DEPARTAMENT OF
ELECTRICAL AND COMPUTER ENGINEERING

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

INÊS FILIPA RAMALHO SANTOS

BSc in Electrical and Computer Engineering

OPTIMIZATION OF SOLAR ENERGY USING A MONITORING SYSTEM TO MANAGE THE USE OF HOME APPLIANCES

MASTER IN ELECTRICAL AND COMPUTER ENGINNERING

NOVA University Lisbon

September, 2023

DEPARTAMENT OF
ELECTRICAL AND COMPUTER ENGINEERING

OPTIMIZATION OF SOLAR ENERGY USING A MONITORING SYSTEM TO MANAGE THE USE OF HOME APPLIANCES

INÊS FILIPA RAMALHO SANTOS

BSc in Electrical and Computer Engineering

Adviser: Filipa Alexandra Moreira Ferrada
Assistant Professor, NOVA SST

Co-adviser: Pedro Miguel Ribeiro Pereira
Assistant Professor, NOVA SST

Examination Committee:

Chair: Bruno João Nogueira Guerreiro
Assistant Professor, NOVA SST

Rapporteurs: Patrícia Alexandra Pires Macedo
Associate Professor, IPS ESTS

Adviser: Filipa Alexandra Moreira Ferrada
Assistant Professor, NOVA SST

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

NOVA University Lisbon
September, 2023

OPTIMIZATION OF SOLAR ENERGY USING A MONITORING SYSTEM TO MANAGE THE USE OF HOME APPLIANCES

ACKNOWLEDGMENTS

First, I would like to express my sincere gratitude to my adviser, professor Filipa Ferrada for her guidance, availability, and support throughout this academic journey. I would also like to thank professor Pedro Pereira, my co-adviser for generously sharing his knowledge in the field of energy and for always being ready to assist when needed.

I would like to sincerely thank my university and department for providing me with the resources and opportunities to pursue this research and providing that I made the right choice 5 years ago.

To my friends, thank you for your unwavering support and for being a source of laughter and encouragement during challenging times. I cannot thank you enough for always being there for me.

A special thanks to my friend Carlos, who made this dissertation thesis possible by providing information on his photovoltaic panels and sensor. Without his support, this achievement would not have been attainable.

I want to take a moment to express my heartfelt gratitude to André. Your constant support, understanding, and love have been a constant source of strength and joy throughout this journey. Your encouragement and belief in me have pushed me to reach new heights. Thank you for standing by my side, and for making every moment together a cherished memory.

Lastly, my deepest appreciation goes to my parents, brother and sister-in-law, whose love, encouragement, and unconditional belief in me have been my pillars of strength. Their guidance has not only carried me through the long months of dedicated work on this dissertation but has also accompanied me throughout my entire academic journey and, most significantly, in my whole life.

To everyone who has been a part of my academic journey, whether through guidance, friendship, or encouragement, I am truly grateful. Thank you for being an essential part of this incredible journey.

ABSTRACT

In the world of modern living, the efficient management of energy resources has become an increasingly serious challenge. This challenge arises from the growing reliance on smart devices and the increasing importance of utilizing renewable energy sources, particularly photovoltaic panels, in the context of residential buildings.

This problem is intriguing due to its direct impact on environmental sustainability and energy costs for consumers. The interconnection of smart devices and the accurate prediction of energy consumption present technical challenges that require innovative solutions. Essentially, the effective management of energy resources within modern households is a growing issue that demands original solutions to promote sustainability and affordability.

The approach proposed in this study involves creating a Smart Home Energy Management System (SHEMS), *OptiSun*, through a web application platform. The focus of this solution is to develop a real-time monitoring system for the energy produced and consumed and to be able to program smart devices for the most promising energy periods. Therefore, a predictive machine learning model was created to anticipate solar energy production values, which will allow the SHEMS to program the use of smart home devices during periods of higher energy production and thus be able to monitor the energy generated by the photovoltaic panels. This allows a more efficient usage of the produced energy, providing the user with a cost-effective and environmentally friendly solution.

The successful implementation of SHEMS allows consumers to expect a reduction in energy costs, a reduced carbon footprint due to the increased use of renewable energy and a more efficient, facilitated and personalized home experience. In addition, the monitoring system highlights the potential of this approach to energy management.

Key words: Photovoltaic Panels, SHEMS, Machine Learning, Predictive Model, Renewable Energies, Monitoring System.

Resumo

Nos dias de hoje, a gestão eficiente dos recursos energéticos tornou-se um desafio cada vez mais sério. Este desafio surge da gradual dependência de dispositivos inteligentes e da crescente importância da utilização de fontes de energia renováveis, nomeadamente painéis fotovoltaicos, no contexto habitacional.

Este problema é interessante devido ao seu impacto direto na sustentabilidade ambiental e nos custos de energia para os consumidores. A ligação entre dispositivos inteligentes e a previsão exata do consumo de energia apresentam desafios técnicos que exigem soluções inovadoras. Essencialmente, a gestão eficaz dos recursos energéticos nas habitações modernas é um problema crescente que exige soluções originais para promover a sustentabilidade e a acessibilidade económica.

A abordagem proposta neste estudo envolve a criação de um Smart Home Energy Management System (SHEMS), o sistema *OptiSun*, através de uma plataforma de aplicação Web. O foco desta solução é desenvolver sistema de monitorização em tempo real da energia produzida e consumida e poder programar os dispositivos domésticos para os períodos energéticos mais favoráveis. Para tal, foi criado um modelo preditivo de machine learning para antecipar os valores de produção de energia solar, o que permitirá ao SHEMS programar a utilização dos dispositivos domésticos inteligentes nos períodos de maior produção de energia e deste modo poder então monitorizar a energia gerada pelos painéis fotovoltaicos. Isto permite uma utilização mais eficiente da mesma, proporcionando ao utilizador uma solução rentável e amiga do ambiente.

A implementação bem-sucedida do SHEMS permite que os consumidores esperem uma redução nos custos de energia, uma pegada de carbono reduzida devido ao aumento da utilização de energia renovável e uma experiência doméstica mais eficiente, facilitada e personalizada. Além disso, o sistema de monitorização realça o potencial desta abordagem na gestão energética.

Palavas chave: Painéis Fotovoltaicos, SHEMS, Machine Learning, Model Preditivo, Energias Renováveis, Sistema de Monitorização.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

SEMS	Smart Energy Management System.
PVGIS	Photovoltaic Geography Information System.
DECE	Department of Electrical and Computer Engineering
VESS	Virtual Energy Storage System.
SHEMS	Smart Home Energy Management System.
ERD	Entity-Relationship Diagram
MVC	Model-View-Controller.
API	Application Programming Interface.
UML	Unified Modeling Language
W	Watt.
kWh	Kilo Watt hour.

1

INTRODUCTION

Nowadays, it is well known that the future of energy production tends to be focused on renewable energies, with one of the most popular being the generation of electric energy through photovoltaic panels. Since the world population is growing at an explosive rate, and the energy demand is getting higher, solar energy will play a key role as a renewable energy source since it is reliable, cost effective, everlasting, affordable and efficient when compared to other renewable energy sources (Da Silva, 2022; Kannan & Vakeesan, 2016).

Since the use of photovoltaic panels has a wide and diverse application, it makes sense to start opting for maximum optimization of the energy they generate. This approach not only helps in reducing the energy costs but also contributes for maximizing the use of sustainably produced energy.

1.1 Smart Houses, Solar Panels, and Motivation

It is a fact that the future is heading towards a reality in which smart technology is quite relevant and it becomes evident that the energy that is consumed at home increases when using smart devices. Therefore, in a concept of a home, in which smart appliances dominate, it becomes useful that the energy that powers them, becomes energy produced in a renewable way due to its cost advantages. Thus, it becomes imperative to maximize energy we generate. This minimizes the necessity of drawing energy from the grid to power our homes as much as possible (Alilou et al., 2020).

In this line, there is an emerging need to establish a symbiotic relationship between the utilization of smart appliances and the generation of energy via solar panels. This synergy aims to reduce the costs for users while simultaneously improving their daily lives.

1.2 Problem and Objectives

Expanding on the concept of harmonizing smart appliances with solar energy generation in smart homes, this dissertation aims to tackle specific challenges and enhance the daily experiences of users. We know that it is possible to store the energy produced by solar panels, by using solar batteries, and that through this storage the user of the panels can reduce their energy costs and become more independent from the grid. However, the use of batteries also has disadvantages. The main disadvantage is the price of the batteries, which makes the price of energy storage very high, costing from 500€ up to 10 000€, depending on quality and needs (Reis, 2019). The complexity of the photovoltaic system also increases as the installation becomes more complex, and this in turn, increases the energy price and complexity of the overall system. Finally, one of the main problems is the life span of a battery, which is very short because solar batteries tend to drain faster than other types of batteries. This is since they are not fully charged as often (*Pros and Cons of Solar Battery Storage – Is It Worth It?*, n.d.).

In this context, this dissertation proposes to create an application with a monitoring system that performs tasks when there is enough energy being produced. The need to invest in a battery no longer exists since we are always using the energy produced in real time, preventing it from being sent to the grid without taking advantage of it, and without having to invest in the installation, purchase, and maintenance of solar batteries.

1.3 Approach and Contributions

To optimize the energy produced by a solar panel in the context of a home, a system will be created that monitors the production values of a panel, and therefore, allowing the user to schedule its home appliances according to the energy produced. All the energy from the panel will be used to control the smart home appliances so that the energy loss is reduced, and the user's need to consume energy from the grid is considerably reduced. Therefore, by employing an intelligent monitoring system, we can assign tasks to an application that will manage all appliances based on the real-time energy output from the panels.

To facilitate the process of preparing the appliances, an algorithm will be used so that, on the previous week, a prediction of the energy produced by the panel is made and thus the user can know how much energy can be consumed using the system.

After obtaining these values, the system will get everything prepared and, if on the next week the predicted values are less than or equal to the values obtained in real time, all appliances will work as predicted the week before. Consequently, the user will take more advantage of their investment in a customized way according to their own appliances and preferences.

In this line, the main contributions this dissertation work offers are:

- Optimization of energy production and reduction on energy costs: focusing on enhancing energy efficiency and minimizing energy expenses. It is all about using energy smarter, achieving more with less, and ultimately saving on energy costs.
- Flexibility and customization of a home application to control home appliances: aiming to create a home application that adapts to various appliances and is

tailored to individual preferences. This means it is versatile, accommodating, and can be personalized to align perfectly with a user's lifestyle.

- Real-time monitoring of the energy produced by the photovoltaic panels and consumed by the user: enabling real-time tracking of energy production from photovoltaic panels and its utilization. This ensures users have instant insights into how much solar energy they're generating and consuming, facilitating informed decisions about energy management.

1.4 Document Organization

This document is divided in five different sections:

1. State of the Art – This part covers the important aspects of a Smart Home and Solar Panels in the context of the project. It also discusses monitoring systems, and algorithms to calculate the expected values of production.
2. Conceptual Approach – Where the requirements elicitation and system's architecture are explained.
3. Implementation – Detailed chapter on how the system and machine learning model were developed and the technologies used.
4. Discussion of Results – This chapter covers the results of both the Web application platform and the machine learning model and whether the goals were achieved.
5. Conclusion – Summary of the objective and benefits of executing this dissertation and future works.

2

STATE OF THE ART

In the world of today's living, the blend of technology, eco-friendliness, and innovation is reshaping our idea of home. Smart Homes, with their connected gadgets and systems, bring us convenience, efficiency, and eco-conscious living. In this chapter, we dive into the world of Smart Homes, looking at their design, the various devices they include, their partnership with renewable energy sources like solar panels, and the role of machine learning in prediction energy production and making the smart homes smarter and more efficient.

2.1 Photovoltaic Panels

In the past years, the electricity demand has been rising significantly and with that, the need to find alternative renewable energy sources increased. The search for a renewable source came with the need to reduce pollution, waste, carbon emissions and even the cost of energy. With this, the solar energy has been considered one of the greatest sources of renewable energy by being one of the most reliable sources (Alblawi et al., 2019).

Since the energy demand has increased, the prices of energy have also risen and therefore, for those investing in a photovoltaic panel, in the context of a home, it is crucial to save the most money one can. However, it becomes hard to optimize solar energy production when the user is not at home to use it in the peak hours of energy production.

2.1.1 Maximization of the Energy Production

When investing in a solar panel, the main focus is to optimize the energy usage and maximize the energy produced by the panel. To get to maximum energy from a panel there are a few concepts to consider to achieve the goal. The location of the panel is the first aspect to have in consideration. The latitude and longitude of a place is crucial because it will affect the amount of energy the solar panel will produce since the solar radiation is related to the location. The inclination angle of the panel in relation to the horizontal plane of the surface of installation is also an aspect to have in consideration since it will change the amount of the energy being absorbed by the panel since the sun rotates around the earth and has different altitude angles throughout the day. Meteorological conditions of a location also affect the performance of a solar panel. Thus, when installing a solar panel, we must consider its annual production of energy and understand that it depends on the factors described above (Nfaoui & El-Hami, 2018).

When analyzing the irradiation data of a place using the Photovoltaic Geographical Information System (PVGIS) we can select the Department of Electrical and Computer Engineering in NOVA School Science of Technology (Lat/Long 38.661, -9.205) as an example and obtain the values of irradiation for this location. The PVGIS provides information on solar radiation and photovoltaic system performance for any location in Europe and Africa, as well as a large part of Asia and America (*Photovoltaic Geographical Information System*, n.d.). This system is used to analyze, plan, and manage solar energy resources by considering factors such as solar radiation, weather patterns, shading, and terrain. The values obtained are seen in Figure 2.1-1 and we can see that during the colder months the values for irradiation are significantly lower than the values obtain during the colder months with an azimuth angle of 0º and a slope of 30º. The azimuth angle is the angle between the meridian and the location of the vertical surface passing through the sun (Nfaoui & El-Hami, 2018). This helps to understand where and how to best install a solar panel since altering the location, slope and, azimuth angles of a solar panel will differentiate the values of irradiation obtained. The slope angle or inclination angle is the angle of the photovoltaic model from the horizontal plane.

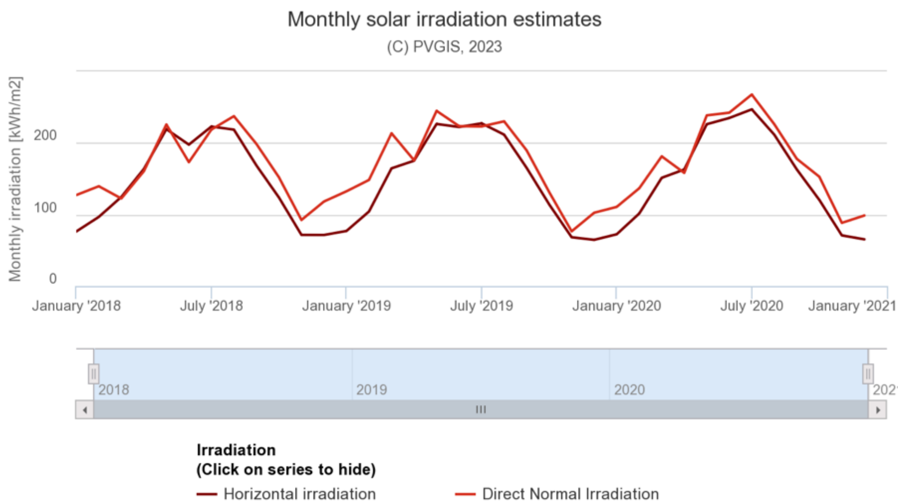


Figure 2.1-1 - Average Monthly solar irradiation in the Department of Electrical and Computer Science Engineering in NOVA SST from January 2018 until January 2021generated from (*Photovoltaic Geographical Information System*)

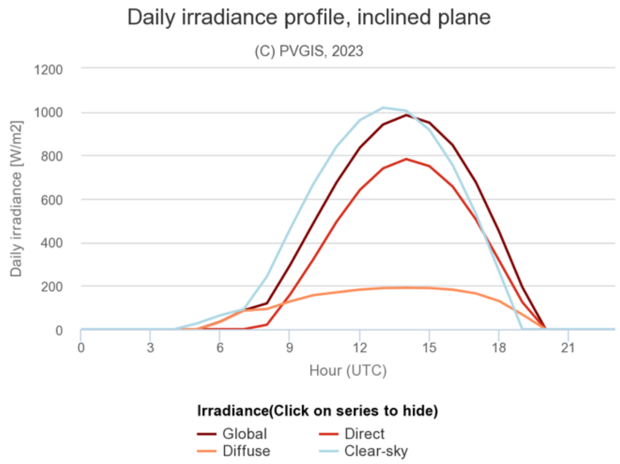


Figure 2.1-2 - Average Daily irradiance in July with an inclination angle of 30º and an azimuth angle of 60º generated from (*Photovoltaic Geographical Information System*) in DECE

The power output of a photovoltaic module is related to the values of irradiance, which is the radiant flux in a surface by unit area, that reach the cells of the photovoltaic panel. Besides discovering the perfect azimuth and slope angles, the physical conditions of the panel itself are also a factor, e.g, the dust accumulation which can cause shading and therefore, reduce the irradiance and power produced by the panel. Consequently, the maintenance and cleaning of the panel must also be taken into consideration. This factor has been facing a lack of attention among researchers once one might have the idea that the rainfall in the location can be enough to clean the panels. However, it might not be enough leading to the need for manual or mobile cleaning, with the help of machinery to perform the task (Maghami et al., 2016).

In this dissertation work the maximization of the solar energy production is considered maximized since the panels are already installed considering the best angles for the location.

2.1.2 Advantages and Disadvantages of using Solar Panels

The usage of solar panels can be very beneficial since they can provide autonomy and safety to its user. When installing solar panels, even when living more than 500m from the grid, it is a benefit. Considering a user installs solar panels in their home, they can enjoy some comfort during the fall and spring seasons. However, it's important to note that their reliability might decrease during the winter months. In some locations the user of solar panels can even sell the energy produced but not consumed to the state, profiting. Additionally, the installation of solar panels extends beyond residential buildings. They can also effectively power billboards, street lighting, and even small consumers that might not have access to the grid. (Boldyrev & Herasymenko, 2020).

Although there are many benefits, as mentioned above, there are also some challenges when deciding on implementing a solar panel. The high cost for installation and maintenance is one of the main disadvantages since it is a high value investment and profiting or saving the money once invested may take some time. The efficiency of the panels might not be best at some points due to meteorological conditions, dust, shading or temperature of the panel (if it reaches high temperatures, the power losses increase, needing a cooling system to keep the temperature levels mainly constant) (Boldyrev & Herasymenko, 2020; Jakhongir Turakul ugli, 2019).

When analyzing both advantages and disadvantages, on the long run, the benefits of installing and investing on a solar panel surpass the challenges once the solar systems are very efficient when working all year, installed properly considering its location and proper installation angles, such as azimuth angles and inclination angles and if taken proper care and maintenance, making them a high cost but sustainable and worth it on the long run investment.

With this dissertation work, it is intended to make the usage of solar panels even more effective considering that the power produced will be optimized and its usage maximized instead of using the electricity from the grid, making the investment much more sustainable and advantageous. The main differences that this dissertation work has from the previous studies are that, for starters, the system of the solar panel will not have energy storage, such as solar batteries, reducing the cost of the installation. It will also be more customizable and fully automated in the sense that the user will have little to no intervention when it comes to the optimization of energy, facilitating its role and utilization.

2.1.3 Apps and Solar Panels

Nowadays, most solar panels companies, besides providing maintenance and installation when investing in a panel, they also provide the user with an app that monitors and informs the user about their consumption and production values for them to have a more enlightened experience and understanding of what they are consuming and producing. This helps the user to optimize their consumption hours, trying to adapt and coincide them with the hours of higher production, if they didn't invest in a battery, so they can take the most out of their investment.

Although this seems helpful, the user has to schedule and coordinate the appliances and devices working hours even though the consumption and production levels are being monitored automatically per se. The user is the one who still needs to analyze when it is better to work the devices and appliances, making this process not fully automated. This is the main goal of this dissertation, so the user won't have to worry about much, except loading and unloading the dishwasher or the laundry machine for example, making this experience more enjoyable and easier.

2.2 Smart Homes

The introduction of Smart Homes in a more global sense is providing its users with many benefits. With the help of Smart Homes and Home Automation we are capable of controlling home appliances and domestic features with just a few buttons on our mobile phones (Gunge & Yalagi, 2016).

When implementing Smart Homes, the primary focus lies in their automation capabilities. When a system is automated, it has the autonomy to operate electronic devices with the minimum involvement of a human being, facilitating the life of the user. Therefore, the use of automation is increasing everyday with numerous benefits. With the use of automated systems in a home, we can control and monitor the most varied appliances and machines such as lights, washing machines, heated floors, air conditioner and fans. This approach enhances the efficiency of homes in terms of electronics and devices, potentially leading to reduced energy and water wastage when overseen by an intelligent system (Gunge & Yalagi, 2016).

2.2.1 Benefits and Challenges of a Smart Home

The main benefits of using automation in a home are that it facilitates domestic activities by making it more convenient, comfortable, safe, and cost-effective.

Although implementing a Home Automation System can be very beneficial, it also provides the user some challenges and difficulties. Since this system is technology centered, the costs of manufacturing, development and installation are considerably high. The fact that this system needs more support, maintenance and additional service costs challenges the usage of the system in question (Gunge & Yalagi, 2016).

Other challenges that can be faced when implementing a Smart home are the internet connection quality, the necessity to keep the home appliances and gadgets as close to the router as possible, and even the compatibility between applications, such as Smart Home Control Apps, Appliance-Specific Apps and even Voice Assistant Apps, and home appliances. In a smart home, the internet connection must be nearly flawless. If the connection fails, the gadgets and appliances will cease to function and most likely disconnect, compromising the core "smart" functionality. Sometimes the gadgets might also go off at the wrong undisclosed times, such as alarms, due to bugs on the system, making it an annoying challenge. There are some also some difficulties on operating a smart home, from families that find it hard to manage their homes, since it may be hard to always have to reach for the phone to control the living area on a regular day at home (Startup Info Staff, 2022).

2.2.2 Architecture and Infrastructure of a Smart Home

In the figure below (Figure 2.1-1) we can analyze the top-level architecture of a typical smart home where there can be seen a server, gateway, or bridge, that makes the connection between the internal networks and external networks. This connection facilitates the integration of the smart grid with the user's devices, which comprehend a wide range of gadgets such as HVAC systems, security systems, as well as diverse household appliances including washing machines, dryers, and TVs, among others.

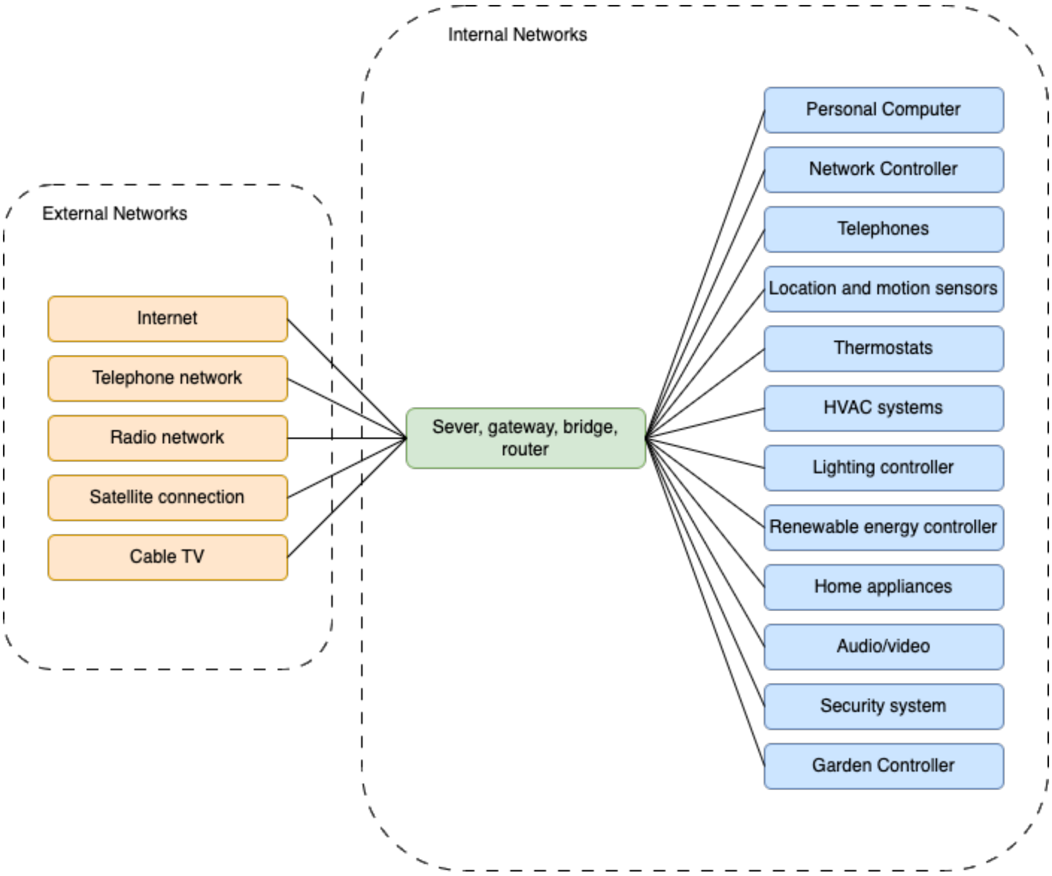


Figure 2.2-1 - Architecture of a typical Smart Home. Modified from reference (Lobaccaro et al., 2016)

Nowadays we can control the environment in a home by implementing a smart home system. This is done by controlling the appliances that oversee lighting and heating depending on the different climatic conditions. Control schemes can now monitor internal environment and independently take pre-programmed actions and operate devices depending on the predefined patterns and even on the user's specific requirements (Al-Ali et al., 2011; Robles & Kim, 2010). This method can be applied in circumstances other than controlling the internal environment of a home but also, for monitoring the energy produced by a solar panel and therefore, operate devices according to the user's preferences, and the amount of energy being produced in real time.

