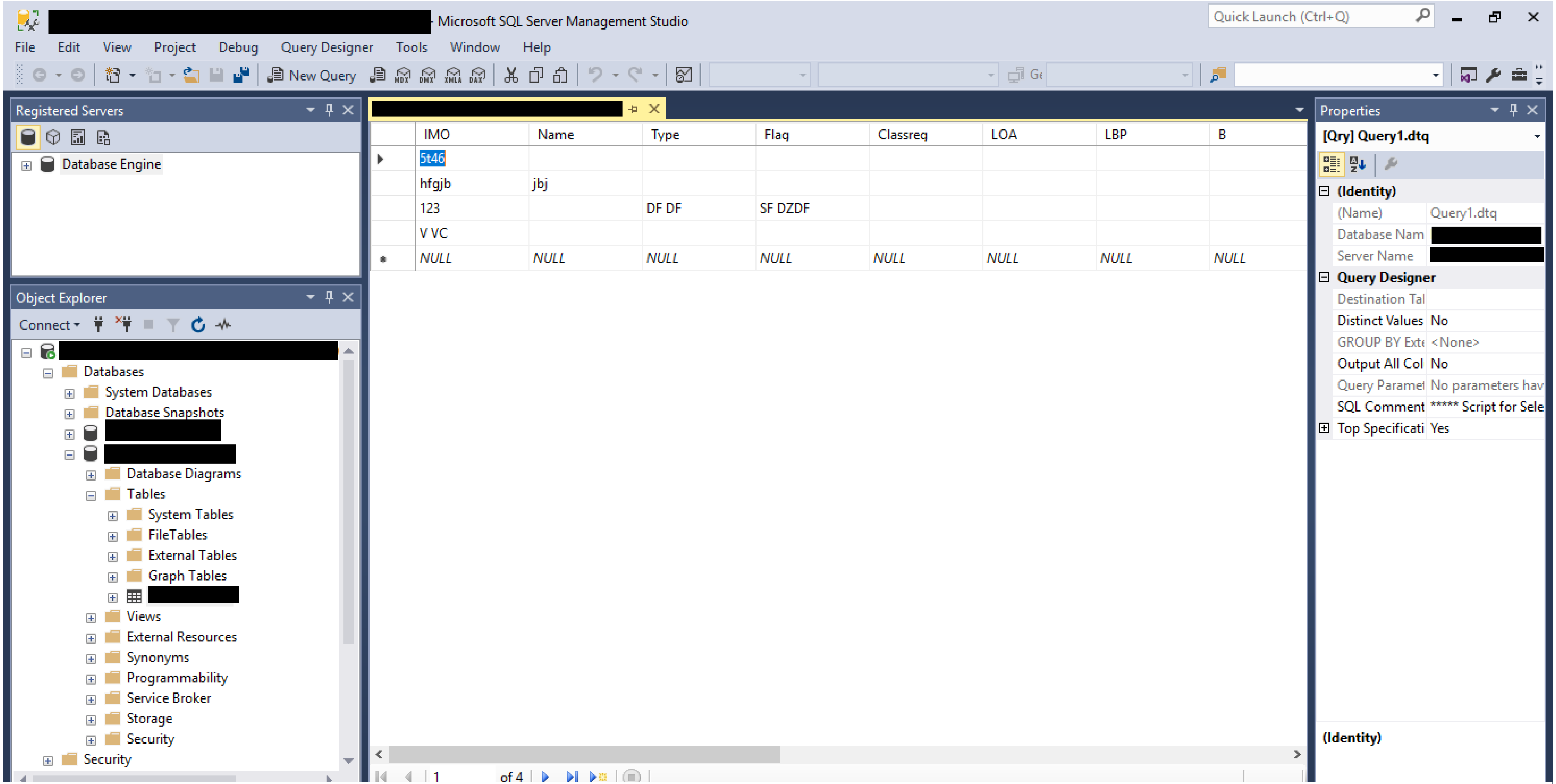

Figure 3.27: Microsoft SQL Server Management Studio 2017© Application

As it is shown above, the database can be modified from the Microsoft SQL Server Management Studio 2017© application environment also. In contrast, with the database tool that was mentioned in the previous chapters, only the admin of the database (and owner of the main server) can manipulate the data from this environment. Here the admin has the following privileges:

- Add Data to the database
- Delete Data from the database
- Read Data from the database
- Change Other users' privileges over database
- Create new Data tables
- Edit create Data tables

In addition to the Microsoft SQL Server Management Studio 2017© application, another application is used, which controls the state of the server. This application is the SQL Server 2017 Configuration Manager©. An example picture of the application is presented below.

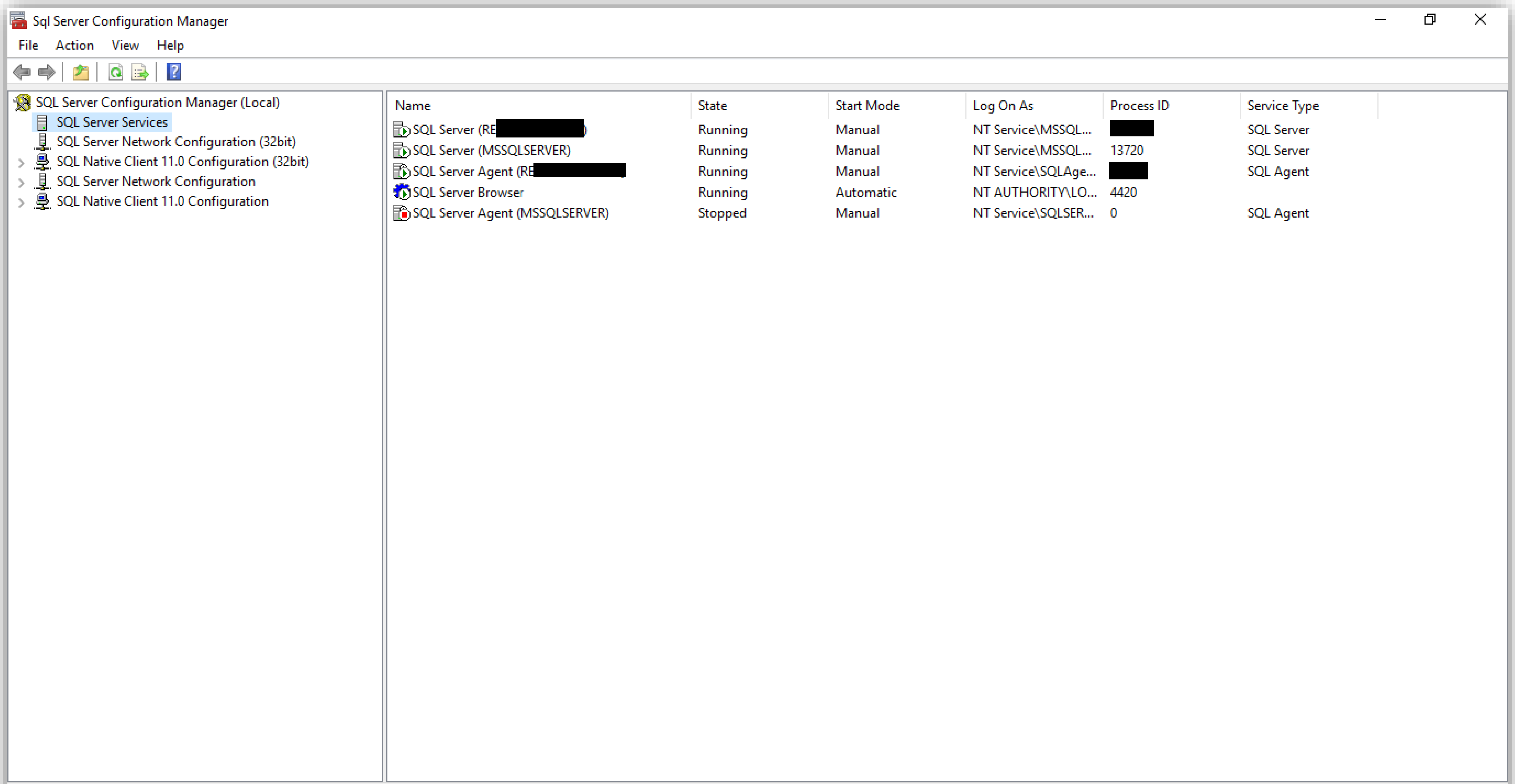

Figure 3.28: SQL Server 2017 Configuration Manager © Application

Using this tool, the admin is able to control the server's availability. The SQL Server 2017 Configuration Manager© supplies the user with the following features:

- Set Server's state to Online
- Set Server's state to Offline
- Set Server to open automatically with the star of the host-computer
- Schedule the desired time for the Server to go Online/Offline

It must be noted, that if the server is disabled from this tool, then it is not possible to enter the database from any other tool or environment.

The database is equipped with a Login database, in which the usernames and passwords are stored. For security reasons, the data stored in the Login Database are encrypted before the input in the system. Furthermore, new users can be added to the database only by the server admin, who owns a specifically designed tool for this reason. The process of encryption of the data will not be presented in this thesis, for obvious reasons.

3.6.2. Batteries Default Values per Type

Another useful tool was added to assist the user in the input of the required data for the calculation. Due to the fact, that the user may not have easy access to technical specifications about the battery's models, a tool that provides prestored models for the batteries is installed in the calculator process.

The tool can be accessed by a drop-down menu in the battery panel, inside the calculator form. From within this menu the user is able to select the desired type of batteries to equip his vessel. This feature is presented in the following figure:

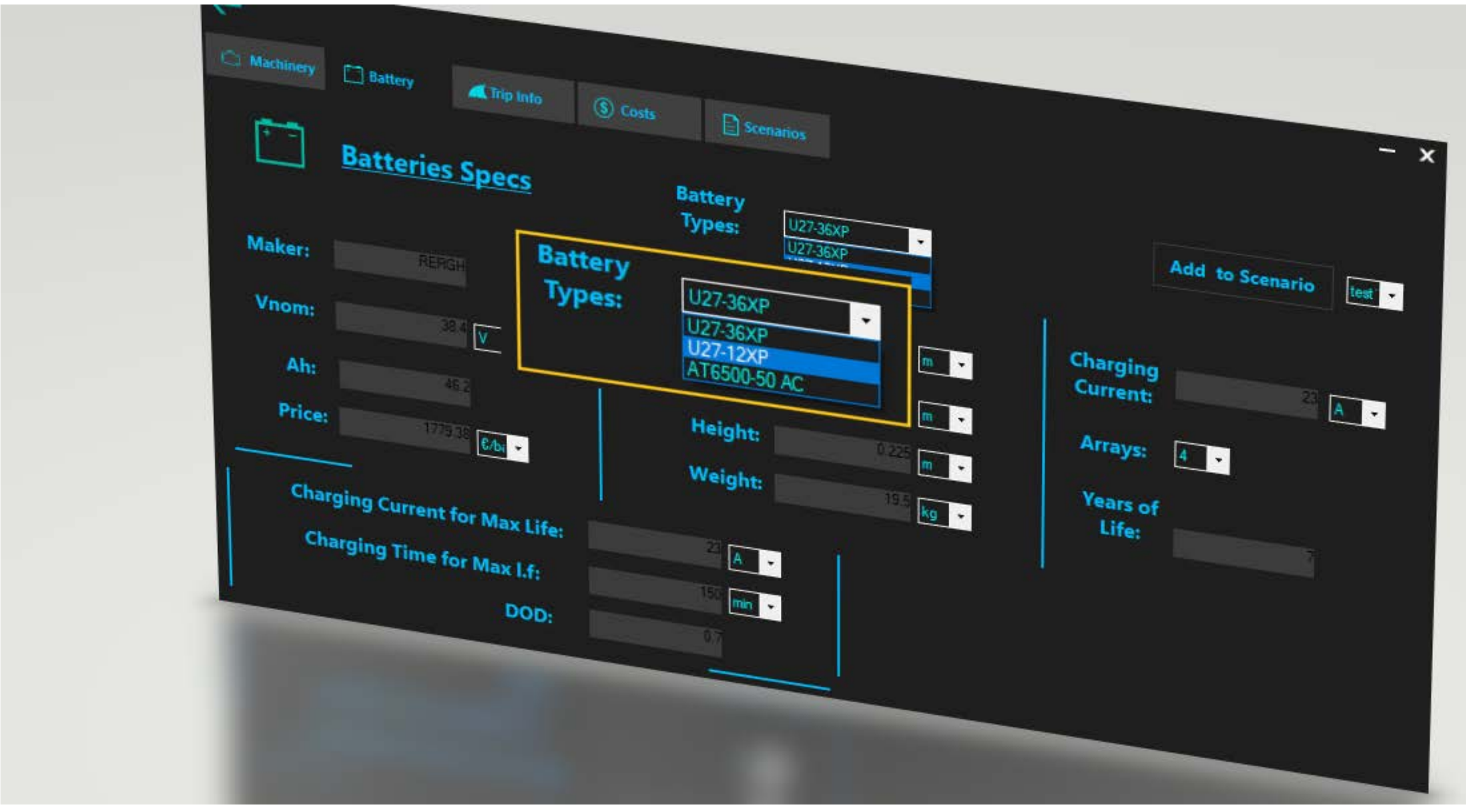

Figure 3.29: Battery Default Types

The program comes with three predefined battery types:

- U27-36XP
- U27-12XP
- AT6500-50 AC

But in later versions, more types are to be added. Furthermore, a tool which will provide the user with the ability to create and store new types of batteries, will be added.