Smart Home Energy-Management System (SHEMS)

SHEMS are efficient services that oversee the monitoring and management of the electricity generation, storage, and consumption in a home. This system's main functions are represented in Figure 2.2-2.

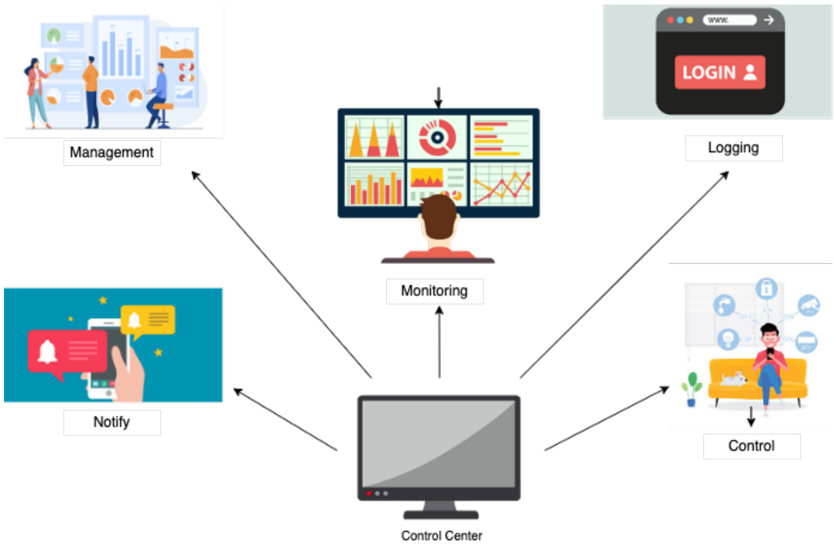


Figure 2.2-2 - Scheme of a SHEMS

In Figure 2.2-2,we can analyze the main functions of a SHEMS scheme which will serve as a foundation for this dissertation work. This SHEMS starts with monitoring the consumed power of different appliances and the home's environment such as temperature or lighting. The monitoring system is a big part in smart homes since it is where the data is collected and analyzed so that the user can know what the conditions are and know what to do. The monitoring system senses what is happening in the home.

The logging function gathers data and stores it within the user's profile. This feature empowers users to personalize their home appliances, choosing their preferred settings and environments. This customization is vital due to the varying electricity consumption associated with each distinct appliance.

Later we have the control function which can be both direct or remote-control. Some appliances can be directly controlled by the SHEMS so it can follow the user's preferences or can be controlled through a smartphone or computer (El-Azab, 2021). The control function executes the tasks that are pre-defined so once which achieve certain condition or conditions, it will execute certain action.

The most crucial part of the SHEMS scheme is the management function. It acts like the conductor that coordinates the energy sources, such as solar panels and home appliances, to make the system work at its best. The ultimate goal is to make everything as efficient as possible, so we can use mostly our own energy rather than relying on the grid. This not only helps the environment but also saves the user some money.(El-Azab, 2021).

Lastly, we want to make sure we receive notifications either on our mobile phones or laptop, that informs us of failures, errors or even if the tasks and actions were rightfully executed. This part is assured by the notify function which, by being connected to the control center and the appliances, can alert us when needed.

To comprehend smart homes more in depth, there are many approaches such as communication systems, social impacts, thermal characteristics, technologies, and trends. The study of a smart-home energy-management systems (SHEMS) can also be done in two different modes, offline load scheduling and real time management.

To understand better how a SHEMS is implemented we must look at its infrastructure and understand its main components:

- **Control Centers:** The control center gives the user the capability to control and monitor the different house appliances in their home. The real-time data is collected by SHEMS and then analyzed to execute the predefined objectives. The control center oversees collecting data from different meters, sensors, commands given by the user, and grid utility through a communication system. The control center also provides the right monitoring and analysis of energy consumption in the home with the support of a smart meter. At last, the control center coordinates between different appliances to give the user the best solution to execute the predefined task (Zhao et al., 2013).
- **Sources of Electricity:** In a SHEMS, energy sources are vital to its operation. The main source of electricity typically aligns with the project's objectives. For instance, if the goal is to reduce energy costs and promote sustainability, the primary source might be renewable energy, such as solar power from photovoltaic panels.
- **Smart meters:** Smart meters are integral components that enable precise measurement and tracking of energy flows within the system. They accurately record the energy consumed by various devices and, crucially, the energy returned to the grid when excess energy is generated. This data is essential for efficient energy management and billing purposes.
- **Communication tools:** Communication tools serve as the connectivity backbone of a SHEMS. They establish seamless connections between smart devices, control centers, and other smart home technologies, facilitating data exchange and enabling coordinated actions. These tools play a pivotal role in ensuring that all components within the system can interact effectively and respond to user commands or system requirements (Robles & Kim, 2010).

In the context of this dissertation work, it's crucial to recognize how these fundamental components of a SHEMS (Smart Home Energy Management System) come together to enhance energy efficiency and user experience. This work's core revolves around the utilization of a control center, allowing users to seamlessly monitor and manage household appliances in their smart homes. Real-time data collected by the SHEMS is analyzed to execute predefined objectives, facilitating energy optimization. Furthermore, the primary energy source is the photovoltaic panel, aligning with the goal of cost-effective and sustainable energy. Smart meters are employed to meticulously measure energy flows within the system, providing precise data for efficient energy management. Lastly, communication tools establish vital connections, development seamless communication between smart devices and the SHEMS, ensuring an integrated and responsive system, all working collectively to elevate the efficiency and convenience of this dissertation work.

2.2.3 Smart Appliances

Smart Appliances are appliances that can communicate through methods like Wi-Fi, Bluetooth, and cloud connectivity to enable remote control and interaction with users and other devices. They're designed to understand and respond to signals from energy sources, adjusting their operations based on user preferences. Their primary role is to provide interaction with users and other platforms or services. Smart Appliances can also connect with smart meters to gather information about energy usage. These appliances were created with the intention of helping people by making their life easier while having the minimum human interaction (Bertoldi et al., 2019).

In (Elkholy et al., 2022) a study was made where the power consumption and the energy consumption of home appliances were measured. This information is valuable for identifying the appliances with the highest electricity consumption. With this knowledge, it becomes possible to find effective strategies to optimize the energy generated from the photovoltaic panel, considering the varying energy consumption of different appliances. The values obtained from the study are explicit in Table 1.

The appliances were organized in three categories based on the priority of usage of each device and customized accordingly to the preferences of each user. These categories are elastic appliances, shiftable appliance, and fixed appliances. Elastic appliances are defined as being appliances that are fully controllable when it comes to both usage time and power consumption. On the other hand, shiftable appliances can be shifted in time without needing to alter the load profile and are characterized by their operation time pre-defined by the user. Lastly, a fixed appliance is an appliance which its usage or power consumption cannot be altered, therefore it has fixed values (Rahim et al., 2016).

It must be taken into consideration that some of these appliances may work on request while others must be always on. By analyzing Table 2 we can understand which appliances are always on and which appliances are on by request.

Table 1- Power consumption and Energy consumption in the case of study. Modified from reference (Elkholy et al., 2022)

Appliance	Type of Appliance	Quantities	Power Consumption [W]	Operational Time Per Day [h]	Energy Consumption per Day [Wh]
Water Pump	Elastic Appliance	1	200	3	600
Fan		4	80	20	6720
Air conditioner		2	1500	7	
Vacuum Cleaner		1	700	2	1400
Water heater		1	1200	3	3600
Washing machine	Shiftable Appliance	1	700	3	2100
Drying machine		1	2000	1	2000
Dish washer		1	1800	2	3600
Refrigerator	Fixed Appliances	1	100	24	1800
TV		2	100	10	2000
Light		20	11	16	3520
Microwave		1	800	3	2400

Table 2 – Schedule of Operation. Modified from reference (Elkholy et al., 2022)

Appliance	User is at home	User is away	User is asleep
Water Pump	On demand	Off	Off
Refrigerator	On	On	On
Air conditioner	On demand	On demand	On demand
Vacuum Cleaner	On demand	On demand	Off
Water heater	On demand	Off	Off
Space heater	On demand	On demand	Off
Washing machine	On demand	On demand	Off
Drying machine	On demand	On demand	Off
Dish washer	On demand	On demand	Off
Fan	On demand	Off	On demand
TV	On demand	Off	Off
Light	On demand	Off	Off
Microwave	On demand	Off	Off

Given that the majority of appliances in this dissertation are smart and capable of operating on demand even when the user is not present, it becomes crucial to enhance solar energy production through optimization. In a context where the maximum energy production in a day happens when the user is not at home, being able to schedule the operation of various appliances facilitates the user in not only a domestic context but also in an economic context, with the added benefit of reducing the environmental footprint as the energy consumed is directly sourced from solar panels.

2.2.4 Apps in Smart Homes

Many systems were created with the purpose of facilitating the communication between the user and the smart devices and appliances and provide infrastructure, enabling information exchange between user and devices (Alaa et al., 2017).

To take the most advantage of smart appliances, there are mobile applications that help the user to utilize those devices from far away and even schedule their working hours, such as:

- Amazon’s Alexa;
- Samsung SmartThings;
- Apple Home;
- Ecobee;
- Philips Hue;
- Wemo;
- Google Home;
- HoneyWell Home;
- Etc (Kavafia, 2023).

As a result, it's evident that these applications simplify and significantly contribute to the customization of a smart home based on each user's preferences, ultimately aiming to provide the client with the best possible experience.

2.2.5 Studies Made and Existing Systems

To optimize the solar energy produced, several solutions were proposed considering the usage of smart homes and/or smart appliances. Below, we address some of them.

In a study made by Rajeev Kumar Chauhan in 2021, a management system was proposed where it would generate switch ON/OFF control actions for appliances to reduce the energy wasted during the period when the appliance is ON but not being used. The smart energy management system (SEMS) would reduce the operative time of the appliances whilst maintaining the user’s comfort. The SEMS is a system that uses advanced technology to make energy processes more efficient, reduce costs, save energy, improve energy usage patterns, control carbon emissions, and enhance overall energy efficiency for both utility providers and consumers (Mohamed et al., 2015). SEMS is categorized based on its management goals and the methods and tools it uses. The SEMS was implemented to also reduce the electricity bill once the appliances would work during periods of lower cost, with the help of an energy storage such as a battery to store the electricity and use it only during the preferred schedule. Thus, the scheduling and management of the smart appliances according to the energy produced by a photovoltaic panel and electricity costs are the focus of the proposed SEMS (Chauhan et al., 2022).

In 2018, a study was made by (Zhu et al., 2019) where an attempt to schedule the household operations was made considering its characteristics and the user's needs to optimize the energy consumption. Engineering models were created to have a better understanding of the electricity demand patterns in the context of a home. To fix the scheduling of the household appliances optimization problem, a solution based in an improved cooperative heuristic approach was proposed. This algorithm was tested to understand its rate of effectiveness and not only if it was effective, but if it would guarantee the user's comfort, reduced electricity costs and flattening of the total loads on the main grid. This was possible by using multiple smart homes that were aggregated to form a Virtual Energy Storage System (VESS) to reduce power peaks and therefore assisting in the process of operating during low-cost and low-emission periods (Zhu et al., 2019). This seemed like a good solution in terms of the algorithm, but it still differs from the approach proposed for this dissertation work in the sense that it wouldn't consider the energy produced in real time via renewable energies. This means that the energy was always available in the grid and mostly used during low-cost periods.

2.3 Monitoring Systems

A monitoring system is crucial to obtain the important information about the system we are monitoring, to make decisions and control the distributed system's tasks and behavior.

A monitoring system is typically built by configuring generic components that fulfill predefined functions throughout its lifecycle. This configuration includes the setup of system components and data sources, data processing and collection, data storage, system analysis and diagnosis, and presentation, which involves visualization, alerting, and decision-making (Mansouri-Samani & Sloman, 1993). In Figure 2.3-1 we can see the scheme of the different stages and steps that a monitoring system follows more in depth.

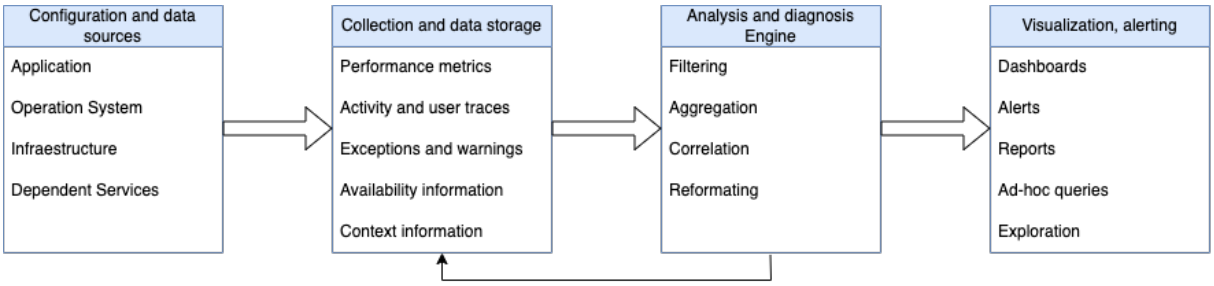


Figure 2.3-1– Life cycle of a monitoring system

Monitoring is also used for many purposes such as debugging, testing and visualization as mentioned above. It manages the more permanent and continuous activities and tasks, and this system might include performance and quality of service management, configuration management, fault, security, and accounting management as their purposes (Mansouri-Samani & Sloman, 1993).

2.3.1 Requirements of a Monitoring System

For a monitoring system to work smoothly, there are several requirements that need to be taken into consideration.

One of the first requirements of a monitoring system are the filtering and correlation capabilities. This requirement is put in place to simplify the comprehension of important and pertinent information, as well as to identify its sources. Sometimes the information collected might not be relevant for that specific level of abstraction or simply not relevant at all to that part of the system managing. Therefore, creating this requirement makes the correlation between important information and the management process easier. Filtering and correlation serve as complementary techniques, effectively handling information within a vast distributed system.

When monitoring a large system with a considerable number of objects, it is always best to monitor the activities as close to the sources of the information being analyzed as possible. This approach allows the components to be more evenly spread throughout the system and therefore reducing the amount of communication, storage and processing facilitating the process of accessing information, making the scalability process being such an important requirement.

Considering that systems suffer many changes throughout their life cycle and implementation period, it becomes important and necessary to change and update the system by adding new machines or removing existing ones. In this work, for example, when a user wants to add a new appliance to their home or update their old by a better or more modern one, it makes sense for the system to update considering the new changes the user made, so they can still use and benefit from their investment. Given the significance of this requirement, it is possible that updates to the system's configuration and functionality might be necessary.

The last requirement for a monitoring system is a functionality that identifies failures. A monitoring service should acknowledge the existence of failures and partial failures since they are intrinsic and inevitable in every system. Both monitored and monitoring components may fail and, therefore, this may lead to wrong or incomplete views of the system's behavior and state (Mansouri-Samani & Sloman, 1993).

2.3.2 Applications of Monitoring Systems

It is known that monitoring systems are helpful since they can inform the user about what is happening with the system and monitor data. Therefore, they have many applications in the day-to-day, facilitating many areas of work such as health, environment, power systems, finances, and many others. These monitoring systems, besides being used in smart homes (Md et al., 2020), can be used to monitor, for example, glucose levels, hand hygiene and even monitor the exercise performed by the user, but also, monitor the energy levels, the power spent, power output and input, and many more. They facilitate so many areas and various tasks, making them, an amazing addition to many companies, hospitals and even users themselves (Kumar et al., 2013; Moparthi et al., 2018; Vashist, 2013; Zanikolas & Sakellariou, 2005).

In this dissertation work, it is proposed a monitoring system that will operate in conjunction with the appliances in the home of the user, presenting the tasks executed and potential failures in a Web Application. This platform will allow the user to define their preferences and tailor the system according to their own home and appliances.

2.4 Machine Learning

Machine learning can be defined as the process of teaching and preparing a computer to perform and execute tasks through repetition in that specific environment.

The technology of machine learning explores and develops computer algorithms which are capable to emulate human intelligence. It applies different areas such as probability and statistics, computer science and even control theory. It can also be applied in various fields like finance, engineering, and health. The reason why this technology is so important and distinct is because the algorithms that are created have the amazing ability to learn as mentioned above by analyzing surroundings and repetition (El Naqa & Murphy, 2015).

When developing machine learning, researchers try to implement an algorithm and an intelligent system that can make decisions independently, thus, the computers must analyze existing data in a way that they can learn from it and later, predict new data coming up with the best solutions and tasks. Therefore, in the area of machine learning, new data and information can be found, analyzed and also predicted (John Wiley & Sons, 2022).

The Machine Learning workflow has 7 main steps that make the process of teaching and coaching a computer possible. These steps are:

- Collecting Data;
- Preparing and Analyzing Data;
- Choosing a Model;
- Training the Model;
- Evaluating the Model;
- Parameter Tuning;
- Prediction Making.

Only by following these steps meticulously we can obtain an algorithm capable of predicting and simulating the human mind when it comes to decision making and consequently, performing tasks (Banoula, 2023). In Figure 2.4-1 we can analyze more in depth the different steps, their purposes, and functionalities.

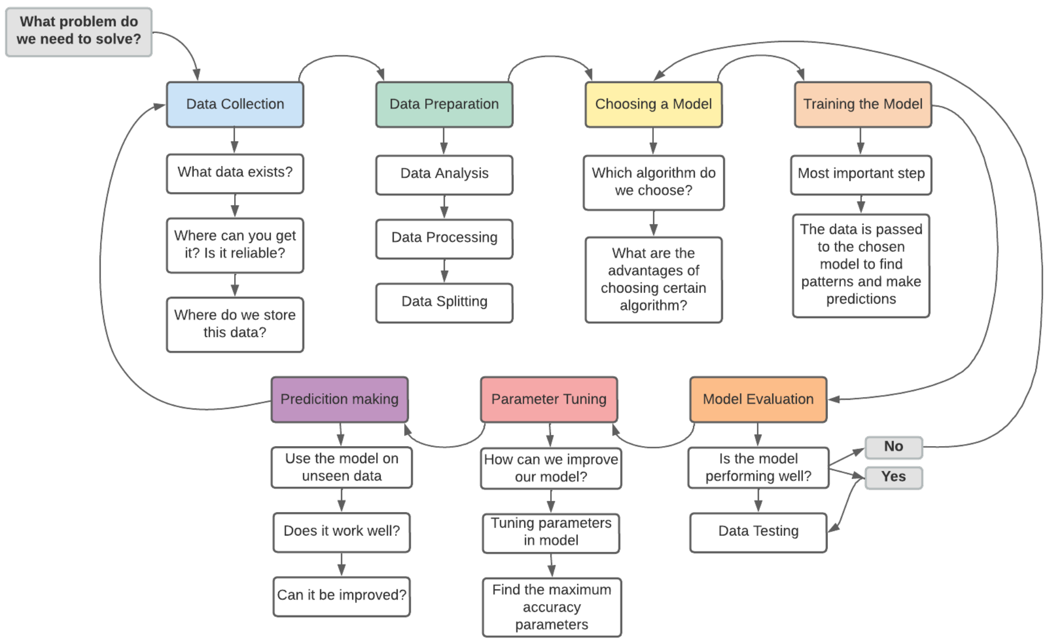


Figure 2.4-1 – Machine Learning Workflow. Modified from reference (Gupta, 2021)

Machine Learning algorithms can be divided in four different categories which differ in purpose and method of learning, with these being (Fumo, 2017):

- Supervised Learning;
- Unsupervised Learning;
- Semi-supervised Learning;
- Reinforcement Learning.

2.4.1 Supervised Learning

Supervised learning consists of the creation of algorithms capable of generation patterns and hypothesis by using externally supplied instances to predict the fate of other future instances. This machine learning algorithm aims at categorizing data from prior information. This classification is important to understand data analytics, pattern recognition and machine learning. (Singh et al., 2016). Supervised machine learning algorithms attempt to model relationships and dependencies between the target prediction output and the input features, so that output values can be predicted (Fumo, 2017).

This type of machine learning algorithm is considered task driven and works in two steps, the classification step and the regression step. In the classification step a diagnostic of the situation is made and we try to understand what the customer wants. Then, in the regression phase, a lot of forecasting is made such as advertising and market forecasting, and an estimated life expectancy of the algorithm is calculated (Petersson & Gillis, 2023).

2.4.2 Unsupervised Learning

Unlike supervised learning, unsupervised learning is trained with unlabeled data and is considered data driven, e.g, the computer learns patterns in data by itself even when the human being doesn't know what to look for in the data being analyzed. These algorithms are mainly used in pattern detection and descriptive modeling and use techniques on the data given to search for rules, patters and even summarize and group some data points, which, later on, can help describe the data in a better way to the users (Fumo, 2017).

Unsupervised learning can improve network performance and provide services, such as, detection of anomalies, quality of service optimization, among others. It is successful in many areas, such as, computer vision, natural language processing and even optimal control. It facilitates the analysis of raw datasets, and therefore, help generating analytic insights from unlabeled data (Usama et al., 2019).

In unsupervised learning there are two types of algorithms that can be developed:

- Clustering;
- Dimensionality reduction.

The clustering algorithm analyses the targeted marketing, customer segmentation and systems recommended to achieve better results for a certain user. On the other hand, dimensionality reduction algorithm visualizes the big data input, chooses a path for a more structured discovery and meaningful compression and also feature elicitation (Pratt & Gillis, 2023) .

2.4.3 Semi-Supervised Learning

Semi-supervised learning falls between both supervised and unsupervised learning, e.g, it has both labeled and unlabeled data inputs. The predictive model is developed by learning from a few labeled examples of data, in a large group of unlabeled ones. This method of machine learning can be applied in various scenarios. However, some situations where unlabeled data was used, a lack in the quality of the performance was observed leading to the need of exploiting unlabeled data in a safe way.(Li & Liang, 2019).

2.4.4 Reinforcement Learning

Reinforcement Learning is a branch of Artificial Intelligence that allows machines and software agents to automatically determine the ideal behavior, when given a specific context, to achieve the best performance possible. The feedback given to the agent is rewarding feedback so that it can learn its behavior better and faster. This feedback is called reinforcement signal (Fumo, 2017).

This type of machine learning learns from trial and error, learning every time from its mistakes. It has various applications, such as, learning tasks, skill acquisition, robot navigation, real-time decision making, game artificial intelligence finance, healthcare and even self-driving cars(Alharin et al., 2020).

In Figure 2.4-2 we can observe and understand a little better the different steps of reinforcement learning.

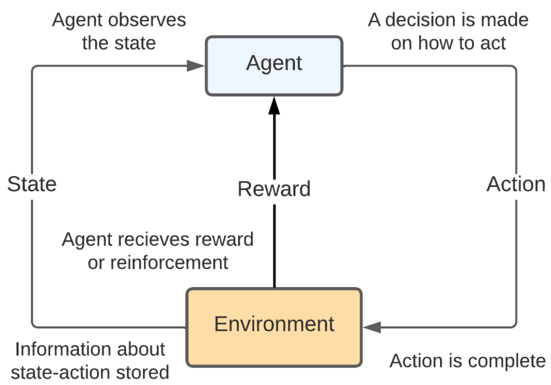


Figure 2.4-2 – Workflow of Reinforcement Learning. Modified from reference (Fumo, 2017)

There are many advantages when implementing a machine learning system since they are great to solve problems where it might be hard to write and come up with rules and algorithms. A machine-learning system can handle complex problems because of its amazing capability of finding new patterns in data that do not exist. Moreover, they perform well with little to almost no human interaction to perform a certain task besides having the capability of adapting over time to changes and updates in data, making them a good system to choose.

Although the advantages seem great, there is always a more negative side to each system. A machine learning system operates with limitations. It requires ample inputs and data to make predictions effectively. When presented with insufficient information, its predictive capabilities are likely to fail. In addition to this problem, these systems can only learn from data they've seen and known before, making it hard to learn new information on their own if a human doesn't teach them new information. For example, a data modeling trained on photos of dogs probably won't be able to recognize a picture of a cat even if very similar since it wasn't taught to do so in advance. Finally, they will never achieve the capability of a human mind since they are not intelligent in the same sense that is the human intellectual, defaulting some areas of work since they lack of common sense (Swaminathan, n.d.).

Implementing machine learning into this work besides being very ambitious, it will bring a more personalized way of functioning since by creating an algorithm that predicts the values of power produced by the solar panels will allow the system to automatically schedule appliances not only facilitating the user's tasks but also reducing their need for intervention.

2.5 Knowledge-Based Systems

A knowledge-based system is considered a form of artificial intelligence that focuses on capturing the knowledge of human experts to support decision making. This system comprehends a problem-solving method and follow a few steps in its architecture which include a knowledge base and an inference engine.

The knowledge base contains a collection of information in a specific area of study. On the other hand, the inference engine gives insights on the information store in the knowledge base. Knowledge-based systems must also include an interface that allows the users to query the system and interact with it.

A knowledge-based system may differ from system to system according to its problem-solving approach, varying from encoding expert knowledge as rule-based systems, and case-based reasoning where the rules are substituted by cases. Cases can be defined as solutions to already existing problems that a case-base system might attempt to apply to solve a new problem.

These systems have various applications, such as health care, avalanche path analysis, industrial equipment fault diagnosis and even cash management, making it quite a versatile system (Moore, 2018).

In Figure 2.5-1 is represented the typical architecture of a knowledge-based system. The problem-solving method of this system consists of a knowledge base and an inference engine. The knowledge base holds a repository of data within a specific field of study. On the other hand, the inference engine extracts insights from the data stored in the knowledge base. Additionally, it includes an interface that allows users to access the system and engage with it interactively. The architecture might vary according to its problem-solving method and approach.

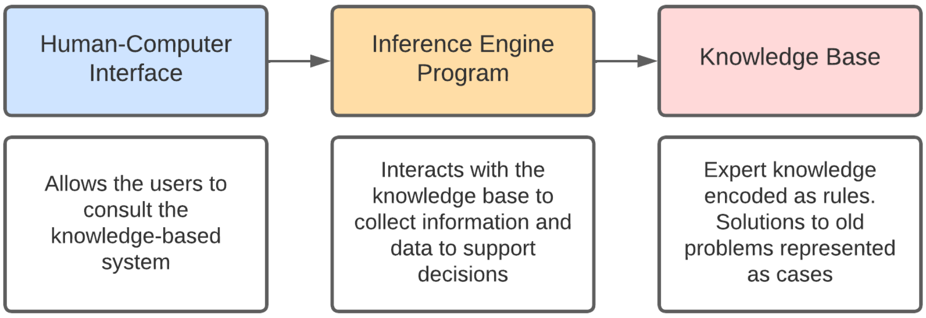


Figure 2.5-1- Architecture of a Knowledge-Based System. Modified from reference (Moore, 2018).

2.5.1 Rule-Based Systems

A rule-based system applies rules made by people, to store, sort and manipulate data, mimicking human intelligence. These systems follow a three-step process: receiving input data, analyzing it based on the rules, and executing actions determined by these rule-based decisions. Key characteristics of these systems include their ability to leverage human expert knowledge, represent knowledge declaratively, accommodate various knowledge paradigms, implement non-deterministic strategies, work with incomplete data, and focus on rule-driven decision-making.

When a statement is triggered, an action will be made according to the rules that were linked to that statement. For example, if you want to be able to handle 50 different actions, you will have to create 50 different rules according to each specific statement. If by chance you want to update a statement or an action, you will inherently have to write a new rule.

A rule-based system works accordingly to seven different basic components: the knowledge base, which stores the necessary knowledge for problem-solving; the database, which contains facts to compare against the rules in the knowledge base; explanation facilities, which clarify how conclusions were reached and which facts were considered; the user interface, which facilitates communication with the user; external interfaces for working with external data files and programs; the inference engine, which performs reasoning by linking rules to facts in the database; and working memory, a temporary storage for information and data.

These systems offer several advantages, such as cost-efficiency, stable outputs, low error rates, ease of understanding, rapid implementation, and quick results. However, they come with certain disadvantages, including the time-consuming nature of manual work, handling numerous variables and constraints, limited self-learning capabilities when compared to other learning systems and also create patterns that might by a little difficult to identify (Swaminathan, n.d.; *What Is a Rules Based System in AI?*, n.d.).

2.5.2 Case-Based Reasoning

Case-based reasoning simulates a human reasoning to solve new problems by reusing solutions that were applied in the past in similar situations and adapting their solutions. This process is iterative, starting with a retrieval step in which a new problem is compared with past case problems. The most similar problem and its corresponding solution are then identified from the stored cases. This process consists of two steps, recalling previous cases and selecting the best subset of them. If the solution that was found does not meet the requirements of the new problem, the next step to be taken is adaptation. In the adaptation process a new solution is created. Then, a received solution and a new problem come together to form a new case, that, later is incorporated in the case base during the final step, the learning step. Like this, this system evolves into a better reasoner as the capability of the system is improved since the stored experience keeps on growing (Avramenko et al., 2002; Tsujioka et al., 2012).

In this dissertation work, knowledge base systems will be a great aid to successfully obtain the wanted results since they help organize vital data about energy usage, appliances, weather, and solar panels. This information guides the SHEMS in making smart decisions in real-time. The knowledge base grows over time, learning from past data to improve the system continuously. They play a big role in the SHEMS, making wise choices to save energy and enhance the user experience, which is exactly what is aimed for in the dissertation work.

2.6 System Simulation Load Values and Real Time Values

In this dissertation work we aim to optimize the maximum energy produced by a solar panel by programming and scheduling home appliances to work during the times of maximum energy production. Therefore, we need to understand when the times of maximum energy production happen and how much energy is being produced. Thus, we will analyze existing algorithms and systems that simulate the load values predicted in a day and, with the help of a sensor that allow the user to access the values produced, consumed, and exported to the grid by the panel, and compare the values that were calculated prior to the real values obtain in real time.

Predictive Model

To predict future results, along the years, predictive models were developed and implemented. These predictive models can forecast the production of energy and therefore, can be of three different types, these being, physical models, statistic models and artificial intelligence and hybrid models. The physical models use physical factors, such as, terrain, pressure, and temperature to estimate, for example, the wind speed. This type of model does not have accurate results when it comes to short-term predictions (Lei et al., 2009). On the other hand, statistic models are based on statistic data, resulting in a reduction of cost of installation and implementation, and are also more suited for short-term predictions. Lastly, hybrid models result in the combination of various models, to reduce the error-rate and presenting more precise values. By combining physical and statistic models, it is possible to better adjust the quality of the forecast and the duration time of the prediction (Conceição, 2021).

Since many models already exist, in 2021, Conceição, a student in NOVA SST developed a predictive model with the intent of predicting the energy production in photovoltaic panels, considering temperature and radiation in the community in study. The main goal was to better understand the community in study, such as the number of habitations and their consumptions and later analyze if the prediction of the photovoltaic production satisfies the needs of the community energy wise and economical wise (Conceição, 2021).

This predictive model will be helpful to this dissertation work since we can apply this predictive model and complement the predicted values to the obtained real time values. This will lead to a lower error rate because we can predict the values of energy production and confirm them with the values obtained in real time with the sensor mentioned above.

In this case, the temperature predictive model presented an average absolute error of 3,4% and an average quadratic error of 0,59% in a 48h range, and the irradiance predictive model presented an average absolute error of 1,79% in a 24h range. Both models presented a low error rate, which makes it reliable since the values were so good. Since the main goal was to create a predictive model to calculate the energy production on a photovoltaic panel, with the predicted values of temperature and irradiance it is possible to obtain the predicted values of the electric power production of the photovoltaic panel (Conceição, 2021).

This predictive model is a good assistance to this work since it successfully predicts the values of energy production. Although the cases of study and contexts might be different, it can still be adapted to this project's needs and enrich it. With the help of the sensor and the predictive model, we can lower the error rate and optimize the energy usage according to the energy production by the panel, both in advance and in real time, turning this into an amazing approach.

3

Conceptual Framework

In this chapter, the fundamental ideas behind the *OptiSun* system are explored. OptiSun is the proposed name for this system, since the main goal is the optimization of the power produced by sun. This chapter covers the requirements needed to implement this dissertation work, the initial mockups, and the architectural design that have created this project. It is explained how *OptiSun* is supposed to work by analyzing the foundation of this Smart Home Energy Management System.

3.1 Requirements Identification and Verification

3.1.1 Context Illustration

The context of this work revolves around the development *OptiSun*. This system aims to optimize energy consumption in real time in homes equipped with smart appliances and photovoltaic panels for renewable energy generation.

There are several actors that play crucial roles in this system, such as the home user, the smart devices, the photovoltaic panels and even *OptiSun* itself.

The home user is the homeowner or resident of the smart home and plays a central role in using the monitoring the system. They interact with *OptiSun* through a Web Application Platform Interface. The home user is responsible for connecting the smart devices such as appliances, Heating, Ventilating and Air Conditioning (HVAC) systems, security systems, and others. On the context of this dissertation work, all the smart devices will be simulated and not in reality connected to *OptiSun* since that would make this dissertation work even more complex and since the time is limited, it was important to focus on the main goal, this being the energy management by using the solar power production predictive model and analyzing its contributes to the user.
Another actor that plays an important role in this work is the photovoltaic panels since these will be the main source of energy and will contribute to a better comprehension of renewable energy sources and their reliability and worth. Connected to the photovoltaic panels there

should be a sensor which will allow the user to have information on the real time solar power production. This should sensor permits accurate data collection to later simulate and train the predictive model to predict the solar power production in the future.

To better comprehend the functioning of the proposed system let's assume a user is about to leave for work in the morning and they check the Web Application Platform. They notice that the system has analyzed the real-time data from the photovoltaic panels and determined that there's a high value of solar energy production expected throughout the day. The user then schedules their dishwasher, air conditioning system, and pool pump to run during the sunny hours when they won't be at home. Due to *OptiSun*'s predictive capabilities, it ensures that these appliances only activate when there is a surplus of solar power available, reducing their reliance on the grid and minimizing energy costs. This scenario exemplifies the seamless integration and energy optimization made possible by a smart home equipped with *OptiSun*, providing convenience and cost savings for the user.

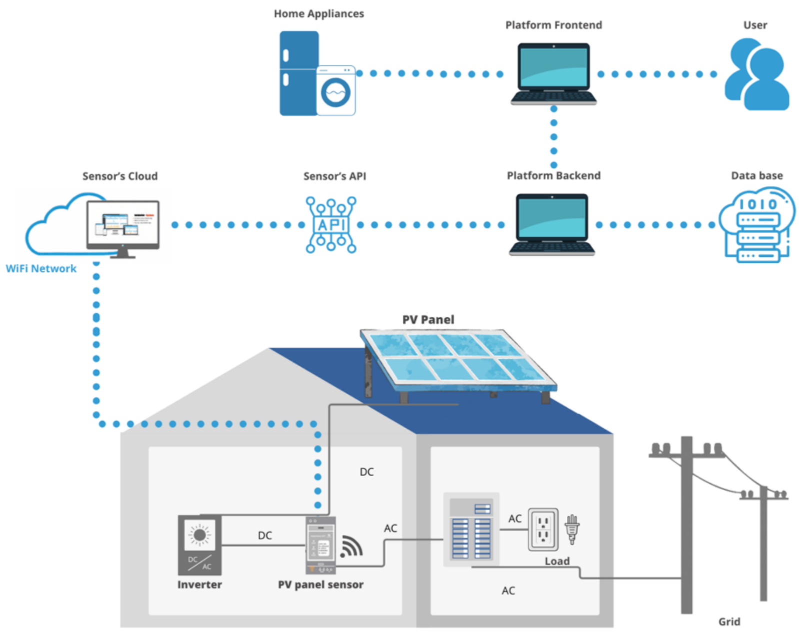


Figure 3.1-1 - Illustration of the *OptiSun* Approach

Figure 3.1-1 is a scheme that represents the relations that create *OptiSun*. It can be seen that the sensor, is connected to its cloud, which is connected to the API. To allow the *OptiSun* system to access the information on the photovoltaic panel in real time, the API is connected to the platform's backend, which will save the values gathered in a database. On the other end the platform's backend and frontend are connected to each other allowing the user to use the Web Application Platform proposed. Finally, the appliances are connected to the platform itself, allowing for the scheduling during the periods or higher energy production. The appliances' connection is only simulated.

3.1.2 Requirements

The identification of requirements from the end-user perspective is a crucial aspect in the process of system design and development. Having into account the system context presented above, a set of functionalities that the system should provide has been identified.

Table 3 - List of Requirements for the *OptiSun* system

Req.	***Description***
R1	*The system must allow an account creation and log in.*
R2	*The system must allow the configuration of several home appliances.*
R3	*The system must be the personalization of each user's needs.*
R4	*The system must allow manual scheduling of the home appliances.*
R5	*The system must allow automatic scheduling of the home appliances.*
R6	*The system must allow the user to see the home appliances' calendarization.*
R7	*The system must give the user access to decide whether they want the system to work manually or automatically.*
R8	*The system must allow the user to access the balance of loads.*
R9	*The balance of loads must present daily, weekly, monthly, and annual values.*
R10	*The system must allow the user access to the real time solar power production.*
R11	*The system must predict the solar power production values up to 7 days in the future.*

3.1.3 Identification of Risks and Constraints

As in any system to be specified, the definitions of its requirements naturally entail a certain degree of risks and constraints in subsequent implementation. In this work, the main identified risks are as follows:

- Technological Risks: The integration of various smart devices and technologies can pose challenges in terms of compatibility. Ensuring that all components work seamlessly together is crucial.
- Security: Given the sensitive nature of data in smart homes, safeguarding user privacy and protecting against potential security breaches are vital.
- Scalability: The system should be able to adapt to changing user needs and accommodate a considerable number of smart devices.

These identified risks will be carefully managed throughout this dissertation work to ensure the successful development of the *OptiSun* system.

3.1.4 Requirements Verification

Aiming at having an early verification of the requirements listed above, a set of mockups have been created. These mockups not only depict the visual appearance of the intended monitoring system but also illustrate how the user interact with the system. Below it is presented, a selection of the most important mockups. For a comprehensive collection of all designed mock-ups, please refer to Appendix A.1.

It is important to have an informative Home Page, in which the user could easily visualize the current weather conditions and the platforms notifications and alerts as can be seen in Figure 3.1-2, above. The main goal of both notifications and alerts is to inform the user whether an appliance can or cannot be powered by the energy produced by the photovoltaic panel during the time of its previous schedule. The user can also click the notifications and alerts and choose between canceling a schedule if there is not enough power being produced or schedule a new appliance if the power is indeed more than enough.

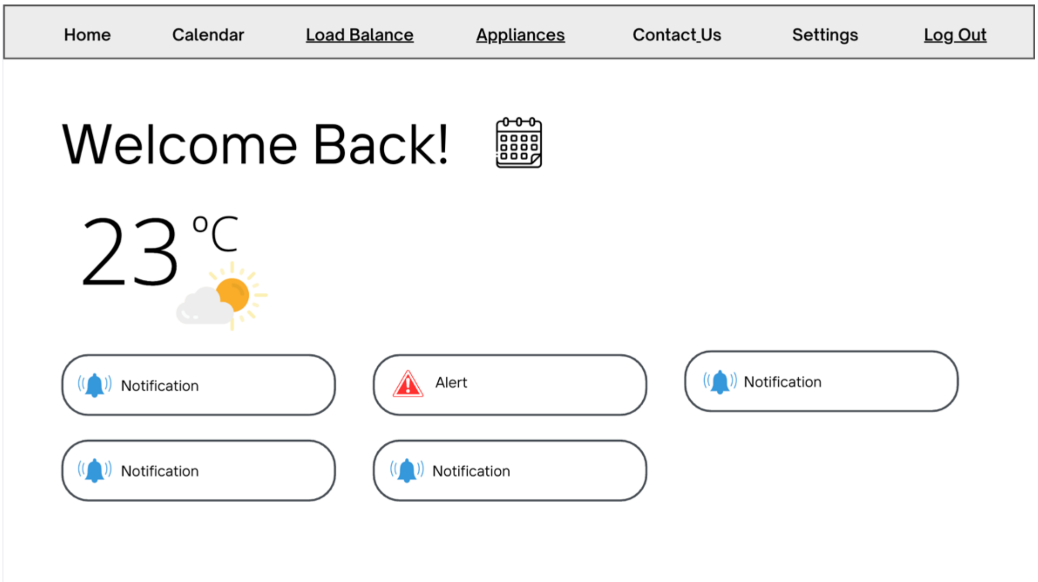


Figure 3.1-2 - *Home Page* logged in mockup

By clicking the calendar button next to the Welcome Back! header in Figure 3.1-2, the user is instantly redirected to the Calendar page, which can be seen in Figure 3.1-3, presented below.

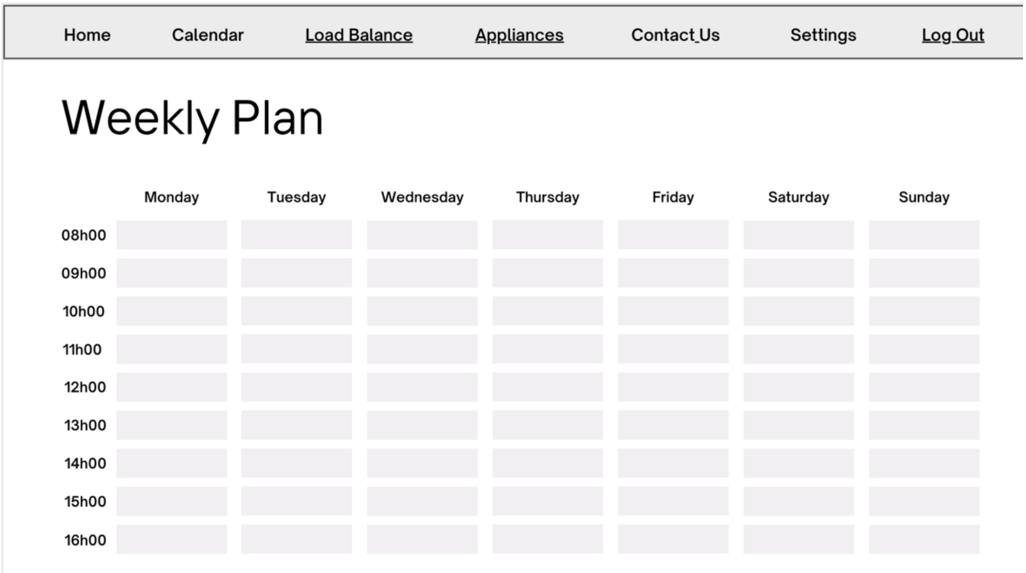


Figure 3.1-3 - *Calendar* page mockup

Once inside the *Calendar* page the user can analyze the *Weekly Plan* of the appliances' schedules. The user must be able to select a time slot and view which appliances are scheduled for that time slot. This calendar facilitates the user's job when scheduling appliances manually and contributes to a better planning and preparing of the appliances throughout the week in question.

One of the most important aspects of the system is its settings. By predefining settings, the system can work more accordingly to the users' needs and allow the platform to be a little more customizable for each user.

To allow a more personalized platform for each user, the settings parameters were considered:

- Mode
  - Manual
  - Automatic
- Season
  - Spring
  - Summer
  - Autumn
  - Winter
- Maximum number of appliances to schedule at once
- Starting Hour
- Finishing Hour
- Tariff
  - Price per kWh

The *Mode* is to define whether the user desires the system to automatically schedule appliances for them. On the other hand, the *Season* is to define how much power produced is the minimum required to start scheduling appliances. This is important to add to settings since along the year, the energy produced in the various seasons is different and therefore, it cannot be expected for example, in Portugal, that the energy produced in the winter is equal to the energy produced in the summertime. In the summertime the power produced by the panel will be substantially higher, therefore, the minimum required power being produced for an appliance to be automatically scheduled can be higher. On the other hand, since in the winter the power production levels are lower, it can't be expected that an appliance will continuously be fully powered by the photovoltaic panel. It can also be useful in other parts of the world where the seasons' weather conditions are more similar or for example in other parts of the world where the hotter season happens in a different time period.

The *maximum number of appliances to schedule at once setting* allows the system to only schedule a specific number of appliances at the same time, to avoid overusing the power being produced by the panel. By defining a maximum number of appliances to be scheduled, neither the user themselves manually nor the system automatically will be able to schedule more appliances at the same time than the defined value in the settings parameter. This value must be equal or lower to the number of appliances connected to a user.

Finally, in the *General Settings* tab, there is the *Starting Hour* and *Finishing Hour* which represent the starting and finishing time the system and user will be able to schedule the appliances.

When it comes to the *Tariff Settings,* there is only one setting which is the *Price per kWh*. This setting represents how much the user is currently paying for the energy consumed from the grid. This will allow the calculation of the saved money by using the web application platform, which can be seen in the *Load Balances* page, explained further along in this chapter.

In Figure 3.1-4 it can be seen the view of the *Settings* page. For more details on the *Tariff Settings*, check appendix A.1.

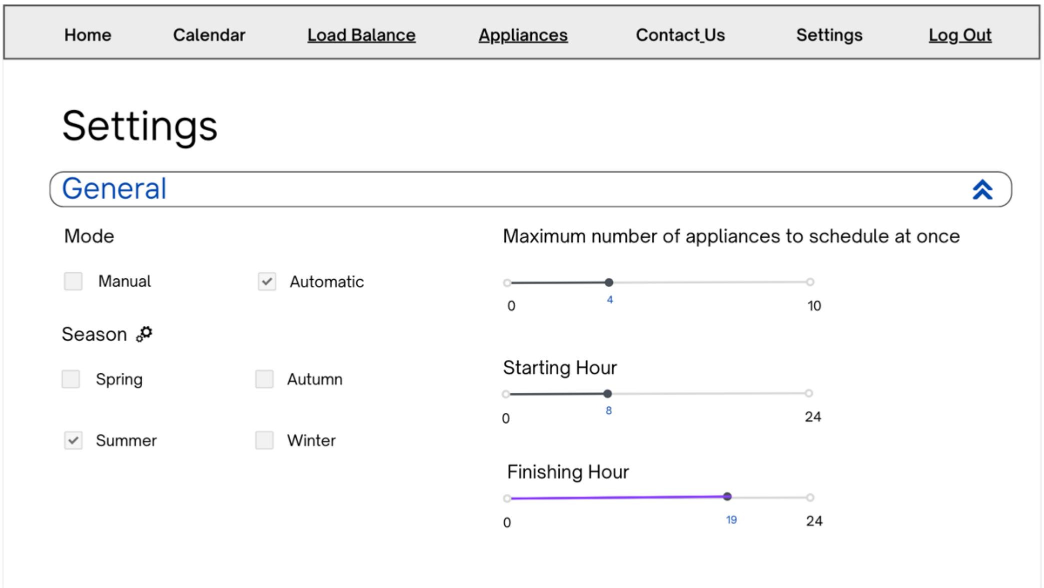


Figure 3.1-4 - *Settings* page mockup

Once the settings are defined, the user must start connecting their appliances to the system. This process is done in the *Appliances* page, which can be seen in Figure 3.1-5, presented below.

To connect an appliance the user must press the + button and select which appliance they wish to connect. After connecting an appliance, the user can change the appliance's settings by pressing the desired appliance's button and will be redirected to that appliance's page settings as can be seen in Figure 3.1-6.

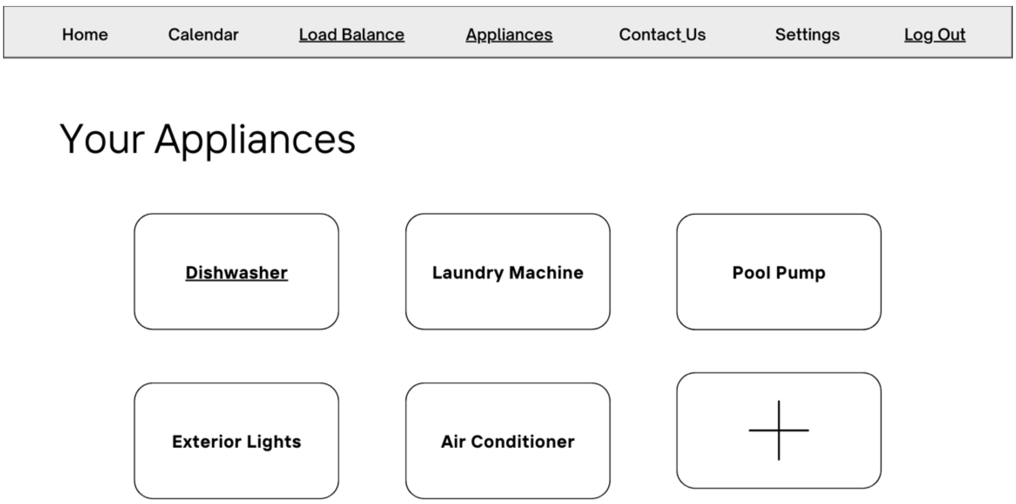


Figure 3.1-5 - *Appliances* page mockup

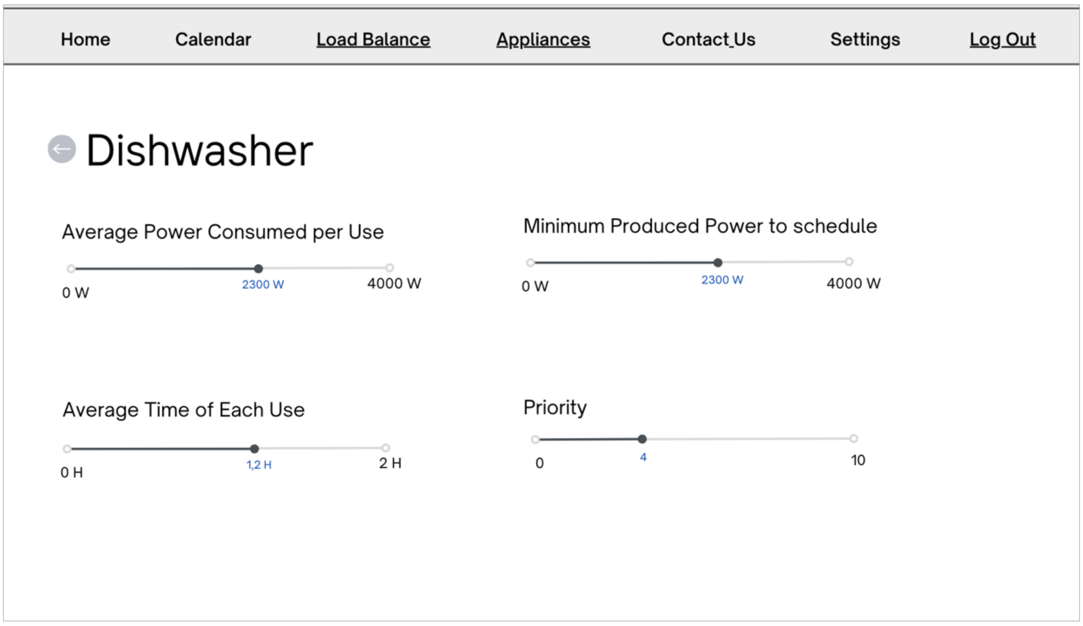


Figure 3.1-6 - Appliance's settings mockup

When accessing an appliance's settings page, the settings presented are:

- Average Power Consumed per Use
- Average Time of Each Use
- Minimum Produced Power to schedule
- Priority

The *Average Power Consumed per Use* setting represents the power that the chosen appliance will consume on average per use. This will allow the system to schedule the appliance according to how much power the appliance will consume. This setting will also allow a more accurate balance of consumed loads.

On the other hand, the *Average Time of Each Use* is related to how much time the appliance will be functioning on average, every time it is scheduled. This will also contribute to a more precise calculation on the loads balance.

The *Minimum Produced Power* to schedule is related to the fact that if the appliance can't be fully power by the energy produced by the photovoltaic panel, the appliance can still be scheduled as long as the minimum power defined is indeed being produced by the photovoltaic panel.