3.6.3. Comparison Tool

A tool, with which the user is able to compare different scenarios of analysis, is installed inside this program. The control for this tool is inserted in the scenarios panel of the Custom calculator form.

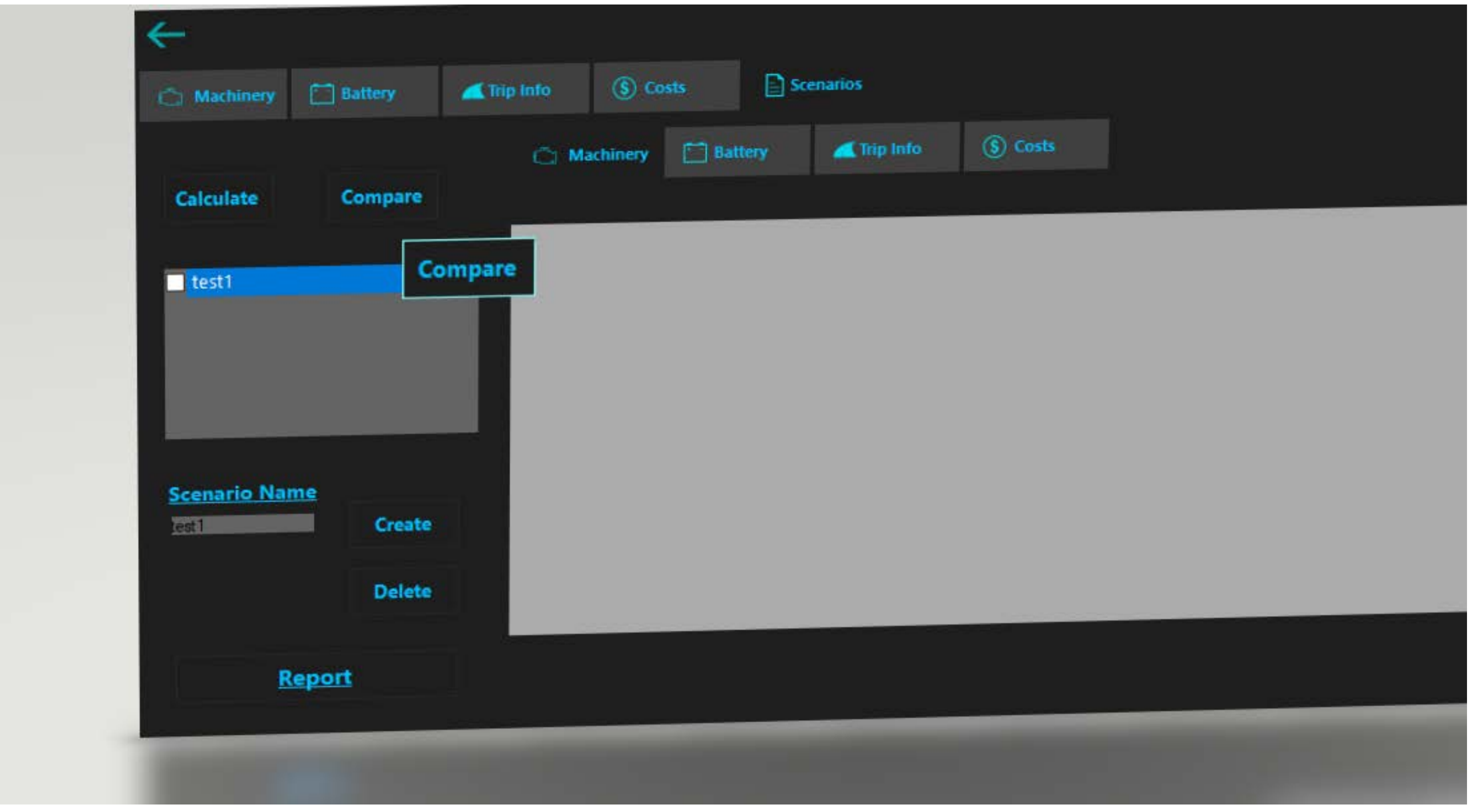

Figure 3.30: Compare Scenarios Feature

When the user presses the **Compare** Button, the scenarios that are checked within the listbox, are stored in a table, which will be used for the visualization of the data in the report form. In other words, the user will not notice any changes in the UI, but he will inform the program of which scenarios he wishes to compare. The comparison process, as described above, can also be initialized by pressing the **Report** button.

The user will see the results of the comparison in the **Report** form. The form after the comparison will look like the figure bellow.

Figure 3.31: Report Form, Comparison Section

The comparison graphs display the different benefits between the selected scenarios, as was discussed in previous chapters.

In conclusion the comparison tool forms a simple, yet powerful, feature that enables the user to compare benefits between different scenarios.

3.6.4. Default values, for quick analysis

In order that the user saves time during data input, default values for each section are provided. The user can simply override the default values and enter new ones. Furthermore, there are default values for the units, where required. These values are contained in drop-down menus accordingly. If the user selects the drop-down menu, without selecting a value, automatically the default value will be inserted.

The default values feature was added to the program, to assist the user in lack of required data situation and speed up the process of data input. The default values can be seen in previous figures in 3.3.1. Machinery ,3.3.2. Battery etc.

4. Results

In this chapter the results of the program for a double end vessel will be presented. This vessel has already been calculated in Mr. Bakirtzoglou's study, so the results are already known. Hence, they will be used for the verification of the program.

4.1. Input Data

4.1.1. Perama-Salamina scenario

The first scenario is retrieved by Mr. Bakirtzoglou's thesis and will be called "Perama-Salamina", from the name of its route. This study is been executed for a Double-Ended Ferry Vessel.

The Input Data for this vessel are shown in the following pictures:

Machinery Data

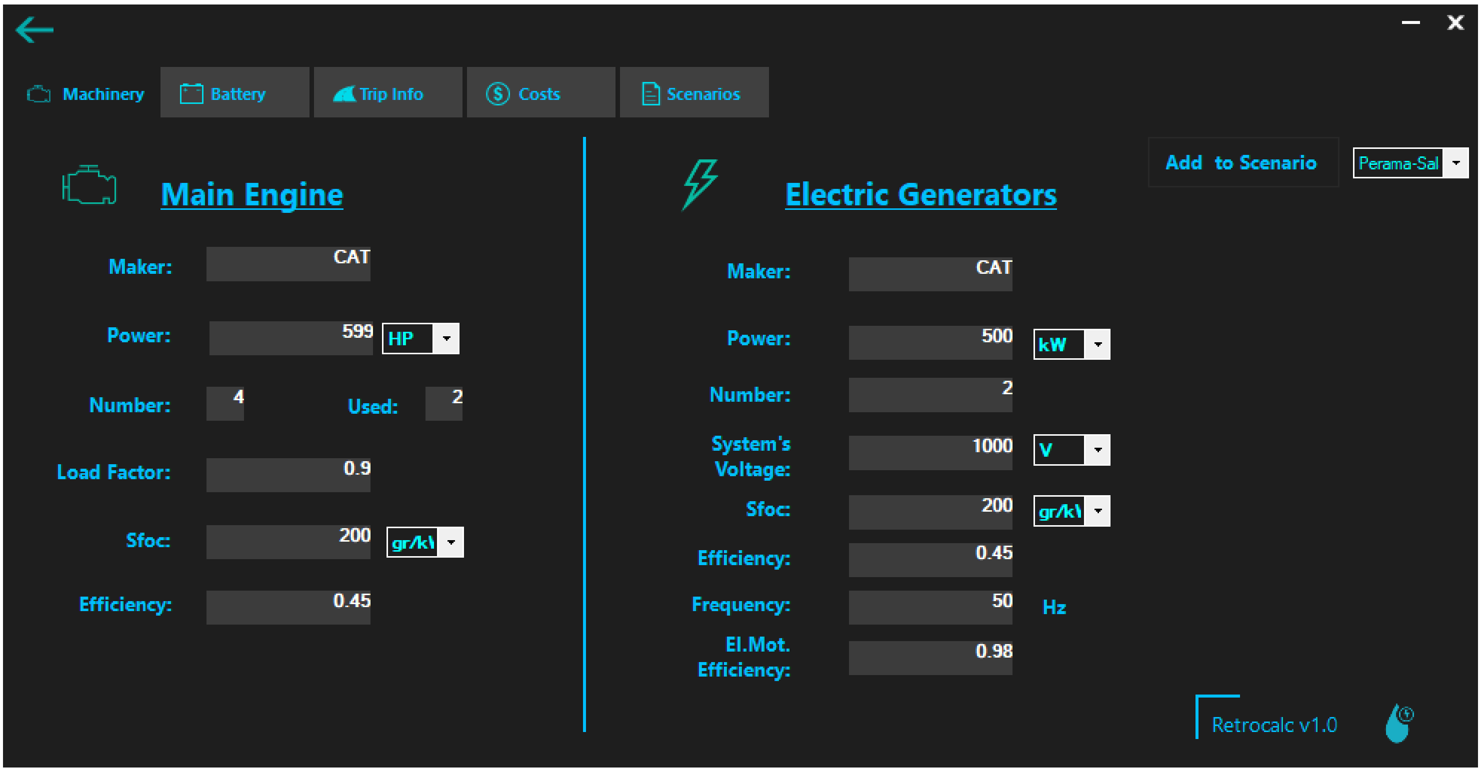

Figure 4.1: Perama-Salamina, Machinery Values

Battery related Data

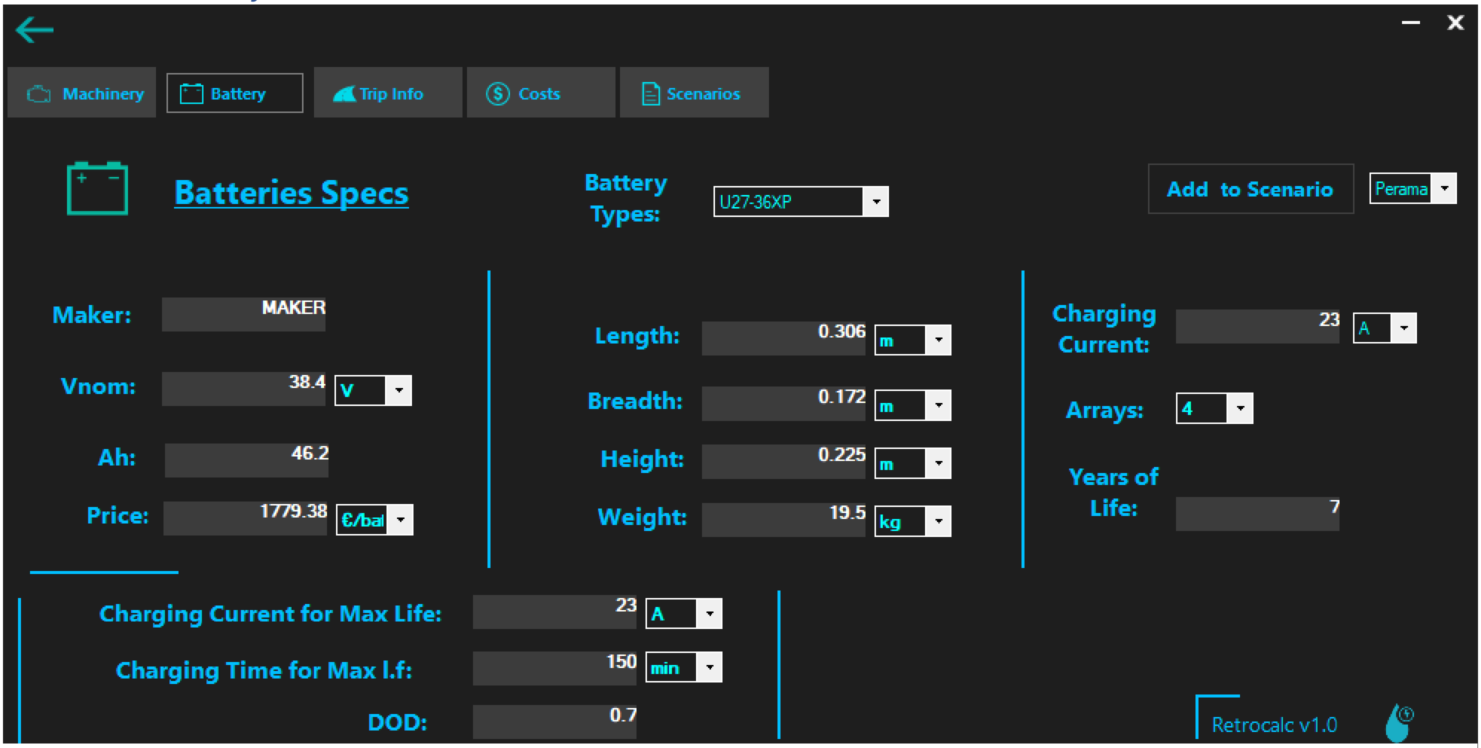

Figure 4.2: Perama-Salamina, Battery Values

Trip related info

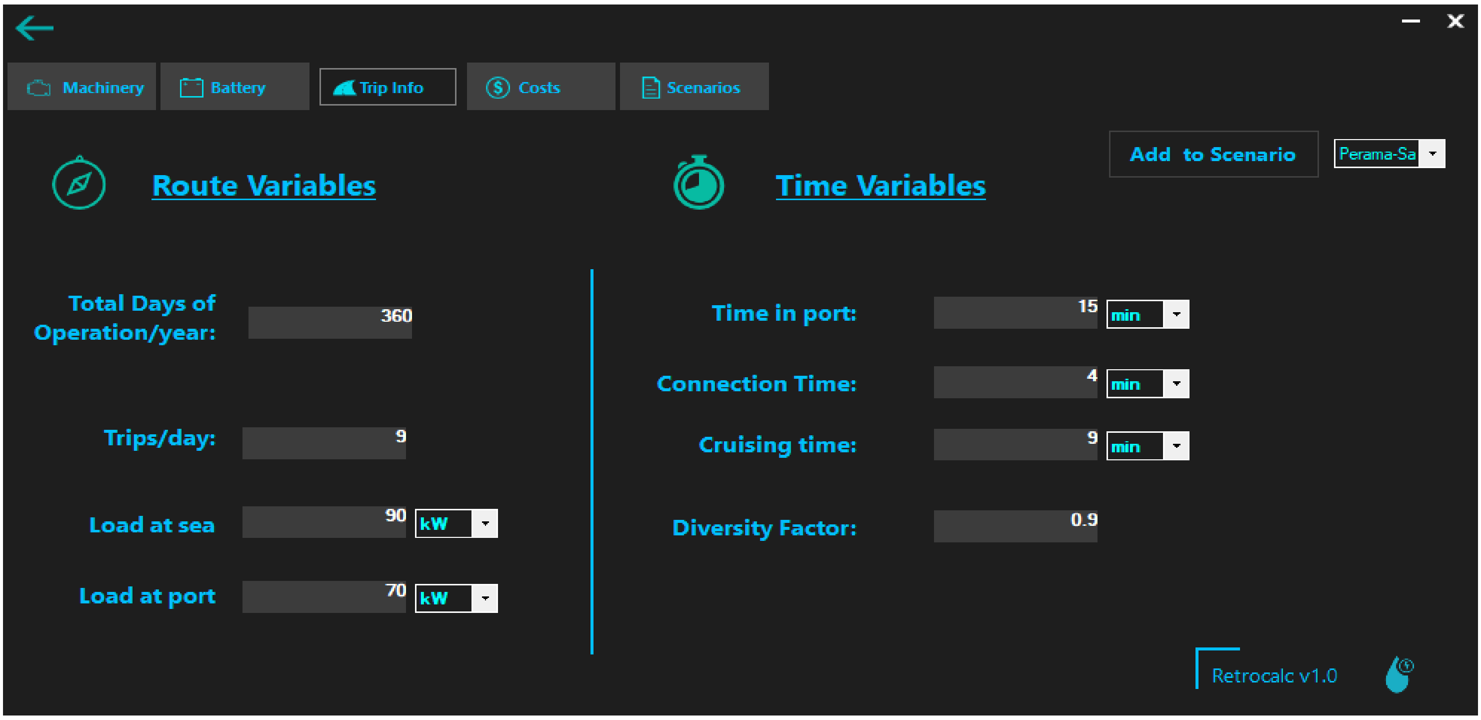

Figure 4.3: Perama-Salamina, Trip Info Values

Costs related Data

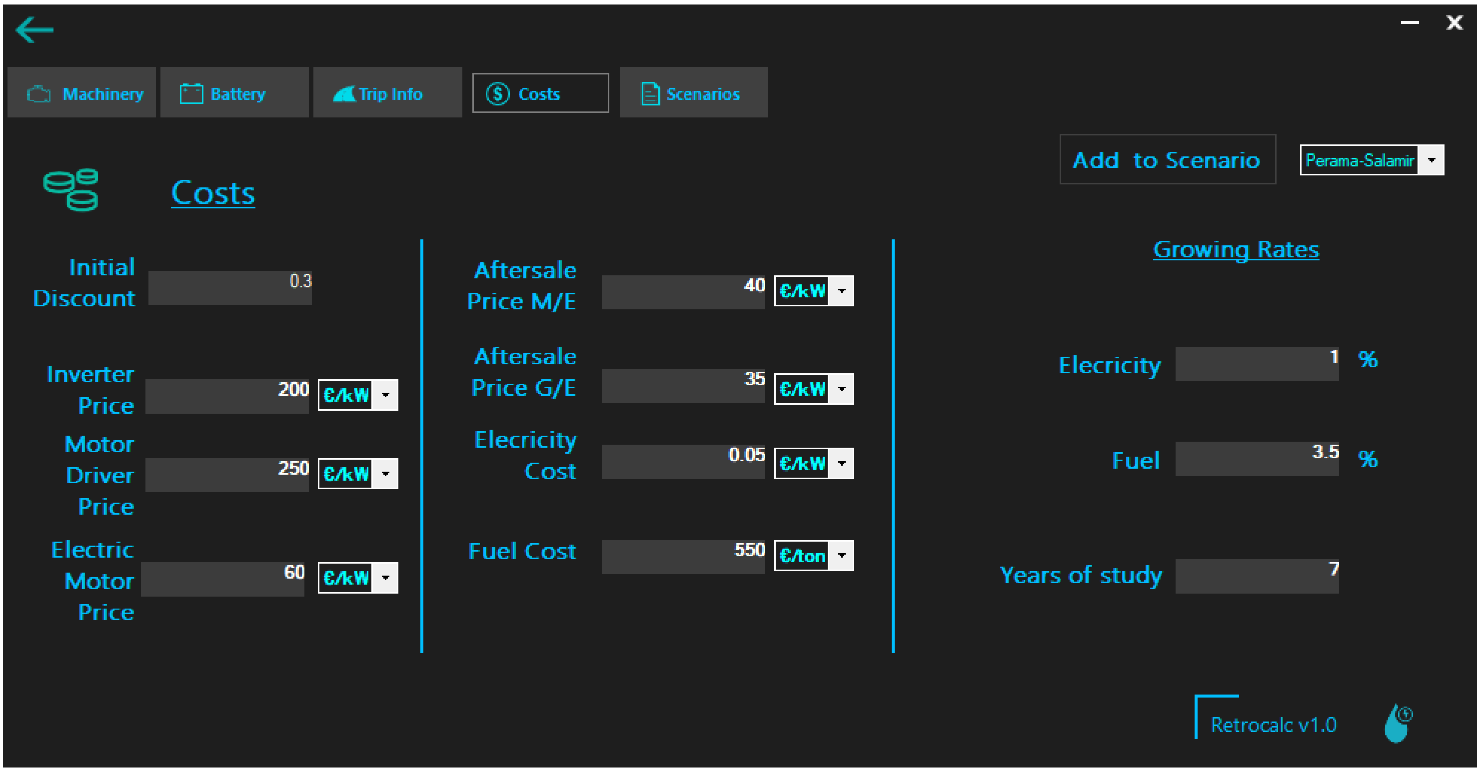

Figure 4.4: Perama-Salamina, Costs Values

4.1.2. Test 2 scenario

Furthermore, the results of another, non-existing, vessel will be shown. The name of the second scenario is “Test 2” scenario. The input data are presented below:

Machinery Data

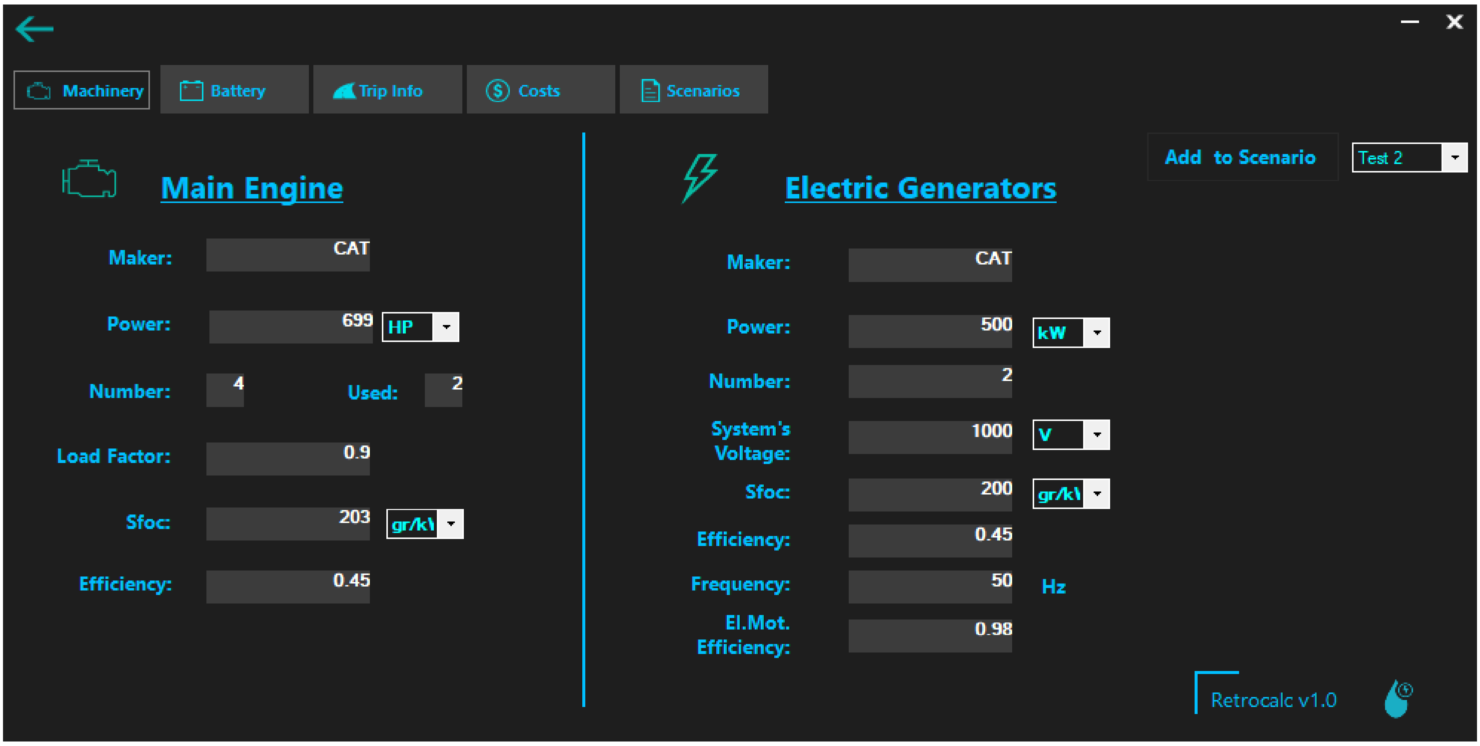

Figure 4.5: Test 2, Machinery Values

Battery related Data

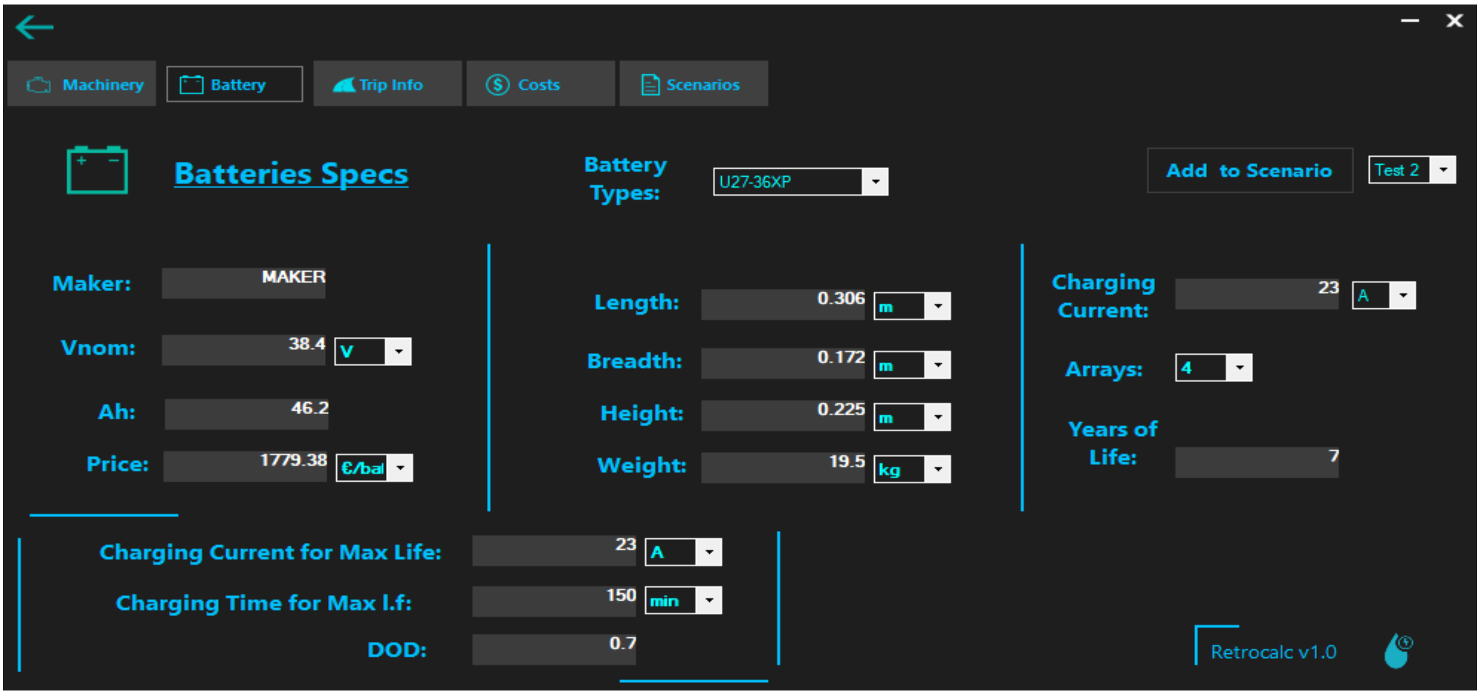

Figure 4.6: Test 2, Battery Values

Trip related info

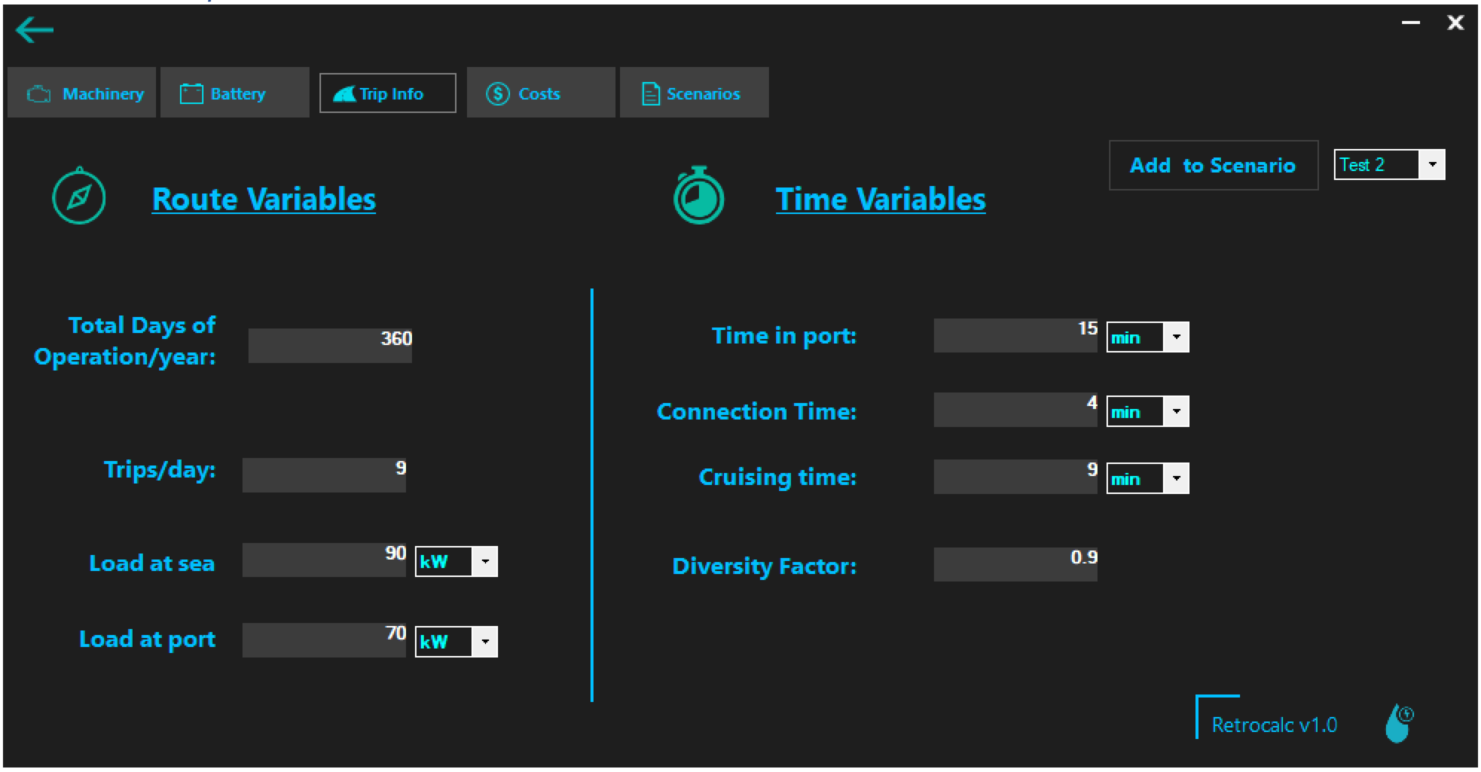

Figure 4.7: Test 2, Trip Info Values

Costs related Data

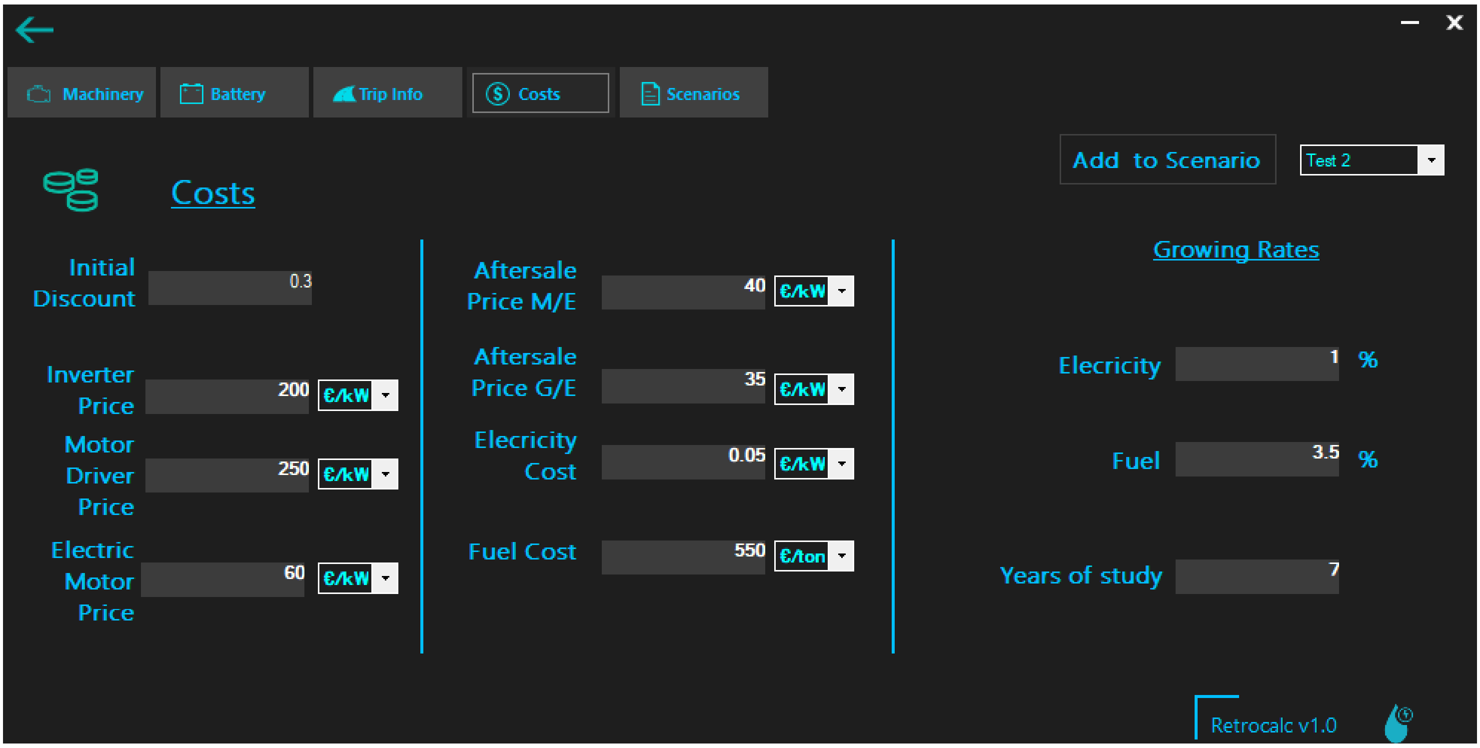

Figure 4.8: Test 2, Costs Values

4.2. Scenarios Presentation

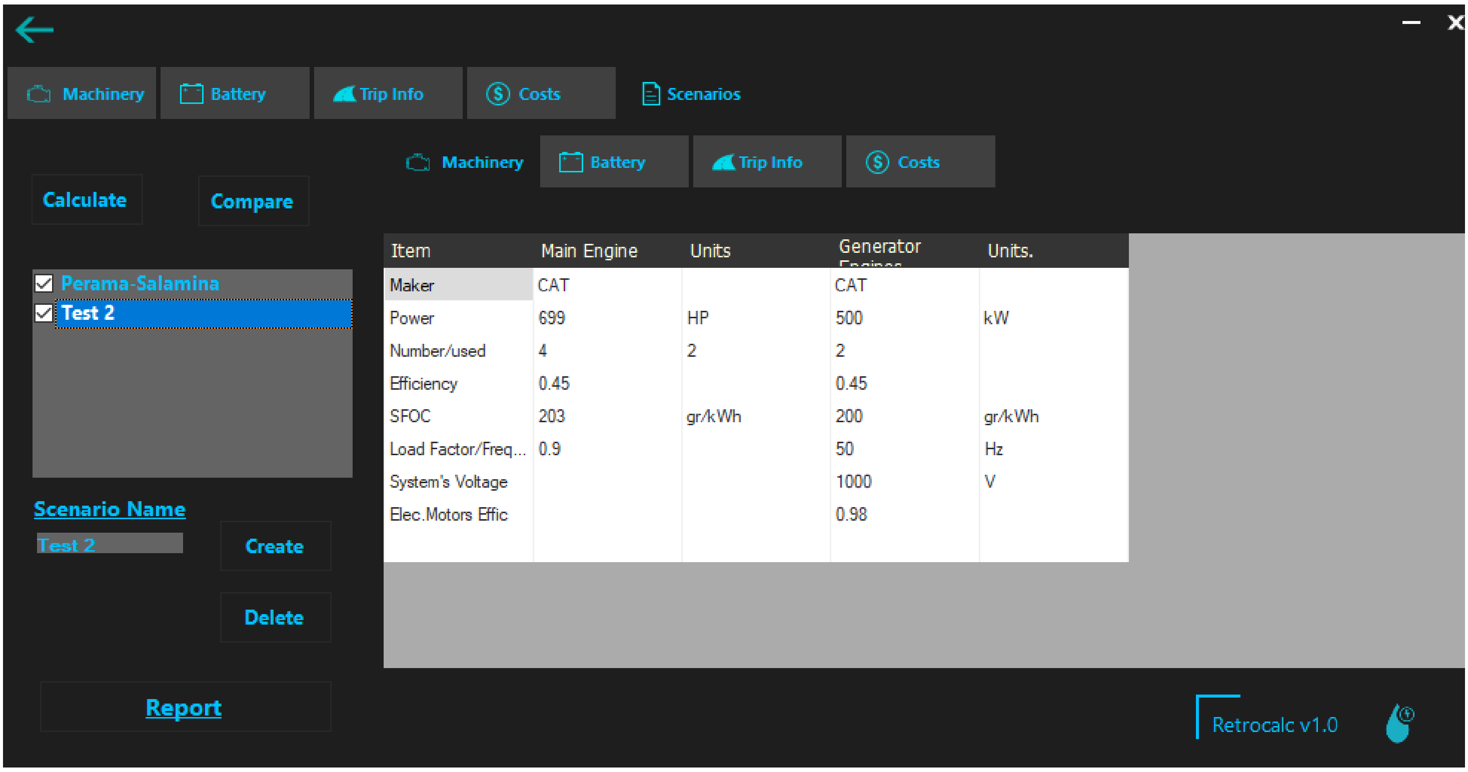

Figure 4.9: Test 2, Scenarios Panel

As it can be seen, the second scenario has almost no differences with the first. The only difference is the power of the Main engines of the vessel, which is one-hundred Horsepower Units bigger than the engines of the first scenario.

After, the input data have been loaded, each scenario is calculated and the report button is pressed, loading the Report Form. The Report Form is presented below.

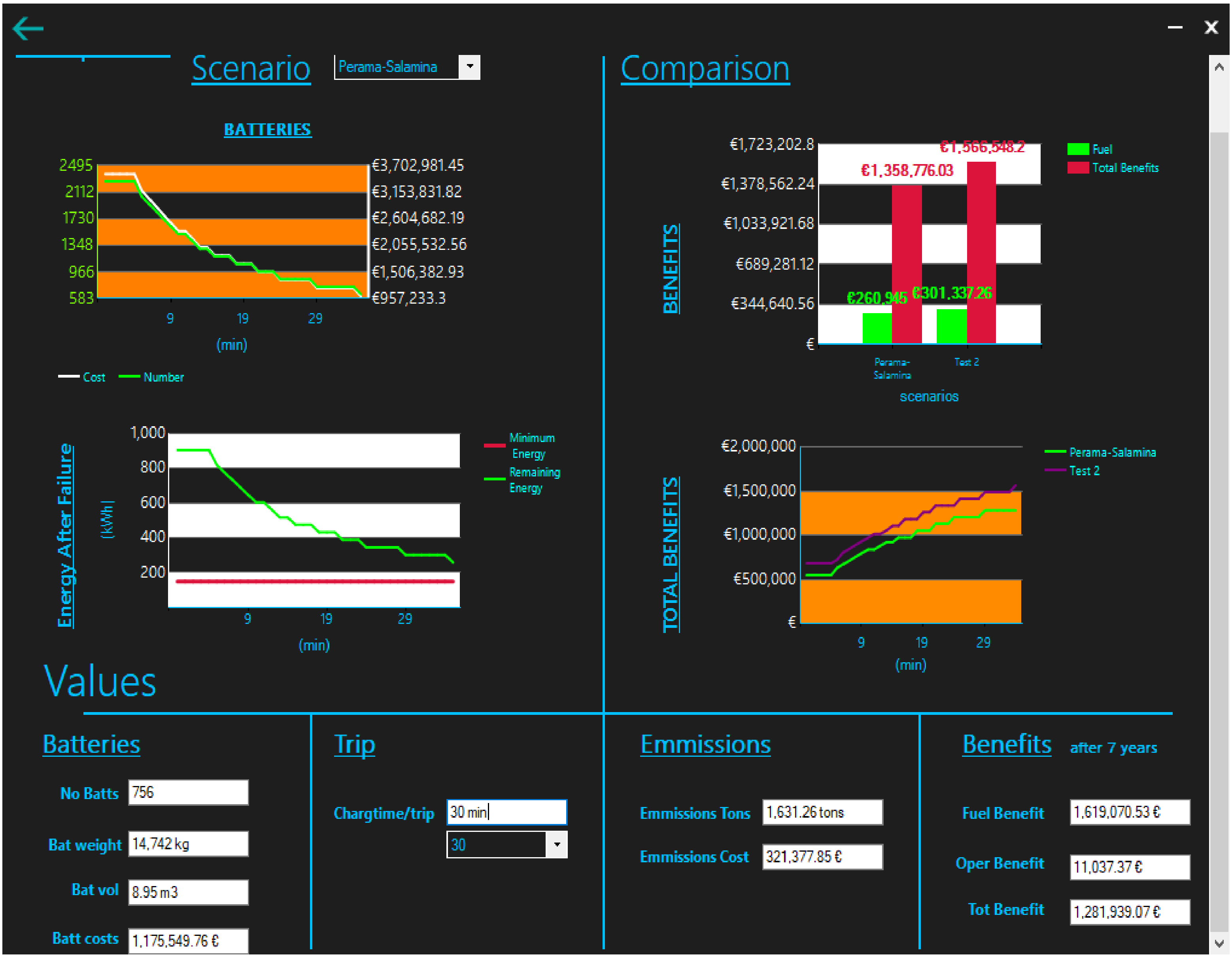

Figure 4.10: Report Form, Comparison of "Perama-Salamina"," Test 2"

In the left side of the form, two graphs are presented, related to the selected scenario. In the above figure, the "Perama-Salamina" scenario is selected.