Finally, the *Priority* setting represents how important that appliance is to the user. If an appliance has a higher value of priority, the system will schedule that appliance first when executing an automatic schedule. Therefore, the system will schedule the appliances from highest priority to lowest.

The final functionality in the Requirements Elicitation is the *Load Balance* page, as can be seen in Figure 3.1-7.

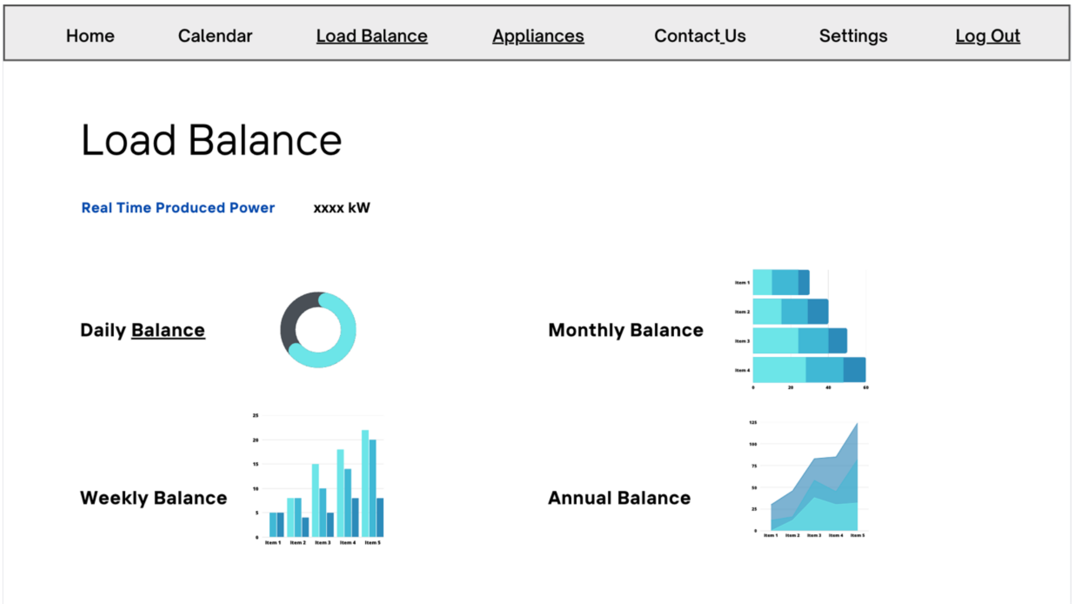


Figure 3.1-7 - *Load Balance* page mockup

The *Load Balance* page provides the user a visualization of not only the *Real Time Produced Power* but also the *Daily Balance, Weekly Balance, Monthly Balance,* and *Annual Balance*.

By pressing any of the balance buttons the user is automatically redirected to the balance chosen. By choosing for example, the *Daily Balance* button, the user is redirected to a page like the one presented in Figure 3.1-8 below.

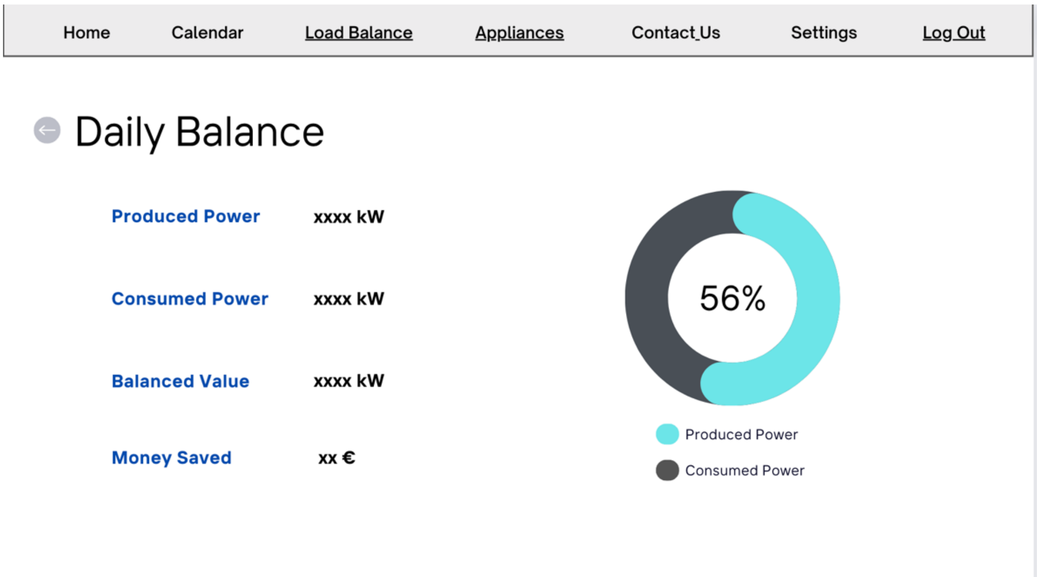


Figure 3.1-8 - *Daily Balance* page mockup

Any of the balance pages allow the user to visualize the *Produced Power, Consumed Power, Balanced Value* and the *Money Saved*. The *Produced Power* is the total amount of power produced by the photovoltaic panel in that period of time, in this case, in that day. The *Consumed Power* represents, on the other hand, the amount of power consumed by the appliances connected to the platform. The *Balanced Value* is none other than the subtraction between the *Produced Power* and the *Consumed Power*, which, if greater than zero, means that the user produced more power than consumed, and therefore, saved energy and money. The *Money Saved* is calculated by multiplying the *Consumed Power* by the *Price per kWh*, predefined in the *Settings* page. This will allow the user to know how much money they saved by using the web application platform.

All the load balances enable users to visualize the power production and consumption throughout the day, week, month, and even the year. This functionality helps users grasp the periods of peak power production and consumption. It also assists in planning appliance usage to maximize energy savings and subsequently, cost savings. This functionality will directly provide the user the most important information on whether the web application platform is or isn't facilitating the energy saving process and if it is well functioning.

In conclusion, gathering the important requirements of the application to develop, is a crucial step that allows the developer to go from a blank page to a structured mapping of functionality details and even visual aspects.

3.2 System's Architecture

Having in mind the system's requirements, a high-level architecture is proposed, as illustrated in Figure 3.2-1.

The first level of the architecture is the Support Infrastructure. This layer forms the foundation of the system, providing necessary resources and connections for the entire program to function seamlessly. This layer covers three different sub- layers:

- The Connection to devices that establishes and manages connections between the program and various devices, in this case, the user's appliances and the respective photovoltaic panel. It ensures that the data from the devices, such as power production data from the photovoltaic panel, is efficiently transmitted to the higher layers for processing and analysis.
- The Database Connection, represents established connections to databases where essential data is stored. This could include historical power production data, appliance usage patterns, and user preferences. These connections facilitate the retrieval and storage of critical information needed for scheduling appliances effectively.
- APIs (Application Programming Interfaces) provide a standardized way for different software components to communicate and exchange data. In this sublayer, APIs are chosen and defined to allow different layers of the architecture to interact with one another.

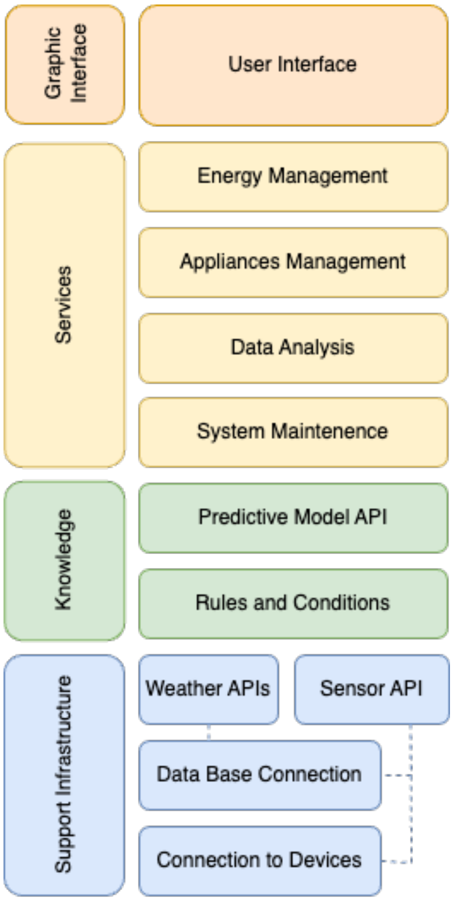


Figure 3.2-1 – High-Level System's Architecture

The second layer, the Knowledge Layer is responsible for processing and analyzing data to make informed decisions about scheduling appliances based on the power production of the photovoltaic panel. It consists of two sublayers:.

- The first sublayer, Rules and Conditions, defines the rules, and conditions that manage how appliances should be scheduled. These rules consider factors such as power availability, appliance priorities, time of day, and user preferences. By incorporating these rules, the system ensures that the appliance scheduling aligns with specific criteria.
- On the other hand, the Predictions sublayer uses historical data retrieved from the APIs and real-time information from the photovoltaic panel. This sublayer generates predictions through a machine learning model about future power production. These predictions are crucial for the system to anticipate the energy supply and make proactive decisions about which appliances can be scheduled without compromising energy availability, allowing the user to save both energy and money, and facilitating their life.

Continuously there is the Services layer. This layer comprehends the core functionalities that enable the actual scheduling and management of appliances. It encompasses the following four sublayers:

- The System Maintenance sublayer handles routine maintenance tasks necessary for the smooth operation of the system. It could involve monitoring the quality of the software, performing updates, and ensuring that the system remains operational and reliable.
- In the Data Analysis sublayer, the system performs in-depth analysis of not only power production data, but power consumption data and saved energy and money data. It also analyzes predictions from the Knowledge Layer. This analysis informs the decision-making process and helps optimize appliance scheduling strategies.
- The Appliances Management sublayer, is responsible for managing the status of appliances. It receives scheduling instructions from the higher layers and communicates with the devices to activate or deactivate appliances according to the determined schedule.
- The Energy Management sublayer focuses on efficiently allocating available energy to appliances. It coordinates with the Knowledge Layer to align appliance schedules with predicted power production, ensuring that appliances are used in an energy-conscious manner.

The final layer is the Graphic Interface. This layer provides a user-friendly front-end that allows users to interact with the system. This layer includes only one sublayer, the User Interface. This is the visual and interactive component of the system that users directly interact with. It presents information about appliance schedules, power availability, and system status. Users can introduce preferences, view recommendations, and potentially manual scheduling decisions through this interface.

In summary, the proposed system's architecture is designed to intelligently schedule appliances based on the power production of a photovoltaic panel. Each layer serves a distinct purpose, from establishing connections and processing data to making informed scheduling decisions and presenting information to users. This structured architecture ensures efficient energy usage and provides an enhanced user experience when it comes to energy saving.

4

IMPLEMENTATION

In this chapter, we step into the world where ideas turn into action. This section explains how the Web Application Platform was developed and implemented, the technologies that were carefully chosen, and the automation that brings it to life and achieves the main goal. This is where concepts mentioned earlier become real, giving users better control over their energy usage for a more eco-friendly and more cost-efficient lifestyle.

4.1 Technologies Used in Implementation

The selection of the technology to be employed in the implementation of the proposed system is a critical aspect as it forms the backbone upon which the entire work is built. In this way, Figure 4.1-1 provides a comprehensive overview of the specific technologies chosen to facilitate the implementation of the different layers outlined in the proposed architecture.

Overall, the implementation approach relies on the development of a Web application based on the Model-View-Controller (MVC) design pattern. The MVC separates the application into three components, these being, Model (data and business logic), View (user interface) and Controller (handles user input and updates the model). This separation of concerns makes the application more organized, maintainable, and easier to navigate and use. For that, it was used a Visual Studio's project type called Model-View-Controller (MVC) .Net Core project.

It also needed to be a .NET Core platform since it can be defined a cross-platform, open-source framework for building modern applications proving a large set of libraries and tools for web development, making it easy to create robust and high-performance applications.

When it comes to the security part of the application, the .NET Core platforms can leverage built-in security features like identity, which simplifies user authentication and authorization. Identity provides secure password hashing, token-based authentication, and role-based access control which was a crucial step since the platform developed required a user registration and log in. Considering the password hashing was made automatically it simplified the process since it did not require any more time investment on that end.

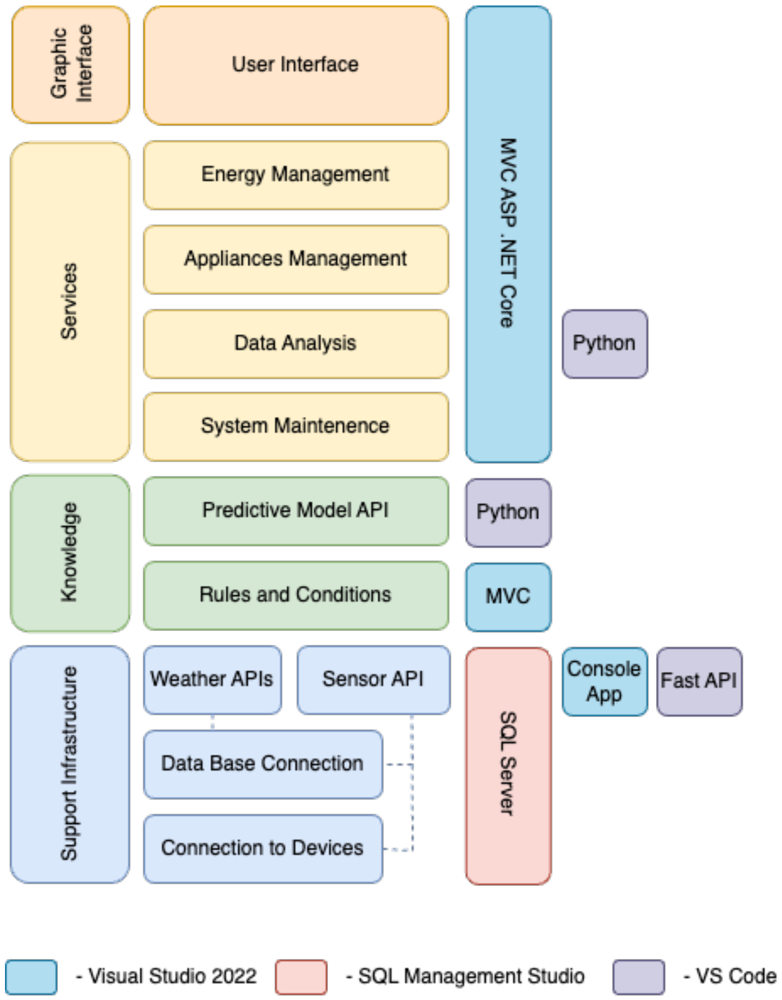


Figure 4.1-1- System's high-level architecture with corresponding chosen technologies

Besides all these important features, considering the vast collection of NuGet packages and their ease of installation and integration into the project, establishing a connection to the database priorly created was simple saving development time and ensuring that it was using well-tested, reliable solutions.

Finally, one of the main reasons that "sold" the choice of using Visual Studio was its testing capabilities, since it was quick and straightforward to run tests, ensuring that the application was reliable, functional. In addition to these benefits, the amount of Documentation available was incredible, making it easier and faster to learn and understand the framework while working.

Choosing Visual Studio for development led to the selection of SQL Server as the database software. SQL Server is well-considered for its versatility, security, and performance. It works seamlessly within the Microsoft ecosystem, which includes Visual Studio. To manage this database, we used SQL Server Management Studio (SSMS). SSMS not only helped integrate the database with our Visual Studio project but also provided a strong environment for designing and managing the database, generating SQL scripts, running complex queries, and working smoothly with Entity Framework. This choice made the development process more efficient, ensuring our application handles data well.

Azure was chosen as the cloud platform for its seamless integration with the chosen development tools, Visual Studio and SQL Server. This integration simplified application development, testing, and deployment, creating a consistent development environment. Azure's ability to adjust data allocation in real-time aligned well with the project's dynamic data needs. Its reliability, rigorous security measures and versatile data storage options made the Azure platform a good choice.

To create the predictive model, I chose to work with VS Code and Python due to my previous experience with these tools, finding them efficient for data cleaning, model training, and testing. Additionally, for exporting the predicted values, I opted for an API developed using the FastAPI framework. This choice was made due to the framework's compatibility with Python, making the entire process more straightforward and seamless.

In Figure 4.1-2, a diagram of all the connections is presented where not only the data base, but also the Platform backend developed and the service that will deploy the APIs is connected to the Azure Cloud.

The sensor used in this dissertation work to measure the values produced by the panel was the IAMMETER sensor. This sensor is installed in the user's home by connecting the photovoltaic panel's electrical panel to the energy meter on the sensor. After this is complete, the sensor sends the information on the panel through Wi-Fi connection via its antenna, allowing the user to have access to the information whether by using the IAMMETER sensor mobile application or by accessing the IAMMETER API, which allows for a creation of a third party application, which is proposed in this dissertation work.

In the diagram it can be observed that the IAMMETER sensor is connected by a meter to the photovoltaic panel. This sensor will gather the production data needed for the predictive model and to calculate the balance of loads, and export it through Wi-Fi Connection to IAMMETER cloud. When saving the data to cloud, the IAMMETER platform will allow the exportation of data for third party users through their API, which, therefore, will be connected to the Platform backend, allowing the visualization of data in real time. This API is also connected to the API deployment Service created that will allow the data storage to the database created. This API deployment Service is subsequently connected to the Azure platform since it is where it is deployed to allow 24/7 data storage in the data base.

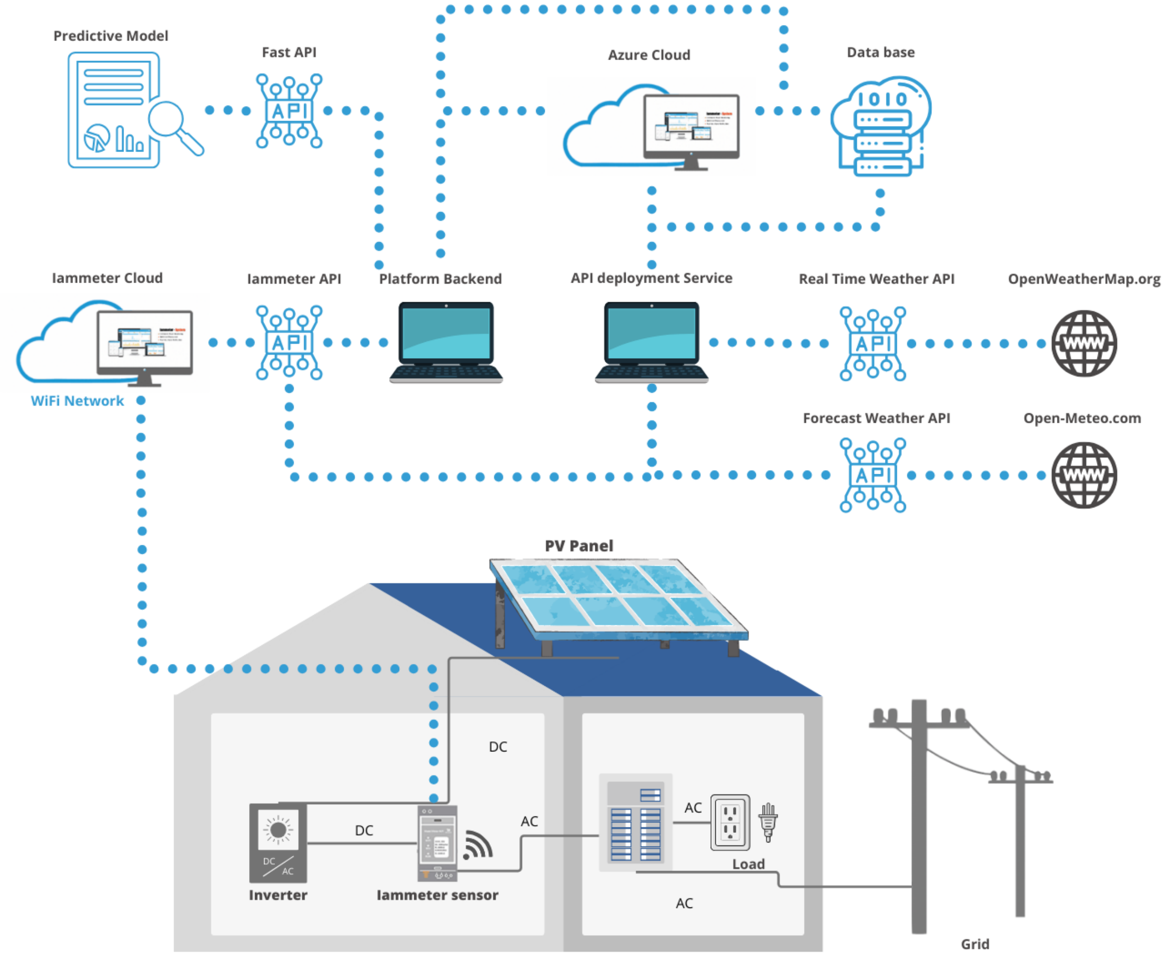


Figure 4.1-2 – System's implementation architecture

Continously, there are two more APIs that are connected to the API deployment Service aswell: the Real Time Weather API, retrieved from OpenWeatherMap.org and the ForecastWeather API, from Open-Meteo.com. Both these APIs will store weather conditions information in the data base continuously with the aid of the Azure Cloud and the API deployment Service created.

Finally, the Platform backend has access to all information in the Azure Cloud, since all data is needed to predict the energy produced by the panel, through the predictive model. The predictive model is connected to the Fast API, which will export the data values to Platform backend, to further automate the home appliances scheduling.

In conclusion, choosing developing tools that facilitate the connection between each other such as Visual Studio and SQL Server and even later Azure, is an important aspect to allow the developer to not only be more concise but also work a little smarter, ensuring the process is simplified and concise.

4.2 Database

In the fast-paced and ever-evolving landscape of web development, developing a successful web application platform demands meticulous planning and strategic decision-making. Among the countless components that constitute our platform, the database stands out as one of the most crucial and foundational elements. This chapter dives into the implementation of the database and emphasizes its vital importance as the bedrock of the web application platform.

The database, serves as the central repository for all application data. Its key role in managing the storage, retrieval, and manipulation of data for various functionalities emphasizes its significance. A well-designed database infrastructure is integral to the smooth functioning of the platform, influencing performance, security, and scalability. By placing the database implementation at the forefront of the development process, it is ensured that an optimized data management is present, and the data maintains its integrity, facilitating efficient data access, ultimately contributing to an enhanced user experience. This proactive approach helps prevent potential inconsistencies that might arise later in development, providing a solid foundation upon which to build and expand the web application platform. However, along the process of developing the whole web application platform, changes and alterations were made to the data base to match the new necessities of the platform, since new ideas arose from time to time, making the data base creation more fluid and consistent overtime.

Database Modeling

The data base created revolved around the creation of a user and its appliances. Considering that each user could have many different appliances, a relationship was made between both, to ensure that this condition was well defined. Besides that, to accomplish the web application's main goal of scheduling the homes appliances automatically, an Appliance Schedule table was also needed to save the dates of schedules as well as the consumed power per scheduled, to better comprehend the savings and consumptions, and the actual importance and worth of the platform developed during these past months.

In Table 4, the functions of each table are briefly explained, and the foreign keys presented. This table summarizes why each table was created. However, each table will be explained more in depth further ahead in this chapter.

In Figure 4.2-1, it is possible to better analyze each table and the relationship between them. In this Entity-Relationship Diagram (ERD) we can see that the column *UserId* in the table **Appliance** is a Foreign Key that refers to the column *Id* in the **AspNetUsers** table. Similarly, the column *ApplianceId* in the **ApplianceSchedule** table is too a ForeignKey corresponding to the *ApplianceId* column in the table **Appliance**.

Table 4 - Database tables and functions

Table	Function	Foreign Keys
AspNetUsers	Used to store user's account settings.	-
Appliance	Used to store appliance's information.	*UserId*
ApplianceSchedule	Used to store data on each appliance's schedules.	*ApplianceId*
Settings	Used to store system's settings.	*UserId*
SeasonSettings	Used to store the settings related to each season.	*UserId*
DailyBalance	Used to store the daily balance of loads.	*UserId*
WeeklyBalance	Used to store the weekly balance of loads.	*UserId*
MonthlyBalance	Used to store the monthly balance of loads.	*UserId*
AnnualBalance	Used to store the annual balance of loads.	*UserId*
PanelReading	Used to store the data gathered from the photovoltaic panel power production values in real time through the IAMMETER sensor's API.	*UserId*
RealTimeWeather	Used to store the real time weather conditions values gathered from the openweathermap.org API.	*UserId*
ForecastWeather	Used to store the forecasted weather condition values gathered from the open-meteo.com API.	*UserId*

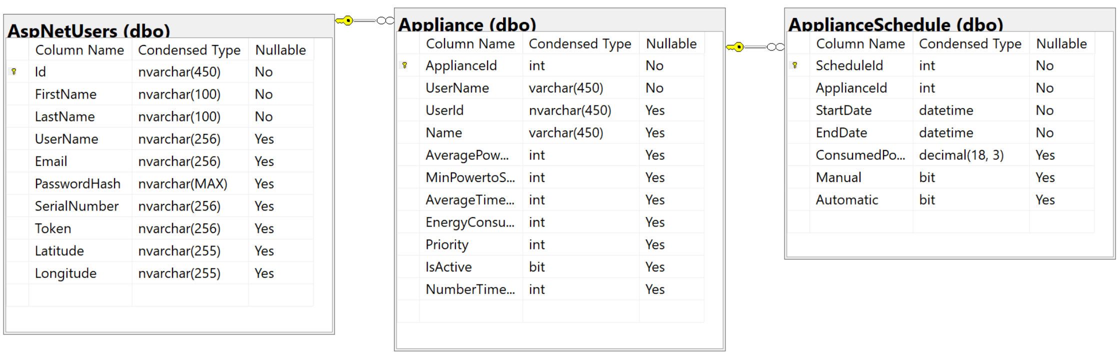


Figure 4.2-1 - ERD comprising the tables **AspNetUsers**, **Appliance** and **ApplianceSchedule**

This was developed this way, since, as explained above, each user will be able to have one or more appliances corresponding to itself and, furthermore, each appliance will have one or more schedules corresponding to them as well.

The table **AspNetUsers** was created automatically when implementing the Registration and Log In functionalities with the aid of Visual Studio 2022 functionalities of the chosen project. Even though this table was automatically created with most of its columns, new columns were added considering the platform's requirements and needs, such as *Serial Number*, *Token*, *Latitude* and *Longitude*. These columns will be explained further ahead in this document.

When diving into the creation process of the database and the platform itself, it was necessary to create a table with the settings (**Settings**) of the platform and the settings of each season (**SeasonSettings**), which we will get more in depth with further along in this document.

Both these tables have a column *UserId* which is a Foreign Key referring to the Id column in **AspNetUsers** table as it can be seen in Figure 4.2-2.

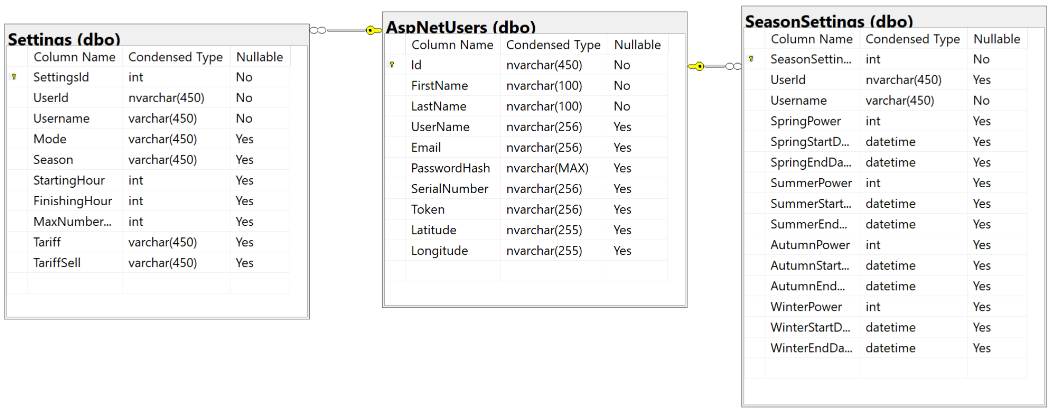


Figure 4.2-2 – ERD of tables **AspNetUsers**, **Settings** and **SeasonSettings**

To ensure that the user can access the Loads Balance functionality during their time using the platform, several tables were created to save and show the user the Load Balance during the day, week, month or even year. The relationship these tables have with the **AspNetUsers** table is the same as the ones above, related by the *UserId* as seen in Figure 4.2-3.

To obtain both the current and predicted power production values of the photovoltaic panel, we needed to access multiple APIs. These included an API providing real-time data from the photovoltaic panel, another with current weather conditions data, and a third forecasting weather conditions up to 16 days in advance. To store these fetched values (for future analysis), three tables were created: **PanelReading**, **RealTimeWeaather** and **ForeCastWeather**.

These tables have a relationship with the **AspNetUsers** table through the *UserId* of each and Id column in **AspNetUsers** as can be seen in Figure 4.2-4.

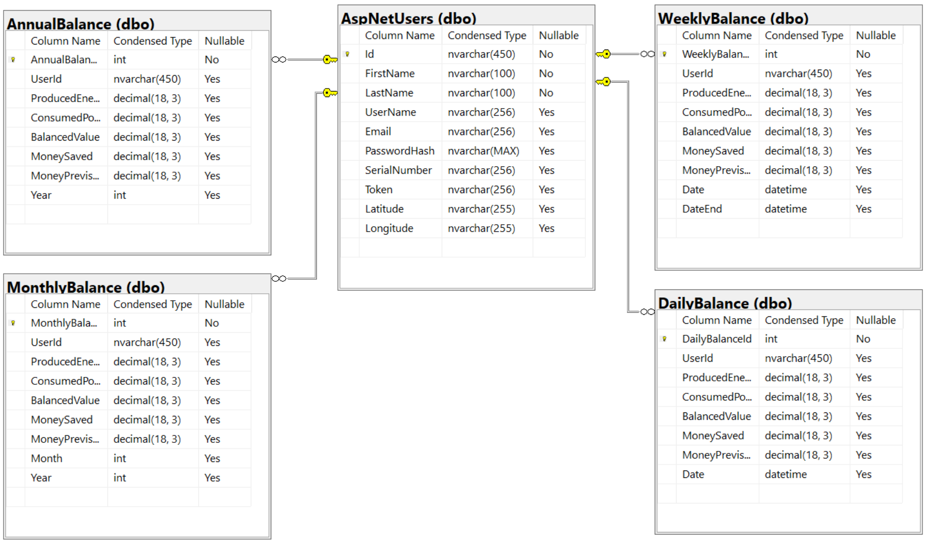


Figure 4.2-3 - ERD of tables **DailyBalance**, **WeeklyBalance** and **MonthlyBalance**, **AnnualBalance** and **AspNetUsers**

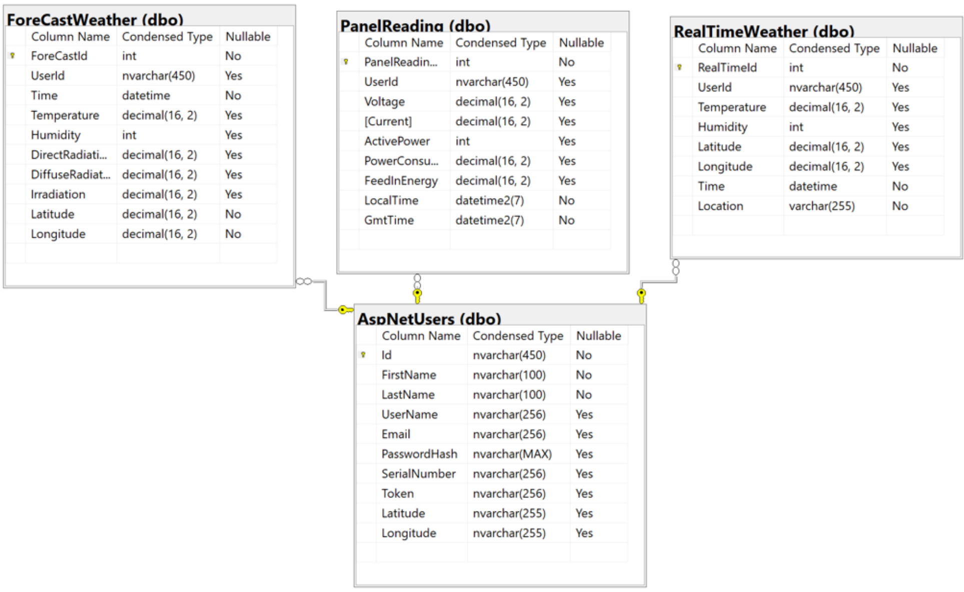


Figure 4.2-4 – ERD of tables **PanelReading**, **RealtimeWeather** and **ForeCastWeather** and **AspNetUsers**

To implement the database, it was used the software SQL Management Studio since it could easily connected to the Visual Studio 2022, which was used to develop the Web Application, through a Model-View-Controller (MVC) .NET Core project and deploy the database to Azure so it could be easily accessible.

4.3 APIs (Application Programming Interfaces)

API stands for "Application Programming Interface." It refers to a set of defined rules and protocols that enable different software applications or components to communicate and interact with each other (tray.io, n.d.). APIs allow developers to create applications that can use the features and data provided by other applications or services without requiring an in-depth understanding of their internal workings.

APIs provide a standardized way for developers to request specific functionalities, data, or services from external sources (Thomas Fielding, 2000). This standardization promotes componentization and seamless integration between diverse software systems, simplifying the development of intricate applications by utilizing pre-existing components.

In this dissertation work, APIs from different sources are used to analyze power production values from a photovoltaic panel and predict these values based on weather data as explained in the previous section.

The IAMMETER Sensor API is used to analyze the active power values produced by the photovoltaic panel. This API provides real-time data regarding the power output of the photovoltaic panel. By integrating this API into the system, accurate and up-to-date information about the photovoltaic panel's performance is accessed.

The real-time weather API data was retrieved from openweathermap.org. This API provided real-time weather data, including information such as temperature, humidity, and other meteorological conditions in a given location, this being the same as the photovoltaic panel's location. By incorporating this data into the system, correlations could be made with the active power values produced by the photovoltaic panel. This correlation helped in identifying patterns and relationships between weather conditions and power output.

The forecast weather API from open-meteo.com allowed access to predicted weather data for future time periods. This data included metrics such as diffuse radiation, direct radiation, irradiance, temperature humidity and respective date and time in a given latitude and longitude, similar to the photovoltaic panel's location. By utilizing forecasted weather data, a machine learning model is created that predicts the values of active power production by the photovoltaic panel. The model uses various weather parameters as features to make accurate predictions about power output under different weather scenarios.

By combining real-time and forecasted weather data with the power production values obtained from the IAMMETER Sensor API, a comprehensive understanding of how weather conditions impact the performance of the photovoltaic panel is developed. This information is then used to create a machine learning model capable of predicting power output, which is a valuable tool for optimizing appliance scheduling based on energy availability.

In conclusion, APIs play a pivotal role in this dissertation work by enabling seamless integration of data from different sources. They allow access to real-time and forecasted weather data as well as power production information, facilitating the development of a predictive model for optimizing energy usage.

Table 5 - APIs' information

API	API source	Fetching time interval	Data type
PanelReading	IAMMETER	1 minute	Produced Power
RealTimeWeather	openweathermap.org	1 minute	Weather condition values
ForecastWeather	open-meteo.com	1 hour	Weather condition values

4.4 Platform Web Application

In this section, the design and implementation process of the Web Application that gives support to this dissertation WORK is presented.

The design and development of the *OptiSun* system is composed of six main operative components, represented in Figure 4.4-1, each with their specific functionalities and methods for processing and managing data. The following sections provide an overview of these components.

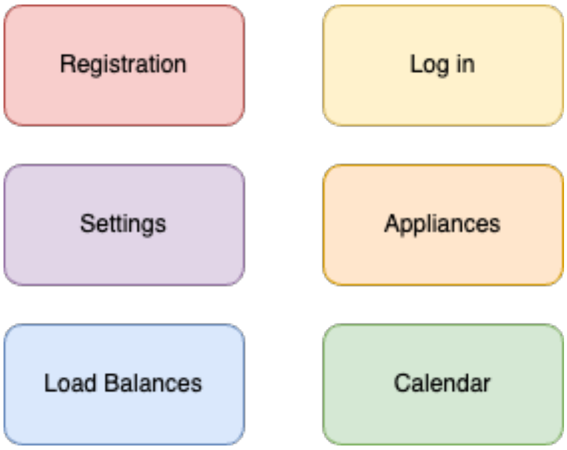


Figure 4.4-1 - Operative components of the developed system

4.4.1 Registration

One of the first processes to implement is the user registration. The registration or Sign-Up process is where a new user creates their account with their own specifications such as first and last names, email, username, and password.

This step in the platform development is important since it enables the experience for each user to become increasingly customized. Later the settings and appliance's settings will be all according to the user's preference.

This functionality was created using the *Identity* package, which contains a set of libraries and tools which simplifies the implementation of user authentication and authorization features and is commonly used for user management, registration and login.

To successfully create an account, all the parameters of the form are required to be filled and both the username and email must be unique, meaning that if a user tries to create an account with an already taken username, an error message appears, and a new username must be chosen. Besides the restrictions with picking both a unique username and email, when deciding on the password, it must include minimum of 8 characters, composed of a combination of letters, numbers, and symbols, to ensure the maximization of security for the user.

To access the SQLServer database and write in it, a SQLServer package which is designed to store, manage, and retrieve structured data efficiently was used to access the created database. This package is used throughout the whole implementation process and in every functionality the needs access to the database, whether to delete, create or update rows in a database table. The process of the implemented Registration process represented in Figure 4.4-2. The presented sequence only happens if a chosen username is not yet taken by another user.

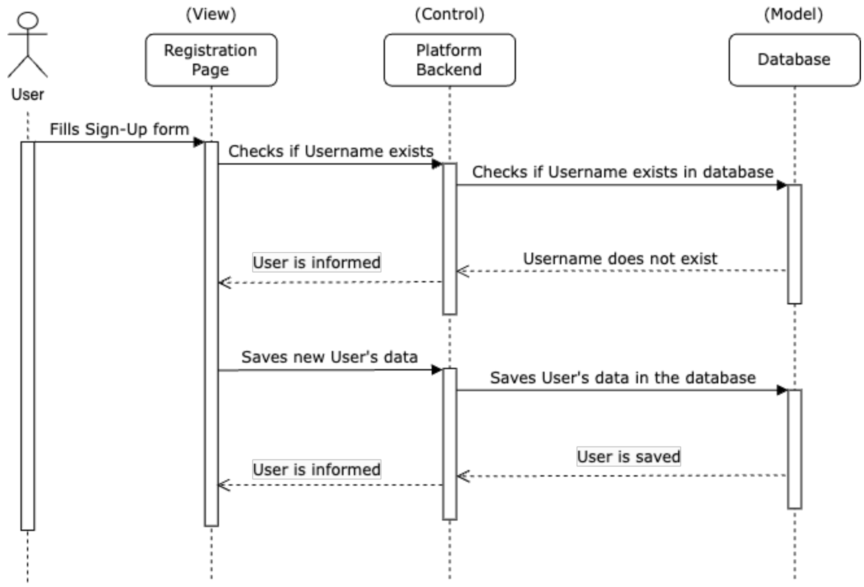


Figure 4.4-2 – UML sequence diagram of the Registration process

4.4.2 Log In

The log in is executed by submitting the username and password. An interesting feature that was automatically added by the Visual Studio project Identity package was the *Remember me?* Checkbox since when checked, it prevents the user from always having to log in every time they want to use the web application and therefore, whenever the user uses the platform after submitting the checkbox, they are already logged into their account, saving a few minutes to the user.

In Figure 4.4-3 is presented the process of a successful Log In. However, if the Log In is unsuccessful due to the username or password being incorrect, the user is informed and does not enter the account, being allowed to try again.

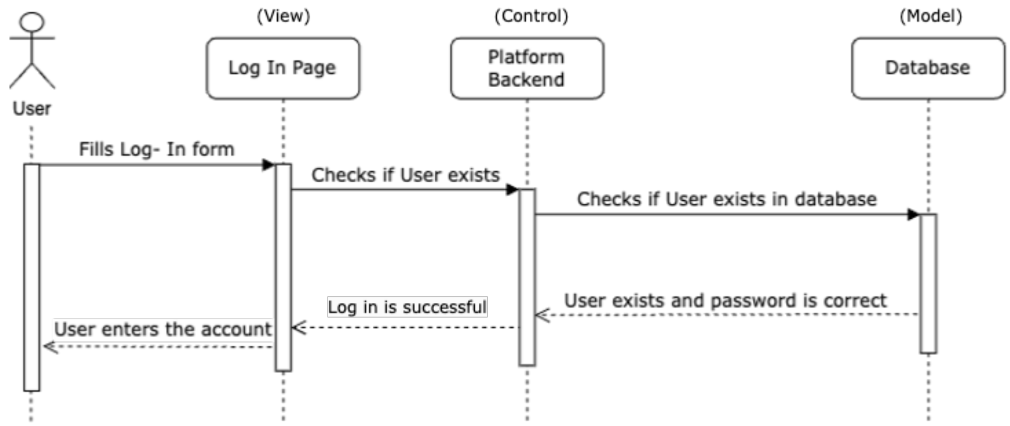


Figure 4.4-3 – UML sequence diagram of the Log In process

4.4.3 Settings

The Settings functionality is dedicated to define the settings of the *OptiSun* monitoring system. This functionality is where the *OptiSun*'s settings are saved to the **Settings** database table. Similarly to the Registration process, the Settings functionality consists of filling and selecting several parameters which once saved, will update the **Settings** database table corresponding to its designated User.

There are several parameters for setting the system, and some have conditions that need to be followed, as illustrated in Table 6:

Table 6 - Settings' parameters and conditions

Parameter	**Conditions**
Mode	Must be Automatic or Manual.
Season	Must be either Spring, Summer, Autumn, or Winter.
Maximum Number of Appliances to schedule at once	Must be greater or equal than 0 and smaller or equal to the number of appliances connected to the system.
Starting Hour	Must be between 0 and 24.
Finishing Hour	Must be between 0 and 24 and greater than *Starting Hour*.
Purchase Tariff €/kWh	-
Sales Tariff €/kWh	-

On the other hand, there is also a Season Settings functionality which defines settings for each season of the year. Each season will also have a beginning and end date and a parameter that refers to the minimum power produced by the panel to schedule an appliance. This means that if the minimum power defined for each season is being produced, the system will still schedule an appliance even if it does not cover all the power consumptions of the appliance. This was implemented considering that the production values in all the seasons are different, so if the system would only schedule if there was enough power to fully cover the appliance's consumption, the system would most likely not schedule during the colder and cloudier seasons. Defining this condition allows the user to still save energy and money since the schedules will still happen even though the power production is reduced. The money and energy savings will also decline during the winter and autumn, but the savings will be greater than zero, which is most definitely a greater option.

The Season Settings functionality, once modified, updates the **SeasonSettings** database table, presented previously in chapter 0 - **Database Modeling**.

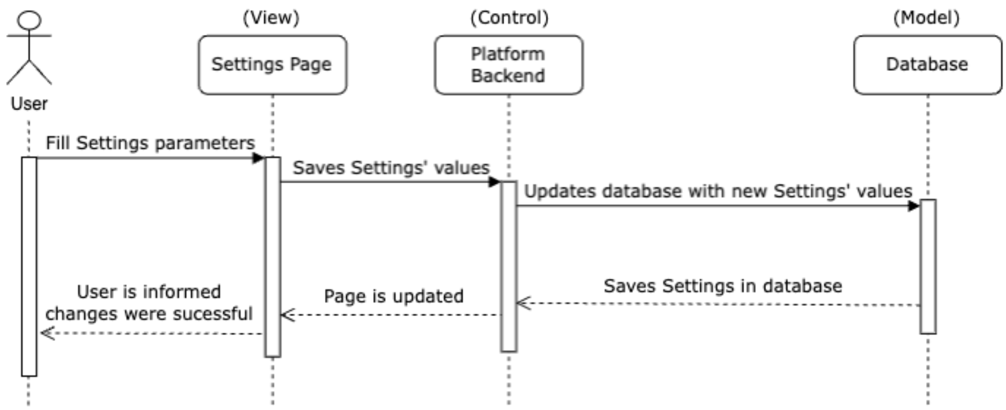


Figure 4.4-4 – UML sequence diagram of the Settings update process

4.4.4 Appliances

For this dissertation work purpose the appliances are only simulated and there are no real appliances connected to the platform, however, this is something that is intended to be done in future developments of this project.

To add an appliance to the system, the following parameters must meet the designated conditions, representend in Table 7.

To ensure that when the schedules are executed automatically, the system knows which appliance to schedule first, the parameter *Priority* was created. This value ranges from 1 to 10, being 10 the highest of priorities and 1 the lowest of priorities. So, an appliance is designated with a high value priority, if the system's mode is *Automatic*, the system itself will start by choosing to schedule that same appliance first, and only after this appliance is scheduled, the system will proceed to the next higher priority appliance. If two or more appliances have the same priority values, the system will schedule them in the order that they were added to the system, starting with the appliance that was created and added first.

Table 7 - Appliance's parameters definition and conditions

Parameter	**Definition and Condition**
Appliance's Name	Must be unique for each user.
Average Power per Use (W)	Power consumed by the appliance per use. Must be greater than 0.
Minimum Power Produced to Schedule (W)	Minimum power being produced by the photovoltaic panel to schedule and appliance for the future. Must be greater than 0.
Average Time of Each Use (h)	Time consumed by the appliance per use. Must be greater than 0.
Energy Consumption (kWh/day)	Amount of energy consumed over a full day. Must be greater than 0.
Priority	Priority level of each appliance, being 10 the highest level. Must be between 1 and 10.
Number of Ideal Times to Schedule per Week	Number of desired times to automatically schedule an appliance in a week. Must be greater or equal than 0.

OptiSun will automatically schedule the appliances, therefore, to ensure that it knows how many times to schedule an appliance, it resorts to the parameter *Number of Ideal Times to Schedule per Week*. With that being said, *OptiSun* will automatically schedule the appliances every Sunday for the following week. Each appliance will be scheduled according to their priority, and how many times the user defined it on the *Number of Ideal Times to Schedule per Week* parameter. Figure 4.4-5 represents the successful creation of a new appliance in *OptiSun*.

Besides being able to connect and create an appliance, it is also possible to schedule an appliance and delete an appliance. The *Delete Appliance* function deletes the appliance from the system. However, it doesn't delete it from the database, but instead, updates the *IsActive* parameter in the **Appliance** database table, to "False".

On the other hand, the *Schedule Appliance* function will "turn on" the appliance scheduling it to the exact time the button was pressed by creating a new row on the **ApplianceSchedule** database table with the appliance's Id. The consumed power is also calculated and added to the new row, just as the *Manual* parameter is defined as "True" and the *Automatic* is set to "False". When scheduling the appliance a prevision of the consumption values is made and automatically added to the load balances, to make sure that no value is missed. By choosing to schedule an appliance automatically the conditions of the minimum power produced to schedule looses its power since this is something manually.

In the future it would be interesting to add a warning to inform the users in case the produced values are indeed lower than the minimum power to schedule but as for now, those warnings are not implemented yet. Figure 4.4-6 represents the process of manually scheduling an appliance via the *Appliance's* view page.

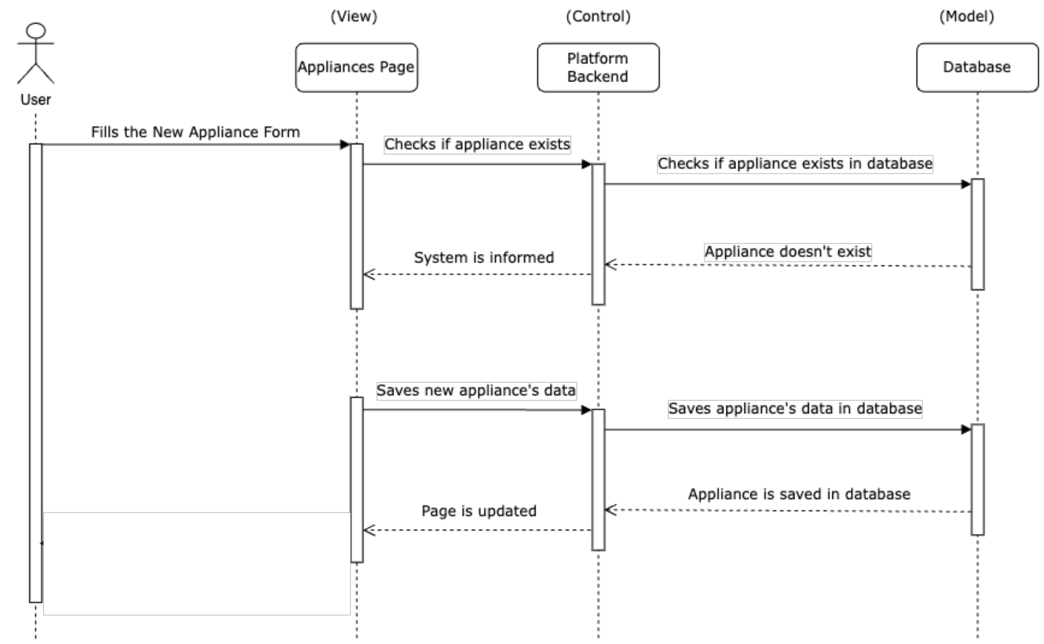


Figure 4.4-5 – UML sequence diagram of the process of creating a new appliance

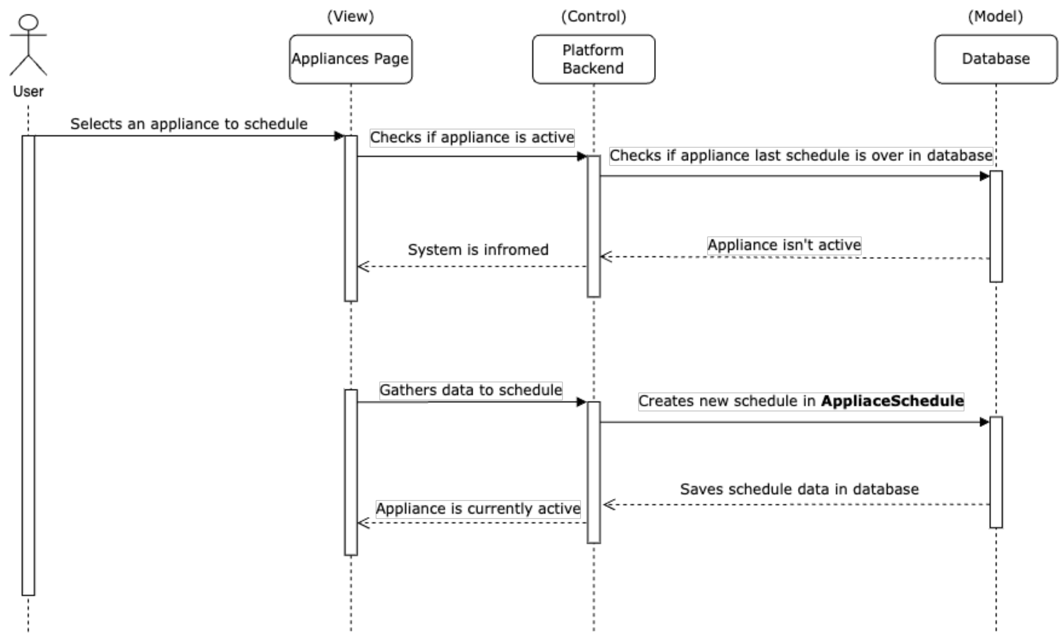


Figure 4.4-6 – UML sequence diagram of the process of manually scheduling an appliance

4.4.5 Loads Balance

One of the most useful functionalities of the developed platform is the Loads Balance. This functionality allows the user to not only view the real time power produced by the panel but also the balance of loads during the day, week, month and even year.