4.3. Results Comparison

4.3.1. Perama-Salamina Results

The results for this scenario, as have been calculated in Mr. Bakirtzoglou's thesis are presented below:

Charging time per Trip		**Units**
Charging time per Trip	30	min
BATTERIES		
Number of Batteries	756	
Weight of Batteries	14742	kg
Volume of Batteries	8,953	m3
Cost of Batteries	1170.333814	€
EMISSIONS DATA		
Total Mass of emissions	1655.67	tons
Total Cost of emissions	322747.7759	€
TOTAL BENEFITS IN 7-YEAR PERIOD		
Total Benefits from Fuel	1,592,258.2	€
Total Benefits from Operational and Maintenance Costs	10,798.0	€
Total Benefits from Fuel and O&M	1,255,310.0	€

Table 4.1: "Perama-Salamina" Results calculated with Excel Worksheet

4.3.2. Test 2 Results

Furthermore, the results of the “Test 2” scenario are presented in the following figure.

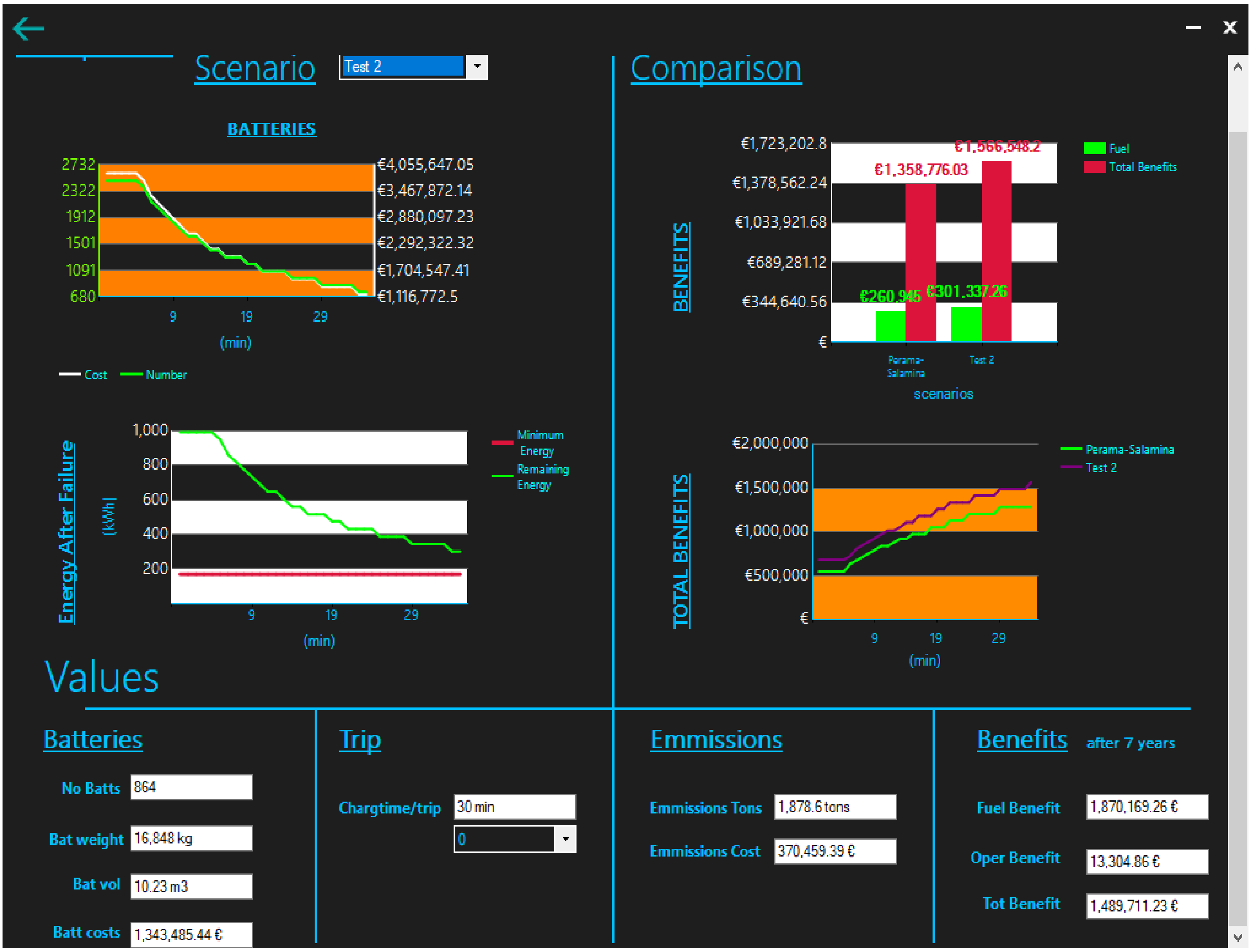

Figure 4.11: Report Form, “Test 2”

4.3.3. Comparison Table

In the above table the results from the program are compared, with those calculated using Mr. Bakirtzoglou's Excel Sheet.

The results from the two method are similar. There are, though, small deviations in the calculated benefits. These differences, between the calculated benefits with the excel sheet and with the use of the program, are emerged from, round down input data in the program

Charging time per Trip	**Calculation Tool**		**Units**
	Excel Sheet	**Retrocalc v1.0**	
Charging time per Trip	30	30	min
BATTERIES			
Number of Batteries	756	756	
Weight of Batteries	14742	14742	kg
Volume of Batteries	8,953	8,95	m3
Cost of Batteries	1,170,333,813.60	1,175,549.76	€
EMISSIONS DATA			
Total Mass of emissions	1,655.67	1,631.26	tons
Total Cost of emissions	322,747.77	321,377.85	€
TOTAL BENEFITS IN 7-YEAR PERIOD			
Total Benefits from Fuel	1,592,258.2	1,619,070.53	€
Total Benefits from Operational and Maintenance Costs	10,798.0	11,037.37	€
Total Benefits from Fuel and O&M	1,255,310.0	1,281,939.07	€

Table 4.2: "Perama-Salamina"," Test 2" Results Comparison

.

4.3.4. Comparison’s Conclusions

From the Comparison Graphs, in the right section of the Report Form, the following conclusions can be extracted:

For the **same time of charging** and **Higher Power installed Engines**:

1. The Benefits from Fuel saving are Higher
2. The Operational Benefits are Higher
3. The Total Benefits are Higher

Also, the results related with **the Batteries** are:

1. Larger Number of Batteries is required
2. A Higher Initial Cost is created

5. Conclusions and Recommendations

5.1. Conclusions

- The use of a powerful computational tool, considering the rising of interest for retrofit conversions, is essential.
- The speed of calculations, including the time for data input, with the use of Retrocalc v1.0 ©, is increased by almost 100%.
- The Simple User Interface makes the Retrocalc v1.0 © approachable to users with little or no Technical knowledge.
- A powerful comparison tool has been created successfully.
- A Database to store basic information for the vessels has been created successfully.
- The objectives set have been accomplished in full.
- The Retrocalc v1.0 © has been proven trustworthy, using the comparison of an already executed study.
- The Retrocalc v1.0 © can be trusted mainly with the use of Double-Ended Ferries or similar Vessels, regarding the operation variables

In general, the objectives set, before the creation of the Retrocalc v1.0 ©, have been accomplished successfully. Though, there is space for further improvement, regarding the calculation procedure. A discussion about the improvement of the program is made in the next chapter.

5.2. Recommendations for further investigation

Ideas for further investigation: More specifically, recommend tools to be added in the current program (e.g. Analysis to be qualified for funding, Add different types of vessels. Or other types of storage for the energy (supercapacitors etc.)

Retrocalc v1.0 ©, is the first computational tool created for the calculation of retrofit studies. Although, a big effort was made creating this tool, there is plenty of space for improvement.

Recommended features to be added in Retrocalc v1.0 ©:

- Modify Calculation Procedure to accept data for different types of Vessels.
- Add Battery Type tool, so the user can enter and store Battery types for later use.
- Add Battery Life tool, which calculates the Battery's life, considering the depth of discharge and the cycles of discharging
- Modify Calculation Procedure to accept data for different types of energy storage (Supercapacitors, Flow Batteries etc.).
- Take into consideration New Regulations and apply the required changes to the Calculation Procedure
- Upload the Retrocalc v1.0 © in a secure Web Page, provided by National Technical University of Athens, so it can be used online, with no need of installation.

The nature of the Retrocalc v1.0 ©, allows the relatively easy modification of its calculation process. For this reason, although there are plenty of improvements to be made, the improvement application is not considered as a time-consuming process. In conclusion, the developers of this program must stay alerted for any new changes related to Regulations or Technical Accomplishments and keep Up to Date the Retrocalc v1.0 © computational tool.

6. References

- “Techno-economical feasibility study on the retrofit of double-ended Ro/Pax ferries into battery-powered ones”, Diploma thesis, Bakirtzoglou C., NTUA March 2017.
- DNV *Guideline for large maritime system.* DNV,2014
- Estimation of annual external health cost of air pollution from ships in the port Piraeus using the impact pathway analysis, Oikonomou,2014
- *Handbook on the estimation of external costs in the transport sector,* CE DELFT,2008
- Air emissions and their impacts, Piraeus port case study, Piraeus, Tzanatos, 2010
- https://www.vbtutor.net
- https://www.dotnetperls.com
- https://stackoverflow.com
- https://docs.microsoft.com
- https://www.codeproject.com

Appendix A

Custom Calculator

Add to Scenario Button

Machinery Tab:

Bellow the code, which runs when the Add **to Scenario** button is pressed, is presented:

```
Private Sub Butto_Click(sender As Object, e As EventArgs) Handles scenariomach.Click
        'Boundaries upper and lower
        'General boundaries for percentage
        Dim percup As Double = 1.0
        Dim perclow As Double = 0.0
        Dim check1 As Boolean = False
        Dim check2 As Boolean = False
        Dim check3 As Boolean = False
        Dim check4 As Boolean = False
        Dim check5 As Boolean = False
        Dim check6 As Boolean = False
        Try
            check1 = emptytext(Panel4)
            check2 = emptytext(Panel5)
            check3 = emptytext(machinery)
            check4 = boundaries(eficme, perclow, percup)
            check5 = boundaries(meloadfactor, perclow, percup)
            check6 = boundaries(efficge, perclow, percup)
            If check1 = True And check2 = True And check3 = True And check4 = True And check5 = True And
check6 = True Then
                If tablemaker.scenarios.Tables.Contains("machchildtable" &
scenariomachval.SelectedItem.ToString) = True Then
                    tablemaker.scenarios.Tables.Remove("machchildtable" &
scenariomachval.SelectedItem.ToString)
                End If
                tablemaker.machtable(gridmach, scenariomachval.SelectedItem.ToString, memaker, genmaker,
mepower, mepowerval, genpower, genpowerval, menum, meused, gennum, eficme, efficge, mesfoc, mesfocval,
gesfoc, gesfocval, meloadfactor, gefreq, systvolt, systemvoltval, elecmotorefic)
            End If
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try


    End Sub
```

First a check is performed to verify if all the required values have been entered by the user. If this check is completed successfully then a second check takes place. Now the program checks if a table with the same name has been created. If a table with the same name exists, it means that values for the selected scenario have been added before. In that case, the existing table is deleted and a new one with the new values is created. The program calls the **tablemaker class** to create the desired table. The class will be presented in the end of this chapter. Click here for the **tablemaker class**

Battery Tab:

Bellow the code, which runs when the Add **to Scenario** button is pressed, is presented:

```
Private Sub addsc_Click(sender As Object, e As EventArgs) Handles scenariobat.Click
        Dim check1 As Boolean = False
        Dim check2 As Boolean = False
        Try
            check1 = emptytext(Panel7)
            check2 = emptytext(Battery)
            If check1 = True And check2 = True Then
                If tablemaker.scenarios.Tables.Contains("batchildtable" &
scenariobatval.SelectedItem.ToString) = True Then
                    tablemaker.scenarios.Tables.Remove("batchildtable" &
scenariobatval.SelectedItem.ToString)
                End If
                tablemaker.battable(gridbat, scenariobatval.SelectedItem.ToString, batmaker, vnom,
vnomval, amperesh, batl, batlval, batb, batbval, bath, bathval, batw, batwval, price, priceval,
chargcur, chargcurval, arraynumval, xyearsoflife, chargcurmax, chargcurmaxval, chargtimemax,
chargetimemaxval, dod)
            End If
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try

    End Sub
```

First a check is performed to verify if all the required values have been entered by the user. If this check is completed successfully then a second check takes place. Now the program checks if a table with the same name has been created. If a table with the same name exists, it means that values for the selected scenario have been added before. In that case, the existing table is deleted and a new one with the new values is created. The program calls the **tablemaker class** to create the desired table. The class will be presented in the end of this chapter. Click here for the **tablemaker class**

Trip Info Tab:

Bellow the code, which runs when the Add **to Scenario** button is presented:

```
Private Sub Button2_Click_1(sender As Object, e As EventArgs) Handles scenariotrip.Click
        Dim daytriplow As Double = 1
        Dim daytripup As Double = 15
        Dim daysup As Double = 365
        Dim dayslow As Double = 1
        Dim check1 As Boolean = False
        Dim check2 As Boolean = False
        Dim check3 As Boolean = False
        Try
            check1 = emptytext(Trip)
            'check boundaries values
            'Set boundaries

            check2 = boundaries(daysyear, dayslow, daysup)
            check3 = boundaries(tripsday, daytriplow, daytripup)
            If check1 = True And check2 = True And check3 = True Then
                If tablemaker.scenarios.Tables.Contains("triptable" &
scenariotripval.SelectedItem.ToString) = True Then
                    tablemaker.scenarios.Tables.Remove("triptable" & scenariotripval.SelectedItem.ToString)
                End If
                tablemaker.triptable(gridtrip, scenariotripval.SelectedItem.ToString, daysyear, tripsday,
loadsea, loadseaval, loadport, loadportval, timecrus, timecrusval, timeport, timeportval, diversfact)
            End If


        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
```

First a check is performed to verify if all the required values have been entered by the user. If this check is completed successfully then a second check takes place. Now the program checks if a table with the same name has been created. If a table with the same name exists, it means that values for the selected scenario have been added before. In that case, the existing table is deleted and a new one with the new values is created. The program calls the **tablemaker class** to create the desired table. The class will be presented in the end of this chapter. Click here for the **tablemaker class**

Costs Tab:

Bellow the code, which runs when the Add **to Scenario** button is pressed, is presented:

```
    Private Sub scenariocosts_Click(sender As Object, e As EventArgs) Handles scenariocosts.Click
        'Boundaries upper and lower
        'General boundaries for percentage
        Dim percup As Double = 1.0
        Dim perclow As Double = 0.0
        Dim check1 As Boolean = False
        Dim check2 As Boolean = False
        Dim check3 As Boolean = False
        Dim check4 As Boolean = False

        Try
            check1 = emptytext(Costs)
            check2 = boundaries(discount, perclow, percup)
            check3 = boundaries(elecgrow, perclow, percup)
            check4 = boundaries(fuelgrow, perclow, percup)
            ' Dim table As DataTable
            ' Dim tablestring As String = "coststable" & scenariocostsval.SelectedItem.ToString
            If check1 = True And check2 = True And check3 = True And check4 = True Then
                If tablemaker.scenarios.Tables.Contains("coststable" &
scenariocostsval.SelectedItem.ToString) = True Then
                    tablemaker.scenarios.Tables.Remove("coststable" &
scenariocostsval.SelectedItem.ToString)
                End If
                tablemaker.coststable(gridcosts, scenariocostsval.SelectedItem.ToString, discount,
inverterprice, inverterpriceval, motordriverprice, motordriverpriceval, motorprice, motorpriceval,
aftersaleprice, aftersalepriceval, aftersalegeprice, aftersalegepriceval, elecprice, elecpriceval,
fuelprice, fuelpriceval, elecgrow, fuelgrow, xyearsofstudy)
                'table = tablemaker.scenarios.Tables.Item(tablestring)
                'gridcosts.DataSource = table
            End If


        Catch ex As Exception
            MsgBox(ex.Message)
        End Try


    End Sub
```

First a check is performed to verify if all the required values have been entered by the user. If this check is completed successfully then a second check takes place. Now the program checks if a table with the same name has been created. If a table with the same name exists, it means that values for the selected scenario have been added before. In that case, the existing table is deleted and a new one with the new values is created. The program calls the **tablemaker class** to create the desired table. The class will be presented in the end of this chapter. Click here for the **tablemaker class**

Calculation Process:

The Calculation Process initializes when the user presses the Calculate button. The analysis of the process will be given with respect to the separate coding blocks.

The code exists inside the Calculator Class. This class will be presented, with respect to the different function that it contains. Furthermore, mathematical equivalents, retrieved from Mr. Bakirtzoglou's thesis, will be given at the end of each code block. These are:

- Basic Values Function
- Charging time available Function
- Energy Demand Function
- Functions used for the calculation of total Number of Batteries
- Batteries energy, cost, weight, volume Function

Basic Values Function:

```
    Friend Shared Function valuescalcforcalc(tripsperday As Integer, loadatsea As Double, loadatseaval
As String, timeatsea As Double, timeatseaval As String, diversity As Double, loadatport As Double,
loadatportval As String, timeatport As Double, timeatportval As String, electricefic As Double, mengpower
As Double, mengpowerval As String, menum As Integer, meused As Integer, meloadfactor As Double) As Double()
        'values transformation
        If mengpowerval = "HP" Then
            mengpower = mengpower * 0.73549
        ElseIf mengpowerval = "PS" Then
            mengpower = mengpower * 0.73549 * 1.01428
        End If
        If timeatportval = "min" Then
            timeatport = timeatport / 60
        End If
        If timeatseaval = "min" Then
            timeatsea = timeatsea / 60
        End If
        If loadatseaval = "HP" Then
            loadatsea = loadatsea * 0.73549
        ElseIf loadatseaval = "PS" Then
            loadatsea = loadatsea * 0.73549 * 1.01428
        End If
        If loadatportval = "HP" Then
            loadatport = loadatport * 0.73549
        ElseIf loadatportval = "PS" Then
            loadatport = loadatport * 0.73549 * 1.01428
        End If
        Dim values(8) As Double
        'propulsion load
        values(1) = mengpower * meused * meloadfactor
        'electric prop load
        values(2) = values(1) / electricefic
```

```
        'load at port
        values(3) = diversity * loadatport * timeatport
        'load at sea
        values(4) = values(2) * timeatsea + loadatsea * timeatsea * diversity
        'energy/trip
        values(5) = (values(3) + values(4)) * 2
        'm/e cons at stanby daily
        values(6) = 0.05 * mengpower * menum * tripsperday * 2 * timeatport
        'g/e cruising consumption/ trip
        values(7) = 2 * loadatsea * timeatsea
        'g/e hoteling consumption/trip
        values(8) = 2 * loadatport * timeatport
        Return values
    End Function
```

With this function, basic values that are not influenced by the charging time are calculated.

Charging time available Function:

```
Friend Shared Function chargingcapacity(tripsperday As Integer, timeinport As Double, timeinportval As
String, dod As Double, chargingcurmax As Double, chargingcurmaxval As String, chargtimemax As Double,
chargtimemaxval As String, chargcurrent As Double, chargcurrentval As String) As ArrayList
        'Values transformation
        Dim back As New ArrayList
        If timeinportval = "h" Then
            timeinport = timeinport * 60
        End If
        If chargtimemaxval = "h" Then
            chargtimemax = chargtimemax * 60
        End If

        If chargingcurmaxval = "mA" Then
            chargingcurmax = chargingcurmax / 1000
        End If
        If chargcurrentval = "mA" Then
            chargcurrent = chargcurrent / 1000
        End If
        'calc charging time per shift adn after every trip & energy demand
        Dim i As Double = 0
        Dim countcharg As Double = 0


        Dim chargtimeshift() As Double
        ReDim chargtimeshift(i)
        Dim chargtimetrip() As Double

        Dim chargcapacity() As Double

        Dim check As Double

        Do
            'chargtimeshift(i) = chargtimemax * chargingcurmax / chargcurrent * dod * (countcharg - 1)
            'chargtimetrip(i) = 60 * chargtimeshift(i) / (tripsperday - 1)
            ReDim Preserve chargtimetrip(i)
            ReDim Preserve chargcapacity(i)
```

```
            countcharg = countcharg + 1
            chargtimetrip(i) = i

            check = chargtimetrip(i)
            chargcapacity(i) = chargcurrent / chargingcurmax * (chargtimetrip(i) / chargtimemax)
            i = i + 1
        Loop While check < (timeinport + 10) 'Edw ginetai elegxos ews otou o xronos fortisis 3eperasei kata
10 lepta ton xrono paramonhs sto limani
        'add countcharg to the last value of chargtime array
        back.Add(chargtimeshift)
        back.Add(chargtimetrip)
        back.Add(chargcapacity)
        back.Add(countcharg)
        Return back
    End Function
```

This function is the heart of the program, as it calculates the charging time available in each port, with respect to the remaining capacity of the battery system. Furthermore, a ten-minute limit is added to the evaluation process.

Energy Demand Function:

```
Friend Shared Function energydemand(chargcapacity As Double(), dod As Double, countcharg As Double,
tripsperday As Integer, energypertrip As Double) As Double()
        Dim i As Integer
        Dim demand(countcharg - 1) As Double

        For i = 0 To countcharg - 1
            'If tripsperday Mod i + 1 = 0 Then
            ' demand(i) = tripsperday * energypertrip / (i + 1) / dod
            demand(i) = tripsperday * energypertrip / ((tripsperday - 1) * (chargcapacity(i) + dod))
            'Else
            'demand(i) = energypertrip * Int(tripsperday / i + 1) / dod

            'End If
        Next
        Return demand
    End Function
```

The Energy Demand Function, calculates and returns the energy demand.

Functions used for the calculation of total Number of Batteries:

```
'calculate n series and n parallel series
    Friend Shared Function series(amperesh As Double, dod As Double, countcharg As Double, systemvoltage
As Double, systemvoltageval As String, vnomval As String, vnom As Double, demand() As Double) As
ArrayList
        'values transfmormation
        Dim back As New ArrayList
        If systemvoltageval = "mV" Then
```

```
            systemvoltage = systemvoltage / 1000
        End If
        If vnomval = "mV" Then
            vnom = vnom / 1000
        End If

        'n series calculation
        Dim nseries As Integer
        If systemvoltage Mod vnom = 0 Then
            nseries = systemvoltage / vnom
        Else
            nseries = Int(systemvoltage / vnom) + 1
        End If
        'parallel series calc
        Dim i As Integer
        Dim seriesbulk(countcharg) As Double

        For i = 0 To countcharg - 1
            If demand(i) Mod (nseries * vnom * amperesh) = 0 Then
                seriesbulk(i) = demand(i) / (nseries * vnom * amperesh) * 1000
            Else
                seriesbulk(i) = Int(1000 * demand(i) / (nseries * vnom * amperesh)) + 1

            End If
        Next


        back.Add(seriesbulk)
        back.Add(nseries)
        Return back
    End Function

    'calc number of batteries in parallel in respect to battery arrays
    Friend Shared Function batparal(countcharg As Double, seriesbulk() As Double, arraynum As Integer,
nseries As Double) As ArrayList
        Dim back As New ArrayList
        Dim i As Integer
        Dim batpar(countcharg - 1) As Double
        Dim batnum(countcharg - 1) As Double


        For i = 0 To countcharg - 1
            If seriesbulk(i) Mod arraynum = 0 Then
                batpar(i) = seriesbulk(i)
            Else
                If seriesbulk(i) + 1 Mod arraynum = 0 Then
                    batpar(i) = seriesbulk(i) + 1
                Else
                    If seriesbulk(i) + 2 Mod arraynum = 0 Then
                        batpar(i) = seriesbulk(i) + 2
                    Else
                        batpar(i) = seriesbulk(i) + 3
                    End If
                End If
            End If
            'total number of batteries
            batnum(i) = batpar(i) * nseries
        Next
        back.Add(batpar)
        back.Add(batnum)
        Return back
    End Function
```

Within these functions, the total number of batteries required is calculated. First, the number of batteries, in series and the number of packs connected in parallel, are calculated. Next, the total number of batteries installed is calculated.