The Real Time Produced Power value is obtained by accessing the IAMMETER sensor API by using the token and serial number values that were added to the user settings, as explained in the previous chapter. By having the token and serial number of the sensor, the API can now fetch the real values and then display them in the Loads Balance page, as illustrated in Figure 4.4-7. The Real Time Produced Power value is an interesting one to have displayed, allowing the user to always have an idea of what is being produced in that exact second, allowing them to manually schedule an appliance in that instance if desired.

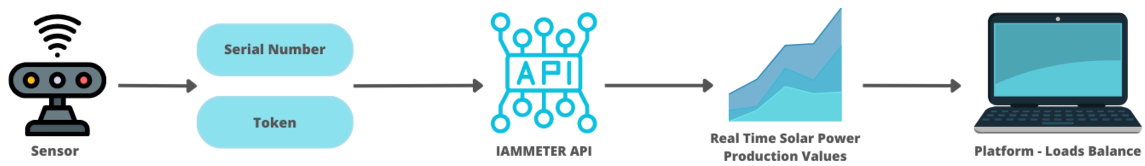


Figure 4.4-7 - Collection of real time solar power produced values process

Four different Loads Balance pages were developed each corresponding to the *Daily Balance*, *Weekly Balance*, *Monthly Balance*, and *Annual Balance* buttons. In each of these views pages were designated several parameters were developed to give more information on the balance of loads for each time interval, as illustrated in Table 8:

Table 8 - Loads Balance parameters and definitions

Parameter	**Definition**
Total Produced Energy	Total power produced in that period.
Consumed Power	Total amount of power consumed by the appliances in the system in that period.
Balanced Value	Difference value between the *Total Produced Energy* and the *Consumed Power.*
Money Saved	Multiplication of the *Consumed Power* by the *Purchase Tariff.*
Money Made in Sales (Prevision)	Multiplication of the *Balanced Value*, by the *Sales Tariff.*

Each of these values are represented for the current day, week, month, and year the user is currently in. This means that if the user check each of these balances' pages on the 2nd of September of 2023, they will be able to see the daily balance of that day, the weekly balance of the week starting on the 28th of August and ending on 3rd of September, the monthly balance of the month of September and the annual balance of the year 2023.

The total power produced in a period of time is calculated by accessing the **PanelReading** database table and summing all the *Active Power* values between that interval of time and converts it from W to kWh. To calculate the total consumed power in a time period, the system accesses the values of the *ConsumedPower* parameter of the **ApplianceSchedule** database table between the desired time period and adds them all together since they were previously calculated in kWh .

In Figure 4.4-8 is presented an example of how the Loads Balance functionalities processes work and how the data is gathered and calculated.

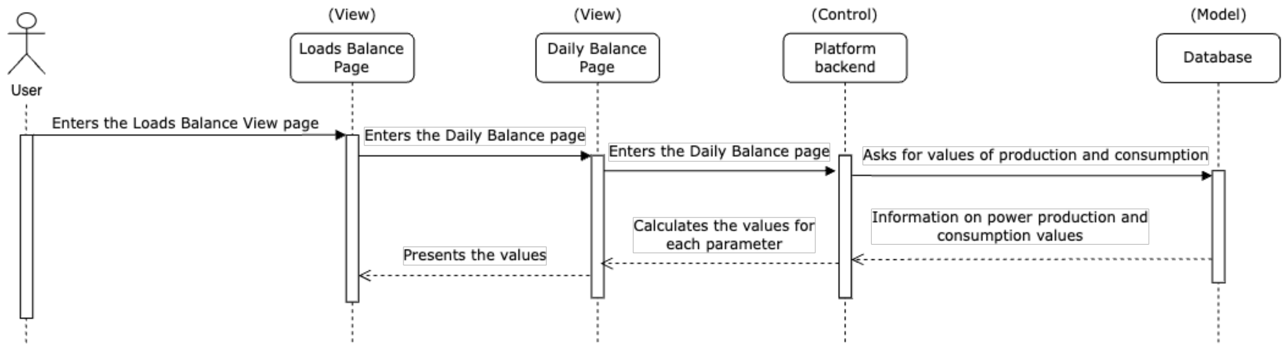


Figure 4.4-8 - UML sequence diagram of the Daily Balance functionality process

4.4.6 Calendar

The Calendar functionality, allows the scheduling of appliances up to 6 days in the future. The calendar has an hourly schedule in which the starting hour and finish hour are defined previously on the Settings component. In each time and date slot it will be possible to select appliances and schedule them, however, only if the time has not passed yet. This means that if it is 13h and the starting hour is 9h, the scheduling will not be possible to schedule for 9h, 10h, 11h and 12h of the current day.

The number of appliances to be scheduled during the desired are and time slots has to be equal or lower to the defined parameter in the Settings component, *Maximum number of Appliances to Schedule at Once*. However, in this case, the *Automatic* parameter in the **ApplianceSchedule** data base is set to "True" and the *Manual* parameter is set to "False", unlike explained in the sub-section above.

Figure 4.4-9 represents the successful manual schedule of one of more appliances via the Calendar component.

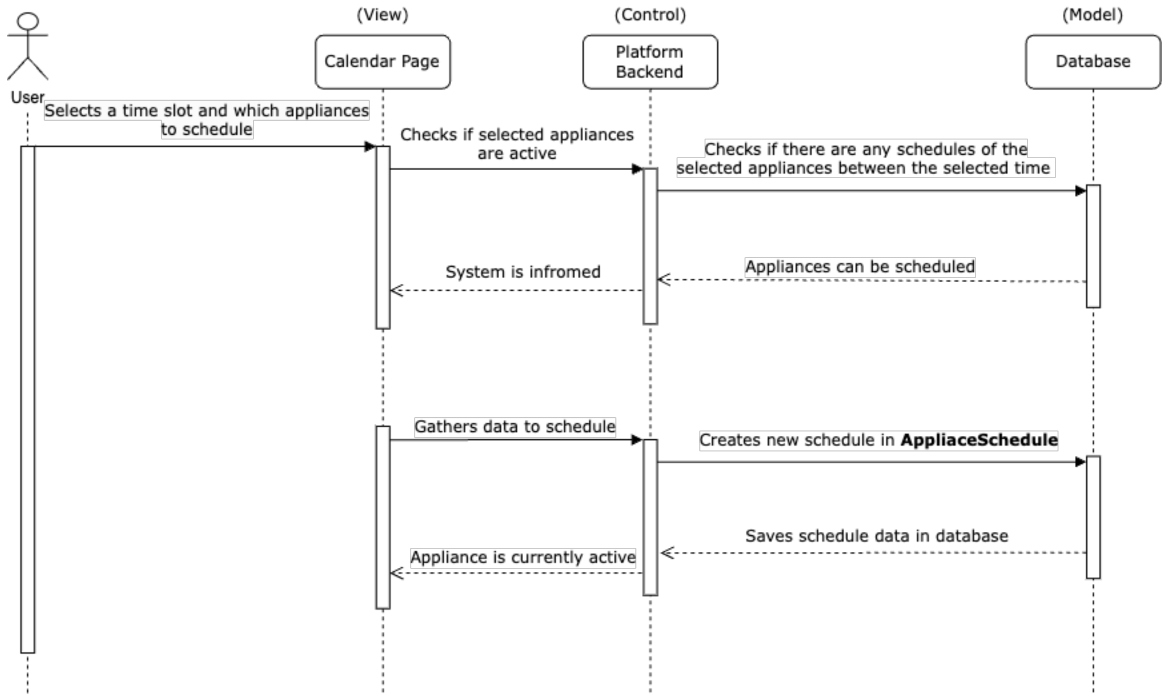


Figure 4.4-9 - UML sequence diagram of the manual schedule of appliances in the Calendar page

4.5 Machine Learning Model

To be able to predict the power production values of the photovoltaic panel in the future, it was necessary to implement a machine learning model that, based on the historical values, would calculate and predict cautiously the future values. As it was mentioned in chapter 2.4 - Machine Learning Predictive Model, to develop a predictive model, there are several steps to be taken as represented in Figure 2.4-1. All these steps were executed when developing the predictive model, in order to achieve the best possible results.

4.5.1 Data Analysis

To achieve the most accurate values possible when predicting, the data analysis of the existing previous values is important, since the developed model will learn from the existing data.

The first step to develop a Machine Learning Model is to collect all the necessary data, clean the data frames and select which values and parameters are the most needed.

In the beginning of the Implementation chapter, more precisely in subchapter 0- **Database Modeling**, it was referenced that three data base tables were created to save the values and data fetched from three different APIs.

The API from the IAMMETER Sensor retrieves values every minute, such as voltage, current, active power, power consumption, feed in energy, local and GMT time. However, in the analysis of data to train the model, from this data base table, it was only necessary to keep the Local Time, which its name was changed to *DateTime*, and the Active Power since the other values such as current and voltage were not relevant to predict the power. The cleaned data frame can be observed in the figure below.

	PanelReadingID	UserId	ActivePower	DateTime
0	1	0	19	2023-05-01 19:35:30
1	2	0	15	2023-05-01 19:39:28
2	3	0	-3	2023-05-01 20:30:11
3	4	0	-3	2023-05-01 20:32:10
4	5	0	-3	2023-05-01 20:33:10
...	...	...	...	...
10792	10793	0	5	2023-07-04 20:34:09
10793	10794	0	5	2023-07-04 20:35:09
10794	10795	0	4	2023-07-04 20:36:08
10795	10796	0	4	2023-07-04 20:37:08
10796	10797	0	3	2023-07-04 20:38:08

10797 rows × 4 columns

Figure 4.5-1 - Cleaned data frame of the **PanelReading** data base table

As can be seen in Figure 4.5-1 the parameters *PanelReadingID* and *UserId* were kept just to demonstrate the ids.

When looking into Figure 4.2-4 in the 0 – **Database Modeling** chapter, the API that fetched the real time weather condition values every minute, saved values such as longitude and latitude that were not needed for the training of the model. Therefore, the data frame of the real time weather values was modified and cleaned to have the Temperature, Humidity and Local Time values, accompanied with *RealTimeId* and *UserId* for the same purposes of the previous dataframe, as can be seen in Figure 4.5-2.

	RealTimeId	UserId	Temperature	Humidity	DateTime
0	1	0	26.53	47	2023-05-02 18:51:00
1	2	0	26.53	47	2023-05-02 18:53:00
2	3	0	21.58	57	2023-05-04 12:07:00
3	4	0	23.69	53	2023-05-28 15:24:00
4	5	0	23.77	53	2023-05-28 15:42:00
...	...	...	...	...	...
15980	15981	0	25.00	53	2023-07-10 09:34:00
15981	15982	0	25.00	53	2023-07-10 09:35:00
15982	15983	0	25.00	53	2023-07-10 09:36:00
15983	15984	0	25.00	53	2023-07-10 09:37:00
15984	15985	0	25.00	53	2023-07-10 09:38:00

15985 rows × 5 columns

Figure 4.5-2 - Cleaned data frame of the **RealTimeWeather** data base table

Lastly, the API that fetched the future values for weather conditions was also altered to only keep the Temperature, Humidity, Direct Radiation, Diffuse Radiation, Irradiance and Date Time as can be seen in Figure 4.5-3, since these were the most relevant values to predict the power production values. The main difference between this data frame and the previous ones is that this one only has values every 1 hour, when both the previous ones had values every 1 minute. This created the need to interpolate new values to make sure that the time interval of this new data frame was also every 1 minute. The interpolation values will be explained later in this chapter.

	DateTime	Temperature	Humidity	Direct_Radiation	Diffuse_Radiation	Irradiance
0	2023-04-02 00:00:00	15.4	87	0.0	0.0	0.0
1	2023-04-02 01:00:00	14.9	83	0.0	0.0	0.0
2	2023-04-02 02:00:00	14.4	84	0.0	0.0	0.0
3	2023-04-02 03:00:00	14.4	87	0.0	0.0	0.0
4	2023-04-02 04:00:00	14.1	82	0.0	0.0	0.0
...	...	...	...	...	...	...
2587	2023-07-18 19:00:00	22.2	49	296.0	46.9	828.0
2588	2023-07-18 20:00:00	20.7	58	122.3	26.1	731.5
2589	2023-07-18 21:00:00	19.0	69	0.0	0.0	0.0
2590	2023-07-18 22:00:00	17.8	77	0.0	0.0	0.0
2591	2023-07-18 23:00:00	17.4	81	0.0	0.0	0.0

2592 rows × 6 columns

Figure 4.5-3 - Cleaned data frame of the **ForeCastWeather** data base table

After cleaning the data frames, the next step to take is to start merging all the data frames to create one single data frame, to prepare the data as was mentioned in chapter 2.4 - Machine Learning, once the data collection process was already complete. To make the process easier, it was decided to firstly merge the first two data frames together, and the merge the result of the merging of the first two data frames with the last data frame.

During the process of merging the Panel Reading data frame and the Real Time Weather data frame, we omitted all the id parameters and trimmed a few rows from each table. This data reduction was necessary to streamline the training process and minimize errors. As a result of the merging of both data frames, a new data frame was created and can be analyzed in Figure 4.5-4, presented below.

	Temperature	Humidity	DateTime	ActivePower
0	25.53	52.0	2023-06-29 18:29:00	200.0
1	25.53	52.0	2023-06-29 18:30:00	197.0
2	25.50	57.0	2023-06-29 18:31:00	193.0
3	25.50	57.0	2023-06-29 18:31:00	193.0
4	25.50	57.0	2023-06-29 18:32:00	189.0
...	...	...	...	...
7258	24.13	53.0	2023-07-04 20:34:00	5.0
7259	24.13	53.0	2023-07-04 20:35:00	5.0
7260	24.13	53.0	2023-07-04 20:36:00	4.0
7261	24.13	53.0	2023-07-04 20:37:00	4.0
7262	24.13	53.0	2023-07-04 20:38:00	3.0

7263 rows × 4 columns

Figure 4.5-4 - Data frame created by merging the *Panel Reading* data frame and the *Real Time Weather* data frame

Finally, all was left to do was merging the new data frame with the *Forecast Weather* data frame to obtain the final data frame, that would make training the model possible.

To prepare the *Forecast Weather* data frame to be merged, an interpolation of new values had to be made to make sure that the time interval between each row would be 1 minute. By creating interpolation values for the columns *Direct Radiation*, *Diffuse Radiation*, and *Irradiance*, it was now possible to merge both data frames together to obtain the data frame that will train the model. The final data frame obtained can be observed in Figure 4.5-5, presented below. This data frame has way less rows, to prevent issues of too much data to be analyzed by the model, that led to bigger errors when predicting. The reduction of rows was done with caution, to ensure important information was not lost and generalization was mantained. This likely happened because the radiation and irradiance values were interpolated and by testing the model with too many rows may have led to overfitting. The possible existence of a larger number of outlier values in a larger datset might have also led to increase in the error values. Therefore, it was decided to cut down on several dataset rows since the error values when testing the model were high. This data frame is composed of the following parameters:

- *DateTime;*
- *Direct_Radiation;*
- *Diffuse_Radiation;*
- *Irradiance;*
- *Temperature;*
- *Humidity;*
- *Active Power.*

It must be taken into consideration that the only values that were interpolated were the radiation and irradiance values since the temperature, humidity and active power values were crucial to be the exact real ones, and not interpolated ones, to ensure the best prediction of the active power values.

	DateTime	Direct_Radiation	Diffuse_Radiation	Irradiance	Temperature	Humidity	ActivePower
0	2023-06-29 18:29:00	324.95	126.69	688.95	25.53	52.0	200.0
1	2023-06-29 18:30:00	321.92	126.30	686.72	25.53	52.0	197.0
2	2023-06-29 18:31:00	318.88	125.90	684.48	25.50	57.0	193.0
3	2023-06-29 18:31:00	315.85	125.51	682.25	25.50	57.0	193.0
4	2023-06-29 18:32:00	312.81	125.12	680.01	25.50	57.0	189.0
...	...	...	...	...	...	...	...
7258	2023-07-04 20:34:00	32.28	30.68	178.66	24.13	53.0	5.0
7259	2023-07-04 20:35:00	31.04	29.50	171.79	24.13	53.0	5.0
7260	2023-07-04 20:36:00	29.80	28.32	164.92	24.13	53.0	4.0
7261	2023-07-04 20:37:00	28.56	27.14	158.05	24.13	53.0	4.0
7262	2023-07-04 20:38:00	27.32	25.96	151.18	24.13	53.0	3.0

7263 rows × 7 columns

Figure 4.5-5 - Data frame created to develop and train the machine learning model

Now that the data analysis is complete, it is time to start training the model to predict the desired value of the active power production.

4.5.2 Model Development

To start training the model it is needed to firstly split variables for training and testing. After splitting these variables, it is now possible to start training model.

To ensure the most accurate prediction would be made, it was decided to train four different models, using four different regressors:

- Extra Trees Regressor;
- Gradient Boosting Regressor;
- Random Forest Regressor;
- XGB Regressor.

The Extra Trees Regressor, or Extremely Randomized Trees Regressor creates many decision trees and the sampling for each tree is random. This allows the creation of datasets for each tree, each with unique samples. The number of features is selected randomly for every tree. The Extra Trees Regressor is unique since the selection for the splitting values for features is completely random, therefore, the algorithm selects a split value randomly, making the trees diversified and unrelated (ArcGIS Pro 3.0, n.d.).

Gradient Boosting Regressor sequentially builds a strong predictive model by combining multiple weak models, typically decision trees. It starts with an initial prediction and then fits subsequent models to the errors of the previous ones, minimizing the residuals through a gradient descent optimization. The final prediction is the sum of all the individual model predictions. This method improves predictive accuracy and captures complex relationships in the data (H. Friedman, 2001).

The Random Forest Regressor constructs a forest of decision trees, where each tree is built using a random subset of the data and features. During training, the algorithm creates multiple decision trees by employing a technique called bagging (bootstrap aggregation). Each tree is trained on a different subset of the data, and they collectively make predictions by averaging or voting, depending on the task (Breiman, 2001).

XGBoost Regressor combines the strengths of multiple weak learners, often decision trees, to create a powerful predictive model. It employs a clever combination of regularization techniques to prevent overfitting and enhance model performance. XGBoost iteratively adds decision trees, focusing on reducing the errors of previous predictions through gradient descent optimization. The final prediction is the sum of predictions from all trees, leading to an accurate and robust regression model (Chen & Guestrin, 2016).

After training each model it is necessary to evaluate them, to understand which model is better to predict. The higher the values of *Explained Variance*, the better the model is to predict. *Explained Variance*, is a metric that measures how well a regression model accounts for variance in the dependent variable using independent variables. It's a value between 0 and 1, where higher values indicate a better model fit, as it explains more variance (Kharwal, 2021). To train the models, the parameters given for training were the *Direct Radiation*, *Diffuse Radiation*, *Irradiance*, *Temperature* and *Humidity*. On the other hand, the parameter for testing was the *Active Power*. By choosing to train and test with these values, the model will always predict the *Active Power*, which is the desired value to predict since the others can always be obtained using the Forecast Weather API.

When analyzing the evaluation of the models, it is possible to see the values of *Explained Variance* of each developed model. The values are presented below in Figure 4.5-6.

	Extra Trees Metrics	Gradient Boosting Metrics	Random Forest Metrics	XGBoost Metrics
Explained Variance	0.999844	0.999698	0.999765	0.999748
Mean Absolute Error	2.106201	3.664796	2.872312	2.786962
Mean Squared Error	23.661702	45.757659	35.609704	38.242184
Root Mean Squared Error	4.864330	6.764441	5.967387	6.184027
R-squared	0.999844	0.999698	0.999765	0.999748

Figure 4.5-6 - Evaluation of models

By looking at the evaluation of each model, it is possible to see that all the models have similar *Explained Variance* values, however, the *Extra Trees Regressor* presents slightly higher values with an *Explained Variance* of 0,999844. For this reason, the chosen regressor to implement the model was the *Extra Trees Regressor*.

After training all models and analyzing which values were the most adequate, it is needed to create a model dump to save the data of the trained model. The following step is to create an API to fetch the predicted values by the developed machine learning model. The API was created using the FastAPI framework. After developing the API, the only remaining task is to post to the API the training parameters, *Direct Radiation*, *Diffuse Radiation*, *Irradiance*, *Temperature* and *Humidity* and wait for a response with the value of *Active Power*, which corresponds to the prediction of the power production by the photovoltaic panel.

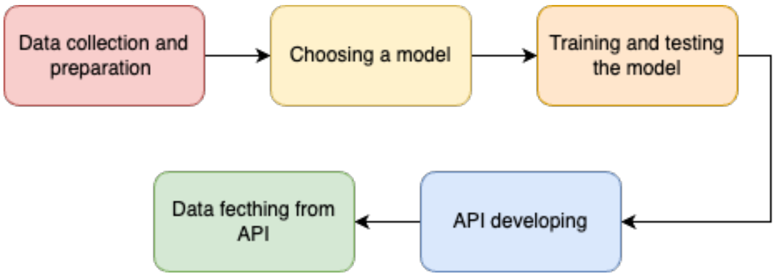


Figure 4.5-7 - Predictive Model developing steps

The model is now complete, and it is now possible to predict the production values and allow the system to automatically schedule the appliances since the power production in now somewhat known.

4.6 Platform Automation

With the development of a machine learning model that forecasts photovoltaic panel power production, the system is ready for implementation to automatically schedule appliances. Having knowledge of the expected power output for a particular date and time enables this automation. This addition will significantly enhance the system, providing comprehensive assistance to the user and further contributing to energy-saving maximization.

4.6.1 Calendar

When accessing the Calendar page, the user can select a specific date and time slot, and subsequently select the appliance they desire to schedule.

To help the user decide when to schedule each appliance, now that the production power values are predicted for the future, in each time slot there can be seen the power production predicted values. To retrieve the predicted values, the system fetches the data from the *ForeCast Weather* data base table with the future values for temperature, humidity, direct and diffuse radiation, and irradiance, and creates a list by date and time. Then, with this list, the values are predicted using the created API which posts the forecast data and retrieves the power production predictions. After creating this list, the created API is called and the values inside the list will be posted in this API, which later responds with the desired value, the prediction of power production. Figure 4.6-1represents the process of automating the Calendar’s functionality.

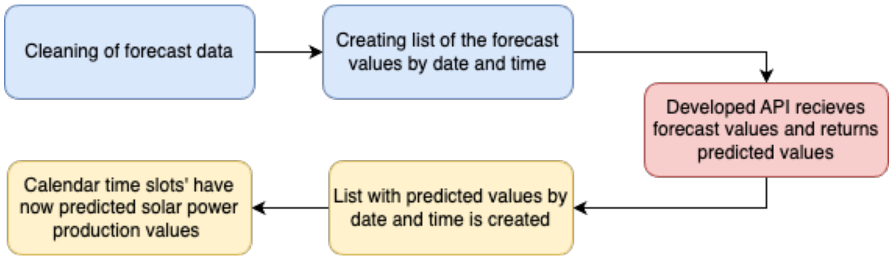


Figure 4.6-1 - Calendar's Automation Process

Initially it was desired to present these predicted values in each time slot inside the respective modal once the button to schedule was pressed. It was also desired to subtract the power consumed by each appliance as each appliance was selected, but due to the lack of time and the priority of other implementations, this was left to be developed in later works.

4.6.2 Home Page

When entering the Home Page, if the system is in automatic mode, every Sunday, the system will automatically execute a function that will schedule all the appliances throughout the week. Each appliance will be scheduled the desired number of times per week, previously set in the settings page.

The functioning process of the automatic schedule is a bit extensive, therefore a scheme was created to better illustrate the scheduling behavior. The scheme is represented in the figure below, in Figure 4.6-2, to aid the reader to better comprehend the many steps of this process.

The first step is to clean the data from the *Forecast Weather* database table, similarly to how it is explained above in chapter 4.6.1 - Calendar, where all the data from that table up to 7 days in the future is retrieved and later ordered in descending order of values. Then, a list is created with the date, time, irradiance, direct and diffuse radiation, humidity, and temperature. After this list is created, in the API that will fetch the power production prediction values, the system will post the values of the forecast data list, excluding date and time.

After the API fetched the power production prediction values, two new lists are created: list number one and list number two. Both lists contain the values of date and time and the predicted value retrieved from the API and are used to better organize and gather the data needed for the schedules. List number one is constituted by seven values, where each value represents the highest value and its respective date and time, for each day, in descending order of power production predicted values. On the other hand, list number two is composed by all the values for each date and time slot, with date, time and power production values, ordered in descending order as well, by the power production predicted values. In both lists, each date and time is the same as each time slot in the *Calendar*. This happens since the *Forecast Weather* data base table only retrieves values hourly from its respective API, since it was the minimum period of availability.

To schedule the appliances, the system will check the priority of each appliance and to do so, a new list is created with the appliance's id, its priority and number of ideal times to schedule per week which will be order by the highest priority to the lowest.

To start the scheduling process, the system will choose the appliance with highest priority, and will start scheduling it in the day and time with highest production value in list number one.

After selecting the date and time with highest power production values, the system will create a schedule and will update the lists with the production values, one and two. To update the lists, the system will start by deleting list number one and updating list number two by subtracting the power consumption value of the scheduled appliance on the selected date and time, from the power production predicted values. By subtracting the consumption from the power production, the new predicted value will be updated because there is already an appliance scheduled for those hours. Therefore, the system will not only subtract the consumption values from the time the appliance is scheduled to start working, but also from the following hours the appliance is predicted to function for. This means that if an appliance is scheduled to start working at 11 hours and is predicted to be working for two hours, the system will be subtracting its consumption power to the predicted produced power at 11 and 12 hours.

Now that list number two has all the new power production values, the list will once again be reordered in descending order of highest predicted production values. Finally, to update list number one, this list will be re-created from list number two, to present the highest values of each day also in descending order of predicted power production.

To proceed with scheduling each appliance, this algorithm will be iterated as many times as the user specifies for a given appliance's weekly schedule. Once this step is complete, the process will be replicated for every appliance linked to the user's account.

Since using a Knowledge-Based System in this project was not the main goal or priority, due to the lack of time, this algorithm was developed using a set of conditions as explained above in this sub-section. The desired method would be to have developed a Rule Based System as mentioned in chapter 2.5.1 - Rule-Based Systems.

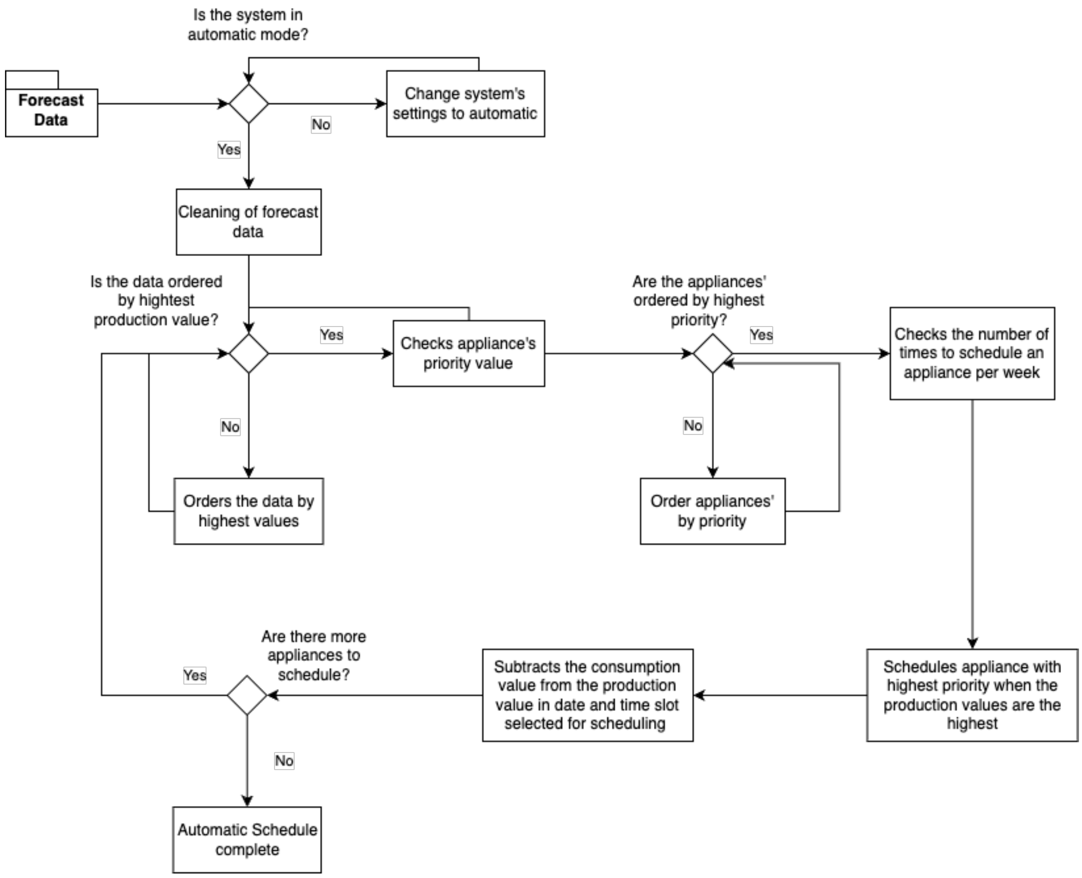


Figure 4.6-2 – UML diagram of the automatic schedule algorithm

5

EVALUATION AND DISCUSSION OF RESULTS

In this section the results obtained in the predictive model are evaluated base on how effective the predictions are. The accuracy of the automated *OptiSun* system and its worth on enhancing the user experience is also discussed. This chapter is all about verifying the real-world impact of this research in making homes smarter and more sustainable by using the implemented Web Application Platform. The *OptiSun*'s interface was not tested when it comes to the user's utilizations due to time limitations. Therefore, it is not possible to fully evaluate how users interact with it.

5.1 Web Application Platform Interface

When entering the platform, the user is faced with the Home Page as can be observed in Figure 5.1-1. After entering the *Home Page,* the user encounters a few navigation tabs, such as *Home*, *About Us*, *Contact Us* and *Sign Up*.

To create a new account, the new user just needs to press the *Sign Up* tab that takes them to a new view page, where the user can fill out a form with the information required to create the account as can be analyzed in Figure 5.1-1.

To successfully create an account, all the parameters of the form are required to be filled and both the username and email must be unique, meaning that if a user tries to create an account with an already taken username, an error message appears, and a new username must be chosen. Besides the restrictions with picking both a unique username and email, when deciding on the password, it must include minimum of 8 characters, composed of a combination of letters, numbers, and symbols, to ensure the maximization of security for the user.

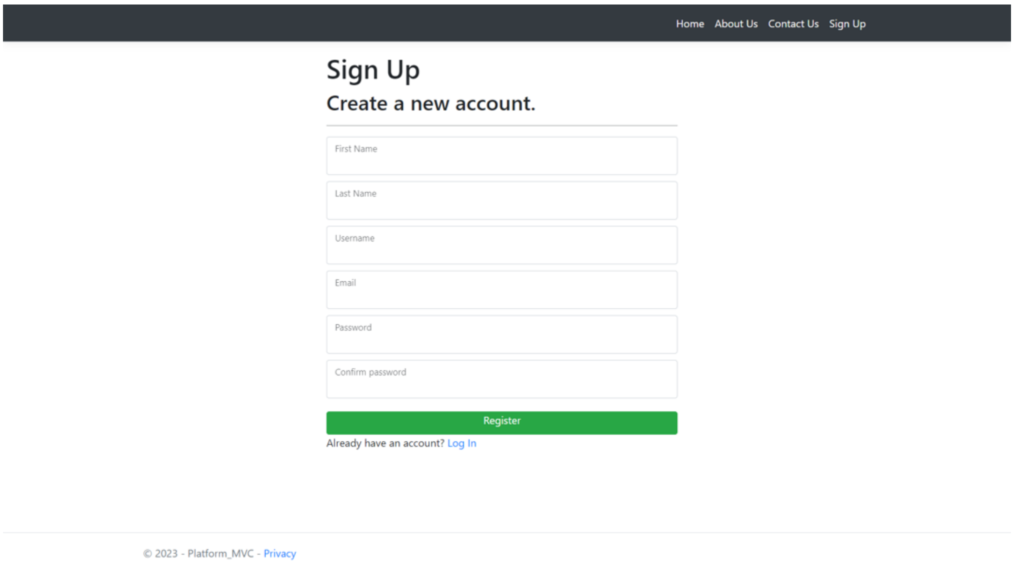


Figure 5.1-1 - *Sign Up* view page and form

Once the user completes the form and everything is submitted with no error messages, the user is now ready to log in and start using their platform.

In Figure 5.1-2, it is possible to observe a simple flowchart illustrating the registration process.

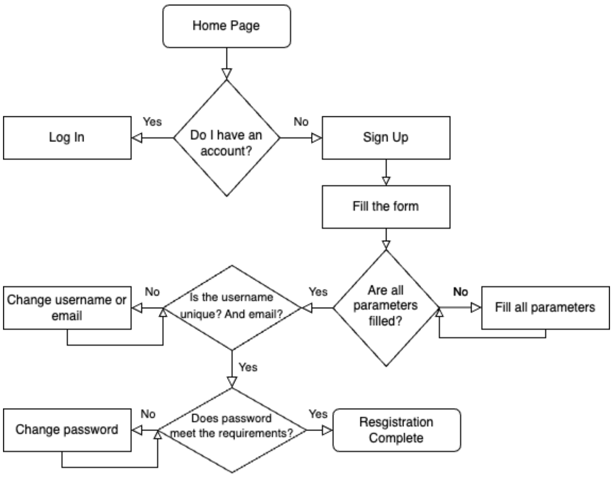


Figure 5.1-2 - Flowchart of the Registration process

To log in the user must click the *Log In* button on the *Home Page* which will redirect to a new view with the log in form, presented in Figure 5.1-3.

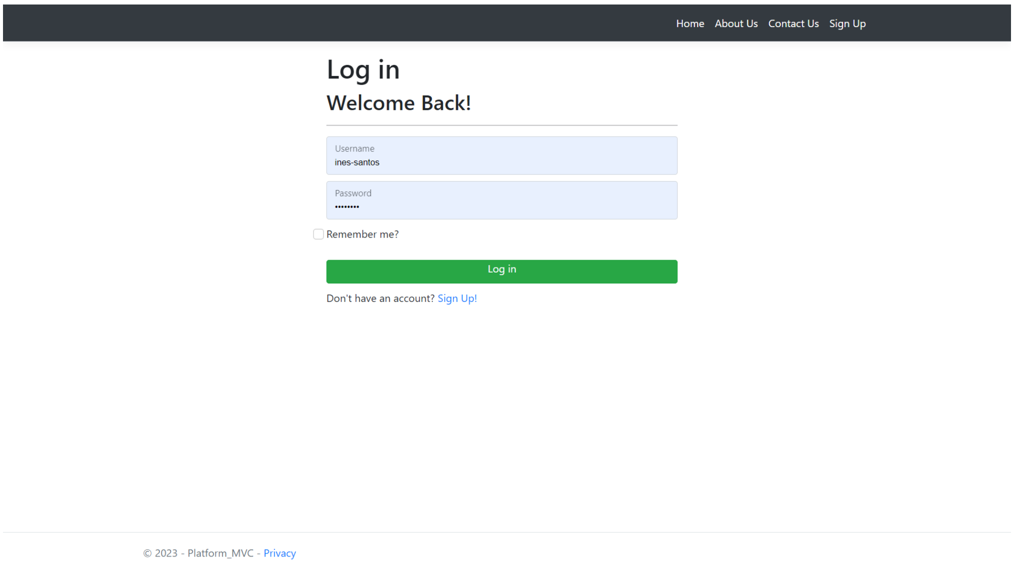


Figure 5.1-3 - *Log In* view page

The log in process is represented and explained by the flowchart represented in Figure 5.1-4.

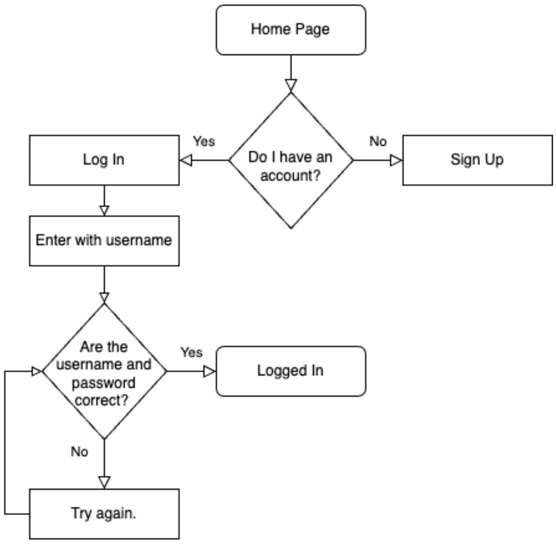


Figure 5.1-4 – Flowchart of the Log In process

When considering a scenario where an account is already created, once logged in, the user is now able to access all functionalities implemented in the system.

To make changes to the account, the user can access the user settings by clicking the *Hello username!* or the *Your Account* tabs, redirecting them to another page that can be observed in the figure below.

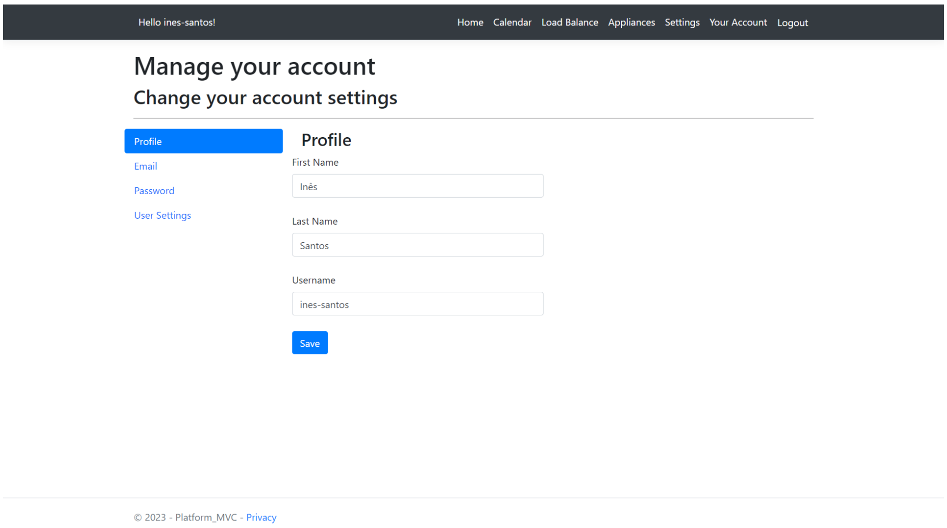


Figure 5.1-5 - *User Settings* profile page

To alter the system settings the user must select the *Settings* tab, which will take them to the *Settings* page and will allow them to make the desired changes. The *Settings* page is presented in Figure 5.1-7 presented below.

When pressing the first tab, the *Profile* tab, the user can change both First and Last Names and even the Username. However, the previous condition of the Username being unique is maintained. With this said, if a user tries to alter the Username to an already taken Username, an error will appear, and the user will have to choose a different Username. This maintains the integrity of the project, even though a user is also represented by an Id. The changes are only saved when the *Save* button is pressed.

To change the email, the user must go to the *Email* tab and write the new email on the parameter called *New email* and press the button named *Change email* to save the changes just made.

If, on the other hand the user chooses to change the password, they will have to enter the current password and then the new desired password and finally type the new password once again to make sure the new password matches correctly on both parameters. Finally, to save these changes, the user must press the button *Update Password* to the save the changes made.

Finally, the *User Settings* tab was created to make sure that the platform can access the IAMMETER API with the correct token and serial number of the photovoltaic panel sensor that the user owns. This allows to retrieve the real time production values from the panel. Besides the token and serial number, the user can also add the longitude and latitude of the photovoltaic panel so that the real time weather values can also be retrieved from the weather API. The weather API values are currently not being fetched since the code to execute it wasn´t implemented but the idea was to be able to access the temperature and irradiance values in real time when logging into the home page. However, this functionality was not as important and relevant and was decided to be implemented in the future. Once again, to save the changes made and update the parameters, the user must press the Save button.

In Figure 5.1-6, a scheme representing the Account Settings complements and facilitates the analysis of this functionality.

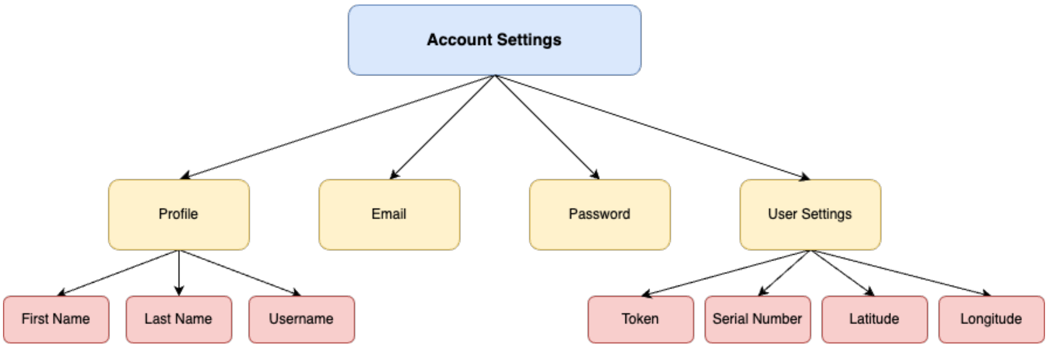


Figure 5.1-6 - Account settings schematic

The Settings page is dedicated to define the settings in the system. In this page the user will find two dropdown menus in which one refers to the *General* settings of the system and the other refers to the *Tariff* settings as can be seen in Figure 5.1-7 - *Settings* page.

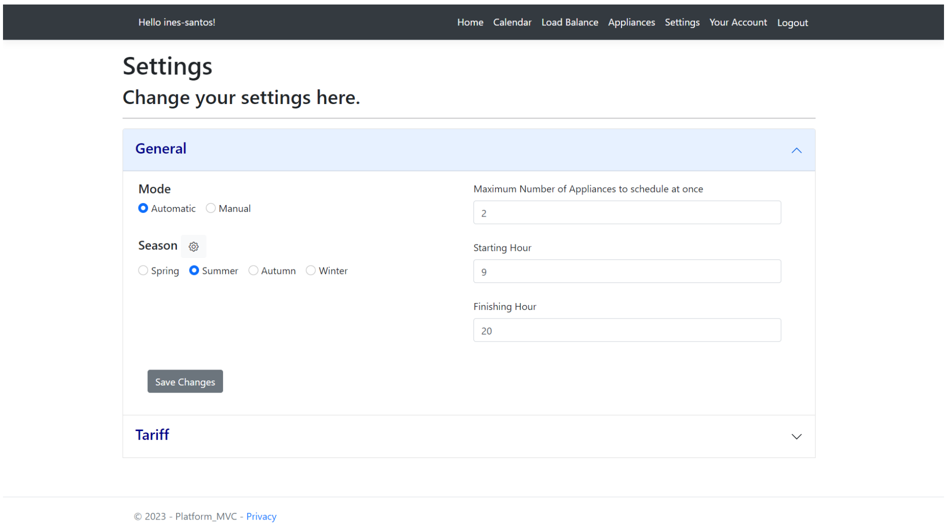


Figure 5.1-7 - *Settings* page

In Figure 5.1-7, it is possible to analyze better how the *General* settings dropdown menu works.

To add an appliance or alter an appliance's setting, the user must go to the appliance's page as explained in the previous chapter. In the scenario presented below, the user has three different appliances connected to the web application platform as can be seen in Figure 5.1-8.

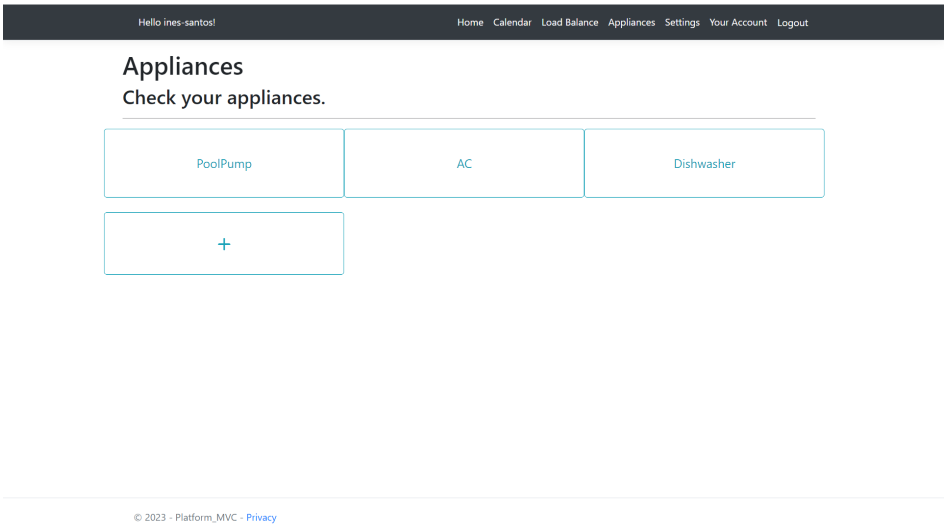


Figure 5.1-8 – *Appliances*' Page

If the user desires to schedule one or more of their appliances at once manually, they must go onto the *Calendar*'s Page and manually select a specific time slot, which will have information on the prediction power production value during the selected time slot and select which appliances are to be scheduled. In this scenario, the predicted values of the power production value can be observed below in the *Calendar*'s page figure.

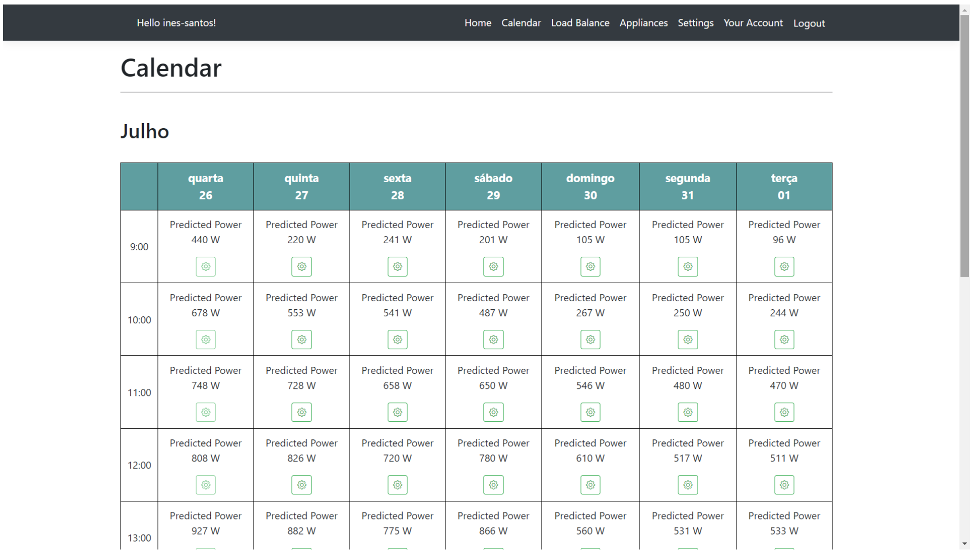


Figure 5.1-9 - *Calendar*'s Page

By pressing the button in each time and date slot the user will be able to select appliances to schedule on that day and time. However, they will only be allowed to schedule the appliances if the number of selected appliances is equal or lower to the defined parameter in the *Settings* page, *Maximum number of Appliances to Schedule at Once*. Once the user selects the maximum number defined of appliances, they will not be able to check anymore boxes since the checkboxes will turn to disabled. So, if the user wants to change the selected appliances, they must unselect one appliance and select another. After selecting the appliances, all is left to do is press the *Schedule* button. This will automatically schedule the appliances for the desired date and time.

The user can also cancel the schedule made previously by selecting the same date and time slot and pressing the *Cancel Schedule* button. However, unfortunately the user will not be able to see which appliances are scheduled through the calendar since the checkboxes will not be selected if scheduled. The appliances scheduled are not presented as checked in each time and date slot since there was some trouble implementing this functionality. Since this functionality was not crucial to the system, even if interesting and useful, its implementation was not prioritized.

In the Figure 5.1-10 is represented the flowchart of the scheduling process in the *Calendar* page.

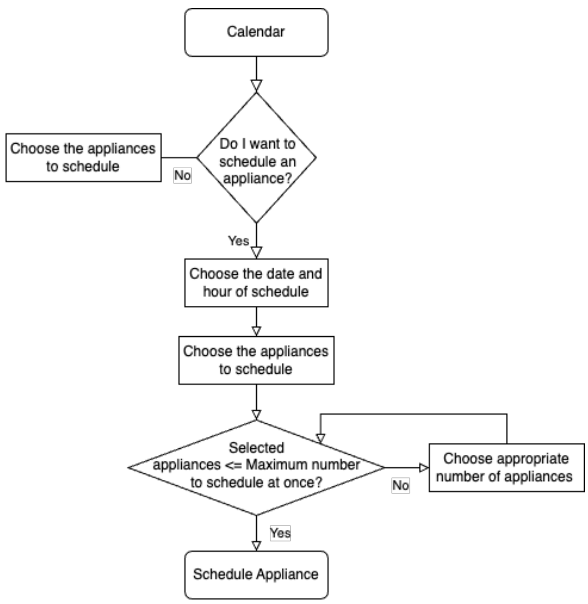


Figure 5.1-10 - Flowchart of the schedule process in the Calendar page

When the user selects a time slot, they will also be able to select how many appliances defined earlier in settings, as can be seen in Figure 5.1-11, presented below.

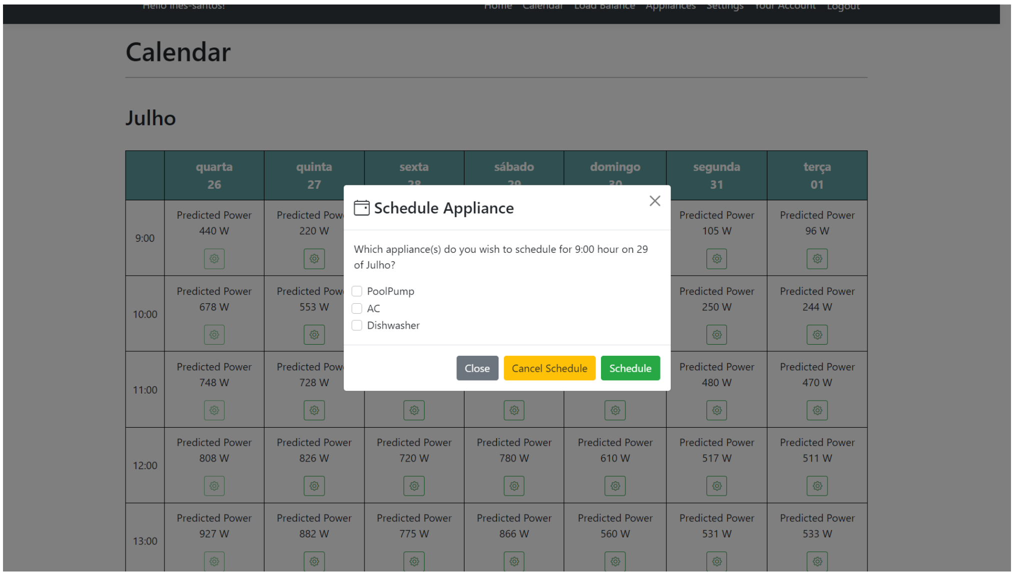


Figure 5.1-11 - Manual Schedule in *Calendar* Page

However, if the user decides to let the system automatically schedule the appliances for the week, or to do the automatic reschedule by pressing the *Reschedule Automatic Schedule for the Week* button, in Figure 5.1-12, in the logged in the *Home* page, the future scheduled appliances will be updated and represented considering this scenario as it is presented in the Figure 5.1-14, below.