Batteries energy, cost, weight, volume Function:

```
'Total installed energy & energy remain after one array fails at 80%dod
    Friend Shared Function totalenergyinst(arraynum As Integer, batnum() As Double, vnom As Double,
vnomval As String, amperesh As Double, countcharge As Double) As ArrayList
        'Values transformation
        If vnomval = "mV" Then
            vnom = vnom / 1000
        End If
        Dim back As New ArrayList
        Dim i As Integer
        Dim instenergy(countcharge - 1) As Double

        Dim instafterfailure(countcharge - 1) As Double


        For i = 0 To countcharge - 1
            instenergy(i) = batnum(i) * vnom * amperesh / 1000
            instafterfailure(i) = instenergy(i) * 0.2 * (arraynum - 1 / (arraynum))
        Next
        back.Add(instenergy)
        back.Add(instafterfailure)
        Return back
    End Function

    'Weight, Volume and price of batteries
    Friend Shared Function weightvolprice(instenergy() As Double, countcharge As Double, batnum() As
Double, weight As Double, weightval As String, length As Double, lengthval As String, breadth As
Double, breadthval As String, height As Double, heightval As String, price As Double, priceval As
String) As ArrayList
        'Values transformation
        If weightval = "tons" Then
            weight = weight * 1000
        End If
        If lengthval = "cm" Then
            length = length / 100
        ElseIf lengthval = "mm" Then
            length = length / 1000
        End If
        If breadthval = "cm" Then
            breadth = breadth / 100
        ElseIf breadthval = "mm" Then
            breadth = breadth / 1000
        End If
        If heightval = "cm" Then
            height = height / 100
        ElseIf heightval = "mm" Then
            height = height / 1000
        End If
```

```
        If priceval = "$/bat" Then
            price = price * 0.88
            priceval = "€/bat"
        ElseIf priceval = "£/bat" Then
            price = price * 1.11
            priceval = "€/bat"
        ElseIf priceval = "£/kWh" Then
            price = price * 1.11
            priceval = "€/kWh"
        ElseIf priceval = "$/kWh" Then
            price = price * 0.88
            priceval = "€/kWh"
        End If
        Dim i As Integer
        Dim batweight(countcharge - 1) As Double

        Dim batvol(countcharge - 1) As Double

        Dim batprice(countcharge - 1) As Double

        Dim back As New ArrayList

        For i = 0 To countcharge - 1
            batweight(i) = batnum(i) * weight
            batvol(i) = batnum(i) * length * breadth * height
        Next
        If priceval = "€/bat" Then
            For i = 0 To countcharge - 1
                batprice(i) = batnum(i) * price

            Next
        ElseIf priceval = "€/kWh" Then
            For i = 0 To countcharge - 1
                batprice(i) = instenergy(i) * price
            Next
        End If
        back.Add(batweight)
        back.Add(batvol)
        back.Add(batprice)
        Return back
    End Function
```

First the remaining energy, after failure of one array, is calculated. Second, the battery system's volume, weight and cost are calculated and returned in the main program.

The next step is the calculation of **The Operational and Maintenance costs**. This procedure is done by the **Opermaintcosts Class**.

This class will be presented, with respect to the different functions that it contains. Furthermore, mathematical equivalents, retrieved from Mr. Bakirtzoglou's thesis, will be given at the end of each code block. These are:

- Installation Cost Function
- Earnings from the sales of existing machinery Function
- Fuel versus Electricity Comparison Function
- Operational and Maintenance Costs Function
- Benefits after given years Function

Installation Cost Function:

```
    Friend Shared Function instcost(countcharge As Double, batcost() As Double, discount As Double,
inverterprice As Double, inverterpriceval As String, inverterkw() As Double, motordrivpricekw As Double,
motordrivpricekwval As String, motordrivekw As Double, elemotorpricekw As Double, elemotorpricekwval As
String, elemotorkw As Double) As Double()
        Dim i As Integer
        Dim batcosti As Double
        Dim softwarecost As Double
        Dim invertercost As Double
        Dim motordrivescost As Double
        Dim elecmotorscost As Double
        Dim instalcost(countcharge - 1) As Double
        'values transformation
        If inverterpriceval = "$/kW" Then
            inverterprice = inverterprice * 0.88
            inverterpriceval = "€/kW"
        ElseIf inverterpriceval = "£/kW" Then
            inverterprice = inverterprice * 1.11
            inverterpriceval = "€/kW"
        End If
        If motordrivpricekwval = "$/kW" Then
            motordrivpricekw = motordrivpricekw * 0.88
            motordrivpricekwval = "€/kW"
        ElseIf motordrivpricekwval = "£/kW" Then
            motordrivpricekw = motordrivpricekw * 1.11
            motordrivpricekwval = "€/kW"
        End If
        If elemotorpricekwval = "$/kW" Then
            elemotorpricekw = elemotorpricekw * 0.88
            elemotorpricekwval = "€/kW"
        ElseIf elemotorpricekwval = "£/kW" Then
            elemotorpricekw = elemotorpricekw * 1.11
            elemotorpricekwval = "€/kW"
        End If


        For i = 0 To countcharge - 1
```

```
            batcosti = batcost(i) * (1 - discount)
            softwarecost = batcosti * 0.5
            invertercost = inverterprice * inverterkw(i)
            motordrivescost = motordrivekw * motordrivpricekw
            elecmotorscost = elemotorkw * elemotorpricekw
            instalcost(i) = (batcosti + softwarecost + invertercost + motordrivescost + elecmotorscost) * 1.1
        Next
        Return instalcost
    End Function
```

In this part, the initial cost for the battery system is calculated. This includes, the cost of the batteries and the cost of the electrical components required for the conversion.

Earnings from the sales of existing machinery Function:

```
    Friend Shared Function salesmachinery(pricekwused As Double, pricekwusedval As String, priceusedgen
As Double, priceusedgenval As String, menpower As Double, menum As Double, genpower As Double, genum As
Double) As Double
        Dim earnme As Double
        Dim earngen As Double
        Dim totalearn As Double
        If pricekwusedval = "$/kWh" Then
            pricekwused = pricekwused * 0.88
            pricekwusedval = "€/kWh"
        ElseIf pricekwusedval = "£/kWh" Then
            pricekwused = pricekwused * 1.11
            pricekwusedval = "€/kWh"
        End If
        If priceusedgenval = "$/kWh" Then
            priceusedgen = priceusedgen * 0.88
            priceusedgenval = "€/kWh"
        ElseIf priceusedgenval = "£/kWh" Then
            priceusedgen = priceusedgen * 1.11
            priceusedgenval = "€/kWh"
        End If
        earnme = menpower * menum * pricekwused
        earngen = genpower * genum * priceusedgen
        totalearn = earngen + earnme
        Return totalearn
    End Function
```

This function calculates the earnings from the sales of the installed machinery. The installed machinery includes the main engines and the generator engines.

Fuel versus Electricity Comparison Function:

```
    Friend Shared Function fuelcostcomp(totaldays As Double, tripsperday As Double, elecefic As Double, energypertrip As Double, electpricekw As Double, electpricekwval As String, mefic As Double, genefic As Double, propload As Double, cruistime As Double, cruistimeval As String, mestandbyperday As Double, gecruistrip As Double, gehottrip As Double, mesfoc As Double, mesfocval As String, gesfoc As Double, gesfocval As String, pricetonnfuel As Double, pricetonnfuelval As String) As Double()
        If electpricekwval = "$/kWh" Then
            electpricekw = electpricekw * 0.88
            electpricekwval = "€/kWh"
        ElseIf electpricekwval = "£/kWh" Then
            electpricekw = electpricekw * 1.11
            electpricekwval = "€/kWh"
        End If
        If pricetonnfuelval = "$/tonn" Then
            pricetonnfuel = pricetonnfuel * 0.88
            pricetonnfuelval = "€/tonn"
        ElseIf pricetonnfuelval = "£/tonn" Then
            pricetonnfuel = pricetonnfuel * 1.11
            pricetonnfuelval = "€/tonn"
        End If

        If mesfocval = "lb/HPh" Then
            mesfoc = mesfoc * 453.952 / 0.7457
            mesfocval = "gr/kWh"
        End If
        If gesfocval = "lb/HPh" Then
            gesfoc = gesfoc * 453.952 / 0.7457
            gesfocval = "gr/kWh"
        End If
        If cruistimeval = "min" Then
            cruistime = cruistime / 60
            cruistimeval = "h"
        End If
        'Array Cost where in i=1 is batcost and i=2 is fuelcost
        Dim cost(8) As Double
        'calcs for batteries cost
        Dim energyperyear As Double
        Dim totbatcost As Double
        energyperyear = energypertrip * tripsperday * totaldays
        totbatcost = electpricekw * energyperyear
        cost(1) = totbatcost
        'calcs for machine fuel
        Dim totfuelvolperyear
        Dim totenergyperyear As Double
        Dim meanualcruis As Double
        Dim meanualstandby As Double
        Dim gencruis As Double
        Dim genhotel As Double
        Dim totfuelcost As Double
        meanualcruis = propload / mefic * cruistime * 2 * tripsperday * totaldays
        meanualstandby = mestandbyperday / mefic * totaldays
        genhotel = gehottrip / elecefic * tripsperday * totaldays
        gencruis = gecruistrip / elecefic * tripsperday * totaldays
        totenergyperyear = meanualcruis + meanualstandby + genhotel + gencruis
        'check if main engine sfoc is the same as gener. engine sfoc
        If mesfoc = gesfoc Then
            Dim sfoc As Double = mesfoc
            totfuelvolperyear = totenergyperyear * sfoc / 1000000
```

```
        Else
            totfuelvolperyear = ((meanualcruis + meanualstandby) * mesfoc + (genhotel + gencruis) * gesfoc)
/ 1000000
        End If
        totfuelcost = totfuelvolperyear * pricetonnfuel
        cost(2) = totfuelcost
        cost(3) = meanualcruis
        cost(4) = meanualstandby
        cost(5) = gencruis
        cost(6) = genhotel
        cost(7) = totenergyperyear
        cost(8) = totfuelvolperyear
        Return cost
    End Function
```

Through this procedure, the total costs of fuel and electricity, needed for the powering of each system correspondingly, are calculated for the length of one year. The results will be used for the calculation of the total benefits.

Operational and Maintenance Costs Function:

```
    Friend Shared Function operandmaint(countcharge As Double, battcost() As Double, discount As Double,
instenergy() As Double, mepower As Double, mepowerval As String, menum As Double) As ArrayList
        Dim back As New ArrayList
        'calcs for batt
        Dim i As Integer
        Dim fixedcost(countcharge - 1) As Double

        Dim varcost(countcharge - 1) As Double

        Dim totbatcost(countcharge - 1) As Double

        For i = 0 To countcharge - 1
            fixedcost(i) = 2 / 100 * battcost(i) * 0.7
            If instenergy(i) <= 1000 Then
                varcost(i) = 1.76 * instenergy(i) 'values in euros
            Else
                varcost(i) = 0.88 * instenergy(i) 'values in euros
            End If
            totbatcost(i) = fixedcost(i) + varcost(i)
        Next
        'calcs fr mach
        'use of formula from papanikolaou's book
        If mepowerval = "kW" Then
            mepower = mepower * 1.34102
        ElseIf mepowerval = "PS" Then
            mepower = mepower * 1.35962
        End If
        Dim totmachcost As Double
        totmachcost = mepower * menum * 13.6 'power in HP

        back.Add(totmachcost)
        back.Add(totbatcost)
        Return back
    End Function
```

Here the cost for the maintenance of each system is calculated. The results will be used for the calculation of the total benefits.

Benefits after given years Function:

```
Friend Shared Function afterXyears(batcost() As Double, instenergy() As Double, xyearsoflife As Double,
countcharge As Double, totbatcostom() As Double, totmachcostom As Double, costfuelcomp() As Double, xyears
As Double, elecgrowrate As Double, fuelgrowrate As Double) As ArrayList
        Dim back As New ArrayList
        'benefits from fuelcostcomp in x years
        Dim i As Integer
        Dim benefitfuel(xyears - 1) As Double

        Dim fuelcost As Double = (100 + fuelgrowrate) / 100 * costfuelcomp(2)
        Dim eleccost As Double = (100 + elecgrowrate) / 100 * costfuelcomp(1)
        Dim totalbenefit As Double = 0
        Dim allinclbenefit(countcharge - 1) As Double

        For i = 0 To xyears - 1

            If i = 0 Then
                benefitfuel(i) = Math.Abs(costfuelcomp(1) - costfuelcomp(2))
                totalbenefit = totalbenefit + benefitfuel(i)
                Continue For
            End If

            fuelcost = (100 + fuelgrowrate) / 100 * fuelcost
            eleccost = (100 + elecgrowrate) / 100 * eleccost
            benefitfuel(i) = Math.Abs(fuelcost - eleccost)
            totalbenefit = benefitfuel(i) + totalbenefit
        Next
        'insert totalbenefit value in last place of array
        ' benefitfuel(xyears) = totalbenefit
        'TOTAL BENEFIT INCLUDING O&M AND FUEL COMP
        Dim opermainben(countcharge - 1) As Double
        Dim extracostforbats(countcharge - 1) As Double

        Dim timesofrenewal As Double
        If (xyears / xyearsoflife) >= 1 Then
            timesofrenewal = Int(xyears / xyearsoflife)

            For i = 0 To countcharge - 1
                'calc extra costs for new batteries after xyearsoflife
                opermainben(i) = totmachcostom - totbatcostom(i)
                extracostforbats(i) = 0.5 * 0.7 * batcost(i) + instenergy(i) * 2.2
                allinclbenefit(i) = totalbenefit + Math.Abs(totmachcostom - totbatcostom(i)) -
timesofrenewal * extracostforbats(i)

            Next
        Else
            For i = 0 To countcharge - 1
                opermainben(i) = totmachcostom - totbatcostom(i)
                allinclbenefit(i) = totalbenefit + Math.Abs(totmachcostom - totbatcostom(i))

            Next
        End If
        back.Add(benefitfuel)
        back.Add(opermainben)
        back.Add(allinclbenefit)
        Return back
    End Function
```

With the use of this function, the program calculates the fuel, maintenance and total benefits in a given period of time. The depth of time, for the analysis, is given by the user. It may be noticed a significant decrease in the benefits, at the years when the battery's life expires and needs to be replaced.

The final step is the calculation of **The Externalities costs**. This procedure is done by the **Emissionscalc Class**.

This class will be presented. Only one function is included in this class. Furthermore, mathematical equivalents, retrieved from Mr. Bakirtzoglou's thesis, will be given at the end of each code block.

Emission costs Function:

```
Friend Shared Function emmisions(cost() As Double) As Double()
        Dim mecuis As Double = cost(3)
        Dim mestandby As Double = cost(4)
        Dim gecruis As Double = cost(5)
        Dim gehot As Double = cost(6)
        Dim totkwh As Double = cost(7)
        Dim totfuel As Double = cost(8)
        'emissions calculation start
        'emissions per kwh/ton  TIER III
        'FACTORS
        'PM2 FACTORS
        Dim PM2(4) As Double
        PM2(1) = 0.3
        PM2(2) = 0.9
        PM2(3) = 0.3
        PM2(4) = 0.3
        'TSP FACTORS
        Dim TSP(4) As Double
        TSP(1) = 0.3
        TSP(2) = 0.9
        TSP(3) = 0.3
        TSP(4) = 0.3
        'NOx 2010 FACTORS
        Dim NOx(4) As Double
        NOx(1) = 12.3
        NOx(2) = 9.9
        NOx(3) = 13
        NOx(4) = 13
        'SOx TIER I FACTOR
        Dim SOx As Double = 0.2
        'NMBOC FACTORS
        Dim NMVOC(4) As Double
        NMVOC(1) = 0.5
        NMVOC(2) = 1.5
        NMVOC(3) = 0.4
        NMVOC(4) = 0.4
        'CO2 IPPC 2006 FACTOR
        Dim CO2 As Double = 3.19
```

```
        'CH4 FACTOR
        Dim CH4 As Double = 0.18
        'N20 FACTOR
        Dim N20 As Double = 1.3
        'TOTAL EMISSION TONNS
        Dim PM2tot As Double = (PM2(1) * mecuis + PM2(2) * mestandby + PM2(3) * gecruis + PM2(4) * gehot)
/ 1000000
        Dim TSPtot As Double = (TSP(1) * mecuis + TSP(2) * mestandby + TSP(3) * gecruis + TSP(4) * gehot)
/ 1000000
        Dim NOxtot As Double = (NOx(1) * mecuis + NOx(2) * mestandby + NOx(3) * gecruis + NOx(4) * gehot)
/ 1000000
        Dim SOxtot As Double = SOx * totfuel / 1000
        Dim NMVOCtot As Double = (NMVOC(1) * mecuis + NMVOC(2) * mestandby + NMVOC(3) * gecruis + NMVOC(4)
* gehot) / 1000000
        Dim CO2tot As Double = CO2 * totfuel
        Dim CH4tot As Double = CH4 * totfuel * 25 / 1000
        Dim N20tot As Double = N20 * totfuel * 298 / 1000
        'COST ESTIMATION
        'PM2 WITH TZAN METHOD
        Dim PM2cost As Double = 85389.1
        'TSP AS AVERAGE DELFT AND OIKONOMOU
        Dim TSPcost As Double = (93109.5 + 57844.522) / 2
        'NOx WITH OIKONOMOU METHOD
        Dim NOxcost As Double = 3635.19
        'SOx WITH OIKONOMOU METHDO
        Dim SOxcost As Double = 4510.4202
        'NMVOC WITH DELFT METHOD
        Dim NMVOCost As Double = 294.03
        'CO2,ch4,n20 FOR YEAR 2017
        Dim CO2cost As Double = 52.704
        Dim CH4cost As Double = CO2cost
        Dim N20cost As Double = CO2cost
        'TOT AIR POLUTION
        Dim Airpolcost As Double = PM2cost * PM2tot + TSPcost * TSPtot + NOxcost * NOxtot + SOxcost *
SOxtot + NMVOCost * NMVOCtot
        Dim CO2GHG As Double = CO2cost * CO2tot + CH4cost * CH4tot + N20cost * N20tot
        Dim emiscost(7) As Double
        emiscost(1) = Airpolcost
        emiscost(2) = CO2GHG
        emiscost(3) = Airpolcost + CO2GHG
        emiscost(4) = mecuis
        emiscost(5) = mestandby
        emiscost(6) = gecruis
        emiscost(7) = gehot
        Return emiscost
    End Function
```

This function receives as input the loads and mass of fuel consumed by the main engine and the generator engines during birthing and sailing. With these data proceeds to the calculation of the mass of the produced emissions and later to the calculation of the total cost of the emissions saved.

Tablemaker Class:

In this chapter the Tablemaker Class is presented. First the complete code of the class will be given.

```
Public Class tablemaker
    Public Shared scenarios As New DataSet("scenarios")
    Public Shared tablecollect As DataTableCollection = scenarios.Tables

    Shared Sub machtable(datagridview1 As DataGridView, scenario As String, memakertxt As TextBox, gemaker
As TextBox, mepower As TextBox, mepowerval As ComboBox, gepower As TextBox, gepowerval As ComboBox, menum
As TextBox, meused As TextBox, genum As TextBox, mefic As TextBox, gefic As TextBox, mesfoc As TextBox,
mesfocval As ComboBox, gesfoc As TextBox, gesfocval As ComboBox, melf As TextBox, gefreq As TextBox,
gevoltage As TextBox, gevoltageval As ComboBox, motorefic As TextBox)
        'create new mach-child table
        Dim table As New DataTable("machchildtable" & scenario)

        'create table's columns
        table.Columns.Add("Item")
        table.Columns.Add("Main Engine")
        table.Columns.Add("Units")
        table.Columns.Add("Generator Engines")
        table.Columns.Add("Units.")

        'create table's rows
        table.Rows.Add("Maker", memakertxt.Text, "", gemaker.Text, "") '01,03
        table.Rows.Add("Power", mepower.Text, mepowerval.SelectedItem.ToString, gepower.Text,
gepowerval.SelectedItem.ToString) '11,12,13,14
        table.Rows.Add("Number/used", menum.Text, meused.Text, genum.Text, "") '2
        table.Rows.Add("Efficiency", mefic.Text, "", gefic.Text, "") '3
        table.Rows.Add("SFOC", mesfoc.Text, mesfocval.SelectedItem.ToString, gesfoc.Text,
gesfocval.SelectedItem.ToString) '4
        table.Rows.Add("Load Factor/Frequency", melf.Text, "", gefreq.Text, "Hz") '5
        table.Rows.Add("System's Voltage", "", "", gevoltage.Text, gevoltageval.SelectedItem.ToString) '6
        table.Rows.Add("Elec.Motors Effic", "", "", motorefic.Text, "") '7
        scenarios.Tables.Add(table)
        'datagridview1.DataSource = table


    End Sub
    Shared Sub battable(datagridview1 As DataGridView, scenario As String, batmaker As TextBox, vnom As
TextBox, vnomval As ComboBox, ah As TextBox, Length As TextBox, lengthval As ComboBox, Breadth As TextBox,
breadthval As ComboBox, Height As TextBox, heightval As ComboBox, weight As TextBox, weightval As ComboBox,
price As TextBox, priceval As ComboBox, chargcur As TextBox, chargcurval As ComboBox, arrays As ComboBox,
xyearsoflife As TextBox, chargmaxa As TextBox, chargmaxaval As ComboBox, chargmaxt As TextBox, chargmaxtval
As ComboBox, dod As TextBox)
        'create new batteries-child table
        Dim table As New DataTable("batchildtable" & scenario)

        'create talble's columns
        table.Columns.Add("Item")
        table.Columns.Add("Value")
        table.Columns.Add("Units")
        'create tables rows
        table.Rows.Add("Maker", batmaker.Text, "") '0
        table.Rows.Add("Vnom", vnom.Text, vnomval.SelectedItem.ToString) '1
        table.Rows.Add("Ah", ah.Text, "Ah") '2
        table.Rows.Add("Length", Length.Text, lengthval.SelectedItem.ToString) '3
        table.Rows.Add("Breadth", Breadth.Text, breadthval.SelectedItem.ToString) '4
        table.Rows.Add("Height", Height.Text, heightval.SelectedItem.ToString) '5
```

```
        table.Rows.Add("Weight", weight.Text, weightval.SelectedItem.ToString) '6
        table.Rows.Add("Price", price.Text, priceval.SelectedItem.ToString) '7
        table.Rows.Add("Charg. Cur.", chargcur.Text, chargcurval.SelectedItem.ToString) '8
        table.Rows.Add("Life", xyearsoflife.Text, "years") '9
        table.Rows.Add("Charg. Cur. max", chargmaxa.Text, chargmaxaval.SelectedItem.ToString) '10
        table.Rows.Add("Charg. Time max", chargmaxt.Text, chargmaxtval.SelectedItem.ToString) '11
        table.Rows.Add("Arrays Num.", arrays.SelectedItem.ToString) '12
        table.Rows.Add("DOD", dod.Text) '13
        scenarios.Tables.Add(table)
        ' datagridview1.DataSource = table
    End Sub
    Shared Sub triptable(datagridview1 As DataGridView, scenario As String, totdays As TextBox, tripsday As
TextBox, loadsea As TextBox, loadseaval As ComboBox, loadport As TextBox, loadportval As ComboBox, timesea
As TextBox, timeseaval As ComboBox, timeport As TextBox, timeportval As ComboBox, diversity As TextBox)
        'create new trip-child table
        Dim table As New DataTable("triptable" & scenario)

        'create talble's columns
        table.Columns.Add("Item")
        table.Columns.Add("Value")
        table.Columns.Add("Units")
        'create tables rows
        table.Rows.Add("Days/year", totdays.Text, "days") '0
        table.Rows.Add("Trips/day", tripsday.Text, "days") '1
        table.Rows.Add("Load at sea", loadsea.Text, loadseaval.SelectedItem.ToString) '2
        table.Rows.Add("Load at port", loadport.Text, loadportval.SelectedItem.ToString) '3
        table.Rows.Add("Cruising time", timesea.Text, timeseaval.SelectedItem.ToString) '4
        table.Rows.Add("Time in port", timeport.Text, timeportval.SelectedItem.ToString) '5
        table.Rows.Add("Diversity", diversity.Text) '6
        scenarios.Tables.Add(table)
        'datagridview1.DataSource = table


    End Sub
    Shared Function coststable(datagridview1 As DataGridView, scenario As String, discount As TextBox,
inverterprice As TextBox, inverterpriceval As ComboBox, motordriverprice As TextBox, motordriverpriceval As
ComboBox, motorprice As TextBox, motorpriceval As ComboBox, aftersaleme As TextBox, aftersalemeval As
ComboBox, aftersalege As TextBox, aftersalegeval As ComboBox, electr As TextBox, electrval As ComboBox,
fuel As TextBox, fuelval As ComboBox, elerate As TextBox, fuelrate As TextBox, years As TextBox) As
DataTable
        'create new costs-child table

        Dim table As New DataTable("coststable" & scenario)

        'create talble's columns
        table.Columns.Add("Item")
        table.Columns.Add("Value")
        table.Columns.Add("Units")
        'create tables rows
        table.Rows.Add("Discount", discount.Text) '0
        table.Rows.Add("Inverter Price", inverterprice.Text, inverterpriceval.SelectedItem.ToString) '1
        table.Rows.Add("Motor Driver Price", motordriverprice.Text,
motordriverpriceval.SelectedItem.ToString) '2
        table.Rows.Add("Elec. Motor Price", motorprice.Text, motorpriceval.SelectedItem.ToString) '3
        table.Rows.Add("M/E Aftersale Price", aftersaleme.Text, aftersalemeval.SelectedItem.ToString) '4
        table.Rows.Add("G/E Aftersale Price", aftersalege.Text, aftersalegeval.SelectedItem.ToString) '5
        table.Rows.Add("Elect. Price", electr.Text, electrval.SelectedItem.ToString) '6
        table.Rows.Add("Fuel Price", fuel.Text, fuelval.SelectedItem.ToString) '7
        table.Rows.Add("Elect. Grow Rate", elerate.Text) '8
        table.Rows.Add("Fuel Grow Rate", fuelrate.Text) '9
        table.Rows.Add("Analysis Years", years.Text, "years") '10
        scenarios.Tables.Add(table)
```

```
        'datagridview1.DataSource = table
        Return table

    End Function
    Shared Function reportbattable(countcharge As Double, scenario As String, batnum() As Double, batw() As
Double, batvol() As Double, chargtimetrip() As Double, batcosts() As Double, emmismass As Double, emmiscost
As Double, fuelbenf() As Double, operbenef() As Double, totbenef() As Double)
        Dim table As New DataTable("report" & scenario)
        Dim i As Integer
        table.Columns.Add("batnum")
        table.Columns.Add("batweight")
        table.Columns.Add("batprice")
        table.Columns.Add("batvol")
        table.Columns.Add("Chargtime")
        table.Columns.Add("emisstons")
        table.Columns.Add("emmisscost")
        table.Columns.Add("fuelbenef")
        table.Columns.Add("operbenefit")
        table.Columns.Add("totalbenefit")
        For i = 0 To countcharge - 1
            If i <= fuelbenf.Length - 1 Then
                table.Rows.Add(batnum(i), batw(i), batcosts(i), batvol(i), chargtimetrip(i), emmismass,
emmiscost, fuelbenf(i), operbenef(i), totbenef(i))
            Else
                table.Rows.Add(batnum(i), batw(i), batcosts(i), batvol(i), chargtimetrip(i), emmismass,
emmiscost, fuelbenf.Length, operbenef(i), totbenef(i))

            End If
        Next
        scenarios.Tables.Add(table)
        Return table

    End Function
End Class
```

This class is used to create and store data tables into RAM (Rapid Access Memory), during the running of the program. The first four functions retrieve the data, which the user previously added, and store them in table form, to be used in later steps of calculation. The last function generates a data table, in which all the data, that will be used in the report form, are stored.