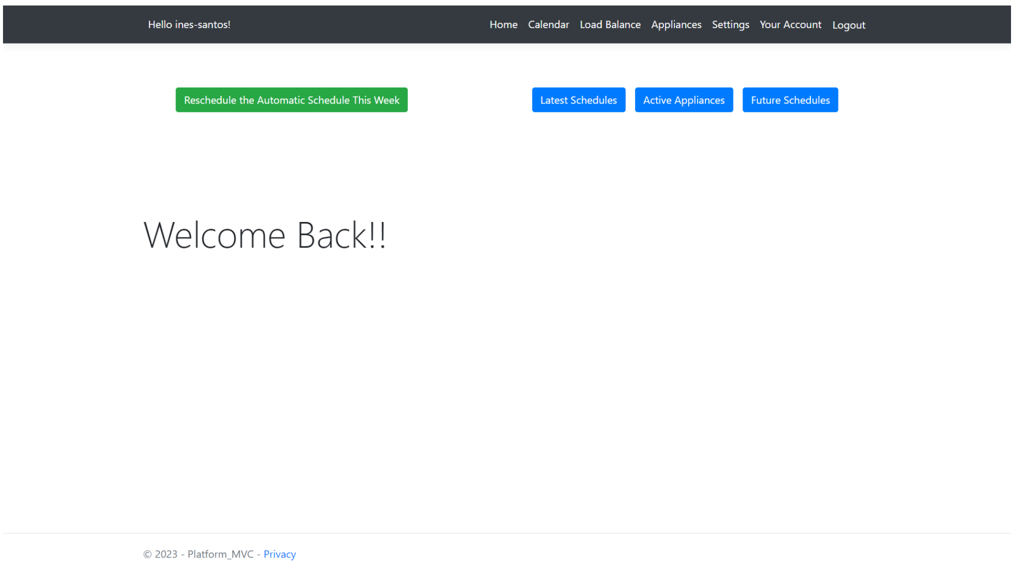


Figure 5.1-12 - Logged in *Home Page*

The first button the user will see is the *Reschedule the Automatic Schedule This Week* button which will re-do the automatic schedule programmed to execute every Sunday to schedule all the appliances according to their priority and number of times to schedule per week. When the user presses this button, all the schedules previously made will be deleted and a new schedule will be made. This allows the user to have a better schedule since the predicted values for power production will be more precise as the days get closer.

On the right side of the page, the user must see another three buttons being the first in the *Latest Schedules* button which will provide the user with a list of all the latest schedule of each appliance.

There is also an *Active Appliances* button that presents a list of all the appliances that are currently active along with their start date and time and end date and time. This list is presented in a checkbox format since you can easily check the desired appliance and *Cancel Schedule*. The *Cancel Schedule* button will turn off the appliance and calculate how much energy was consumed from the moment the scheduling started until the appliance was turned off.

Finally, the last button is a *Future Schedules* button that presents a list of all the appliances that are still to turn active in the future and a checkbox if the user desires to cancel a future scheduling.

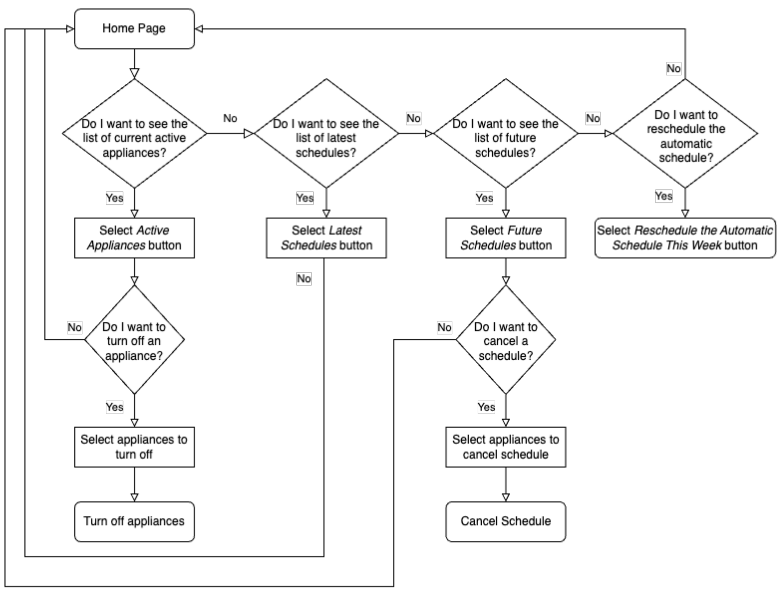


Figure 5.1-13 - Flowchart of Home Page functionalities

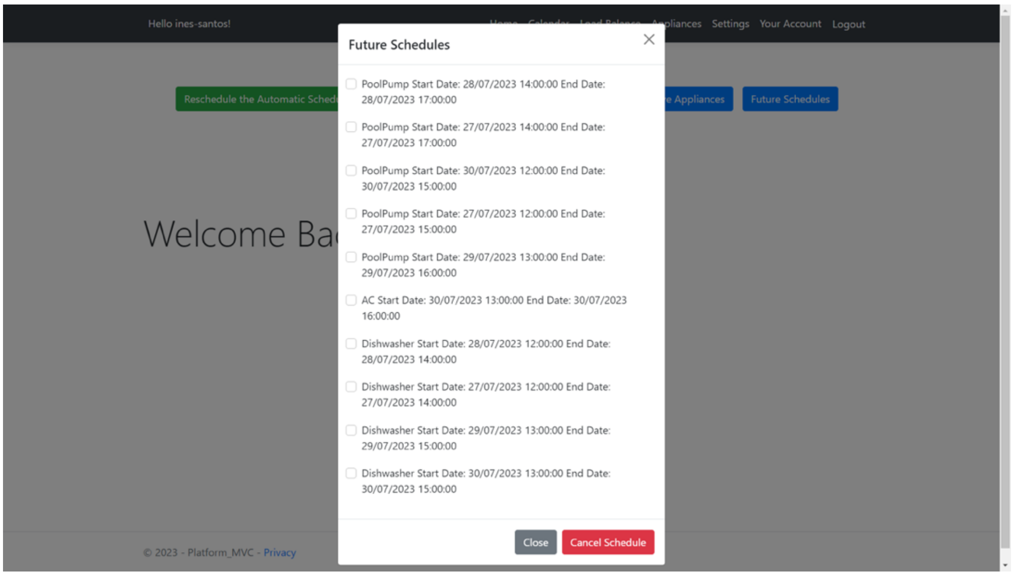


Figure 5.1-14 - Future Schedules of presented scenario

As can be seen in Figure 5.1-14, the system scheduled the appliances in order of highest priority and higher predicted production power values, which was the desired intention and goal of the whole platform, facilitating the user's job. When analyzing the future scheduled appliances, it can be said that the system is functioning well.

Finally, to visualize and access the energy production and consumption information the user can go the *Loads Balance* page. This page presents not only the power produced in real time as can be seen in Figure 5.1-15, but also a balance of the daily, weekly, monthly, and annual energy production and consumption values as explained in the previous chapters.

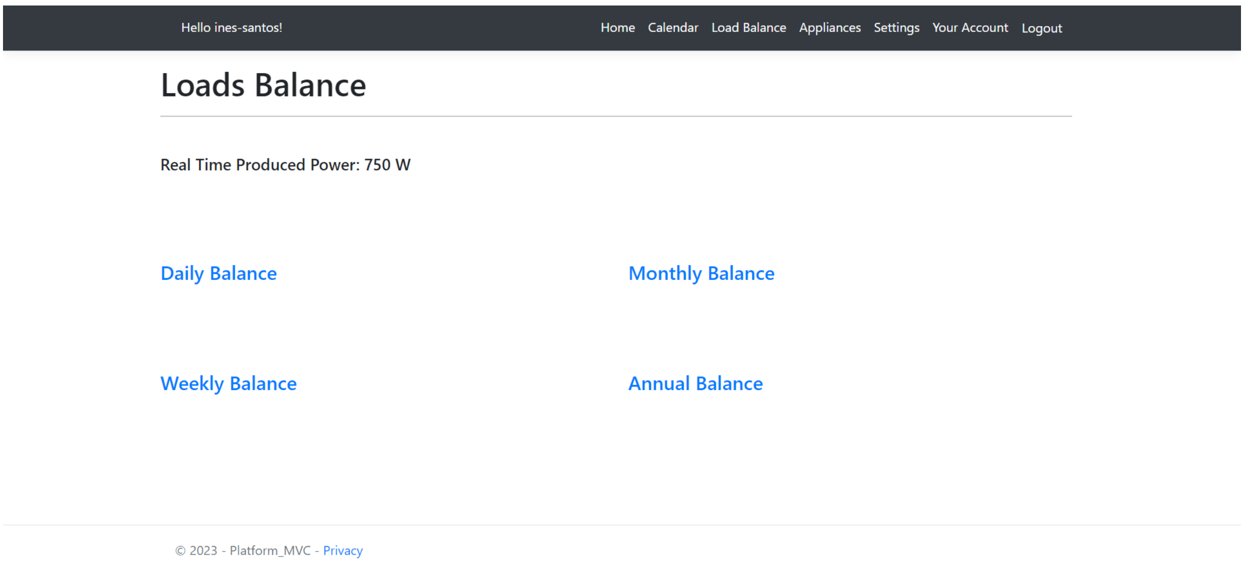


Figure 5.1-15 – *Loads Balance* page

On this page there are four buttons, corresponding to the *Daily Balance*, *Weekly Balance*, *Monthly Balance*, and *Annual Balance* buttons. Each of these buttons, when pressed will take the user to a whole new page where the user can visualize not only a handful of significant values but also a chart that represents the produced power and consumed power, giving the user an easier and more interesting way to analyze data.

To better understand the scheme of the pages corresponding to the Loads Balance, the schematic below, in Figure 5.1-16 contributes to an easier analysis.

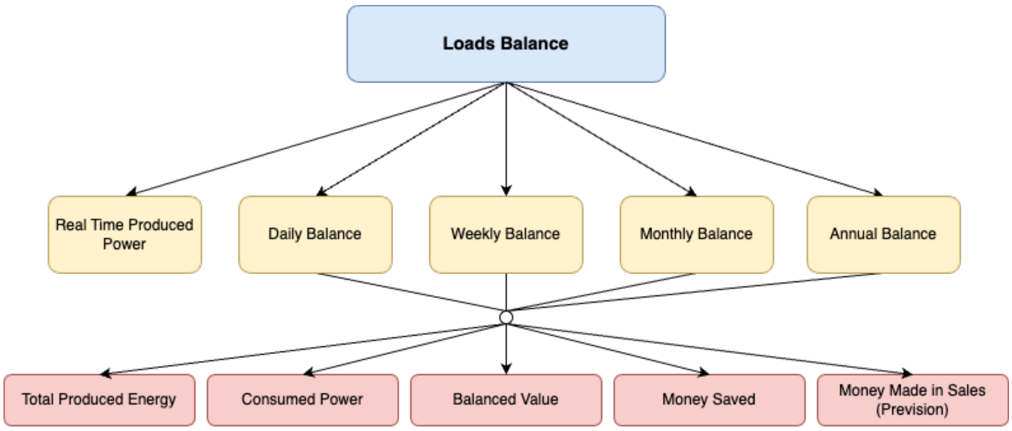


Figure 5.1-16 - Loads Balance schematic

In the figure below it is possible to analyze the monthly balance of the user's scenario.

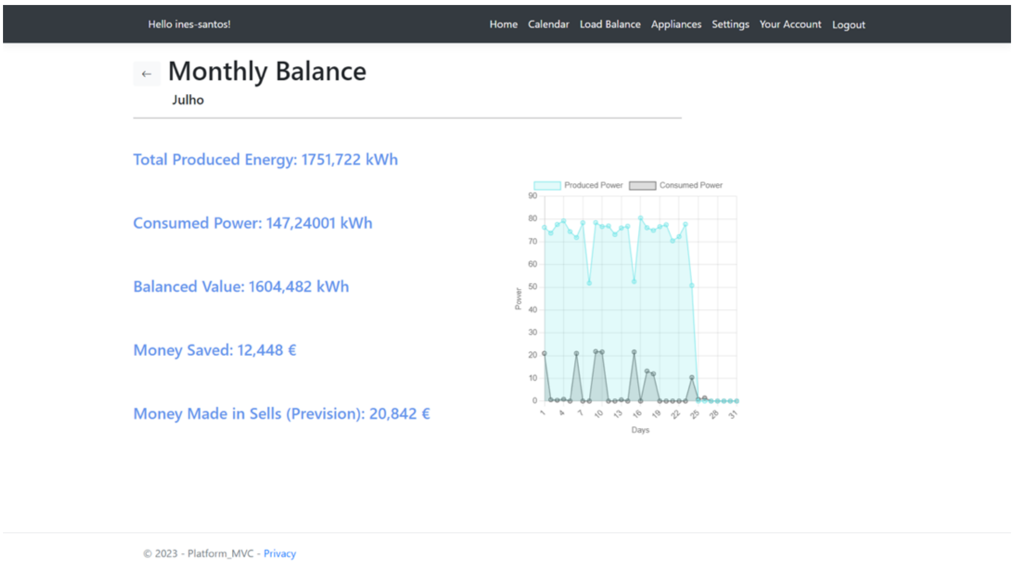


Figure 5.1-17 - *Monthly Balance* page

For more details and screenshots of the rest of the Web Application Platform check appendix A.2.

5.2 Machine Learning Model Results

To put the model to the test, it was necessary to retrieve future values from the same forecast weather API that was mentioned in the previous chapter to ensure the model was working correctly. In the test done, it was retrieved the future values of *Direct Radiation*, *Diffuse Radiation*, *Irradiance*, *Temperature* and *Humidity* up to 16 days in the future from August 21st until September 9th.

Next, after having the values in a clean data frame, all was left to do was test the model with the obtained values from the API and save them in a chart as can be seen in Figure 5.2-1.

By analyzing Figure 5.2-1, it is possible to understand that the model is working well since there can be seen increases and decreases of power production during each day. The maximum value is 929,88W on August 29th at 14h00, which is an accurate value for power production during that time of year at that hour.

Each day starts with a value of power production close to zero and as the day goes by, the power production value increases until around 14h00 or 15h00 which is usually when the sun and irradiance values are highest and therefore, the production values are also highest. After that peak of power production, the production values start continuously decreasing until it returns to zero, when there is no longer value of radiation and irradiance.

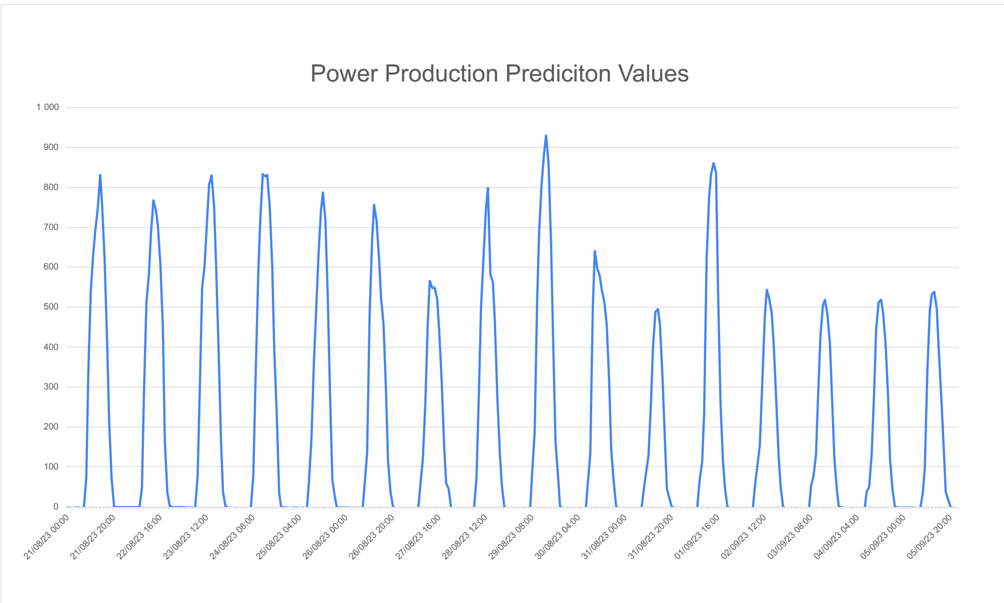


Figure 5.2-1 – Solar Power Production Prediction Values (Active Power)

The values obtained with the machine learning model are considerably accurate when compared to Figure 5.2-2, which represents the real solar power production values in the same time. We can observe that the predicted values are slightly lower most of the days and more consistent. The real values have more variations that can be caused by clouds passing on top of the photovoltaic panel, which alter the values of diffuse and direct radiation, and other sudden variations in temperature or humidity. Overall, the predicted values are considerably accurate.

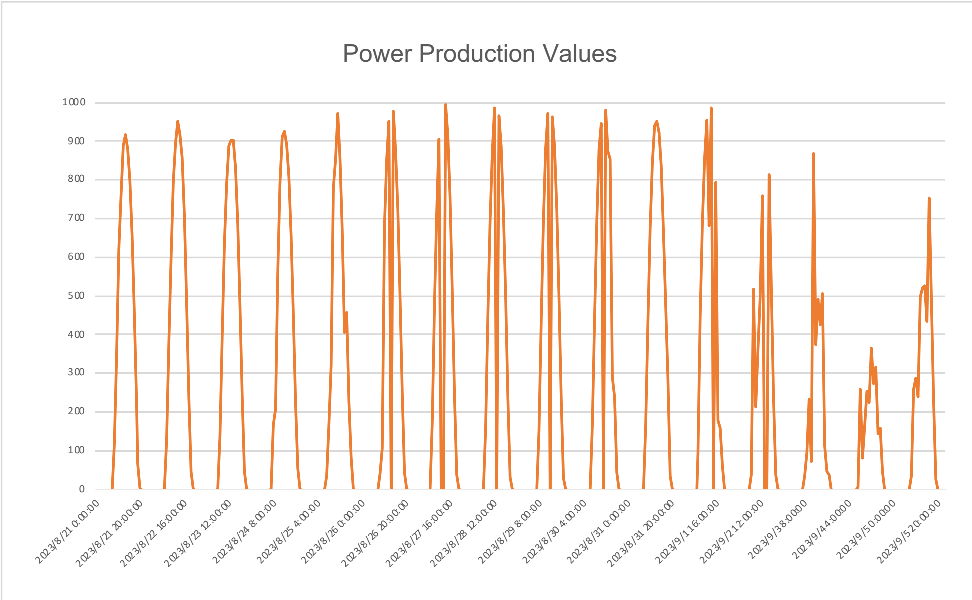


Figure 5.2-2 - Solar Power Production Values (Active Power)

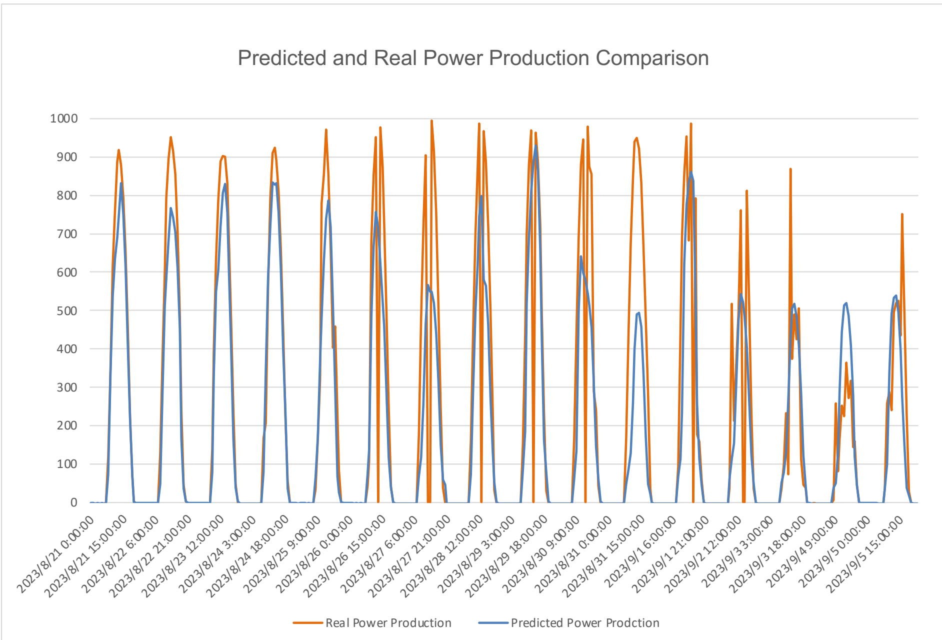


Figure 5.2-3 - Predicted and Real Power Production Values (Active Power)

5.3 Discussion of Results

Overall, the obtained results were positive since the goals of the system developed and the machine learning process were achieved.

In the machine learning model, it can be asserted that the predicted values obtained through the collection and analysis of data were concise and made sense considering the time and date of prevision. However, it is possible to analyze in Figure 5.1-3 that some predictions were below the real production value.

A chart was created to visualize the error percentage between the forecasted and real weather conditions. This chart aimed to determine whether the predictive model's errors were attributed to imprecise weather forecasts or the model's own accuracy. This chart is presented in Figure 5.3-1.

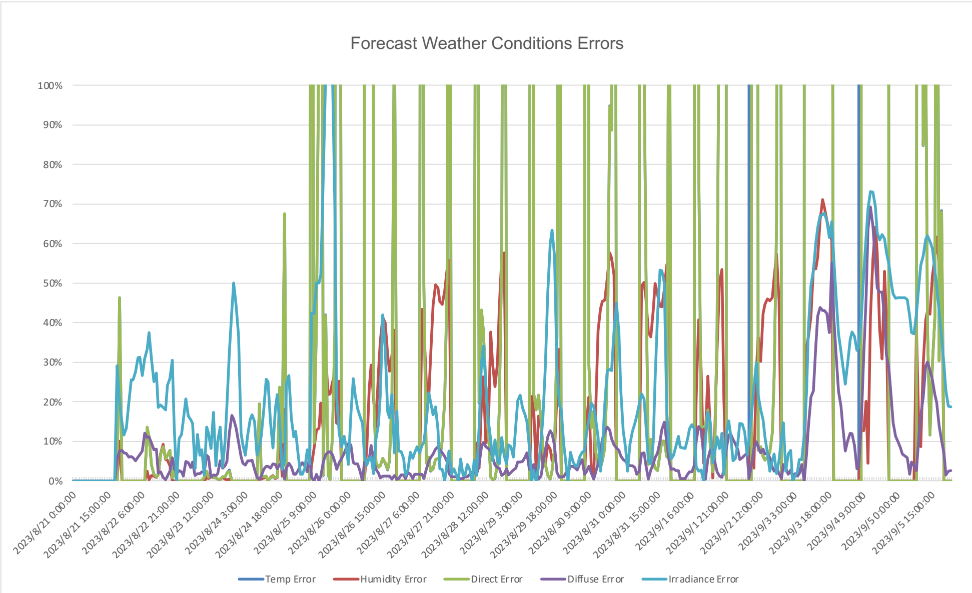


Figure 5.3-1 - Forecast Weather Conditions Errors chart

By analyzing Figure 5.3-1, it is possible to see that on the first days of forecasting the weather conditions, the errors are lower, leading to a more accurate prediction. However, by approximately the fourth day of forecasting, errors begin to rise, lowering the precision of weather condition predictions.

To ensure that a good reading of the value's accuracy is made, a chart with the predictive model's percentage error was created too, presented in Figure 5.3-2.

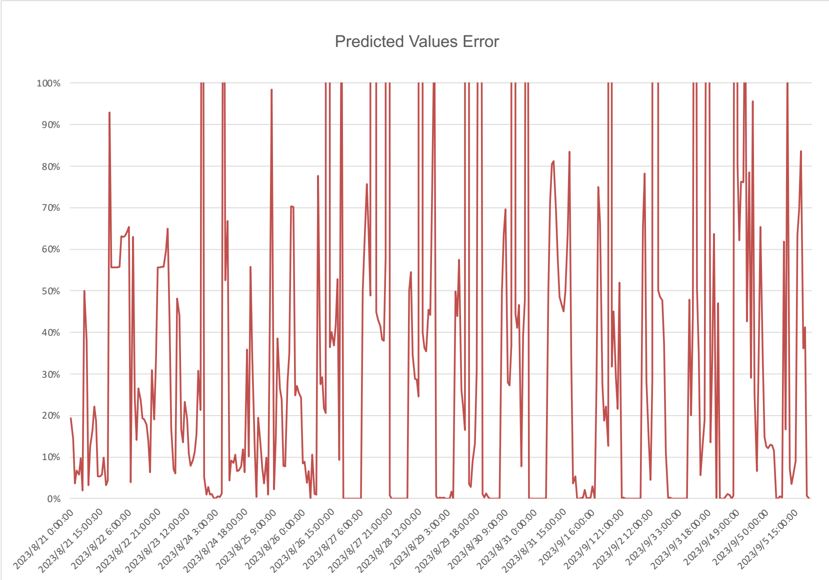


Figure 5.3-2 - Predictive Model Percentage Error chart

After analyzing each situation where the error values were substantially high individually where the error percentage value is considerably higher, above 70%, it can be concluded that almost all of these high percentage errors were due to two situations. The first situation was when the predicted value was above 700 W and the real produced value by the photovoltaic panel in that instant was almost zero. This can be explained by the increase in clouds passing the photovoltaic panel location, contributing to a lower direct radiation and a higher diffuse radiation, which result is a decrease in power production. This situation is of low relevance since the moments are usually of low duration, without influencing the platform's usual functioning. The second situation happened usually at sunrise or sunset hours, where the model would predict a value of around 0 or 1 W, when the real power produced by the photovoltaic panel was around -3W, which resulted in an error percentage above 200%. Therefore, I thought it was a good option to remove each of these particular cases, which are not that relevant to the comparison of results since are either really instant or during time periods where the appliances most likely will not be functioning and create another Error Percentage chart for the predictive model. This new chart, presented in Figure 5.3-3 will only contain relevant situations to the platform's functioning.

When dissecting Figure 5.3-3, a cleaner chart can be seen, where the errors are considerably less and lower. After calculating the mean percentage error, it averages at 21,62%, a considerably lower value than the previously calculated average of around 1561,65%.

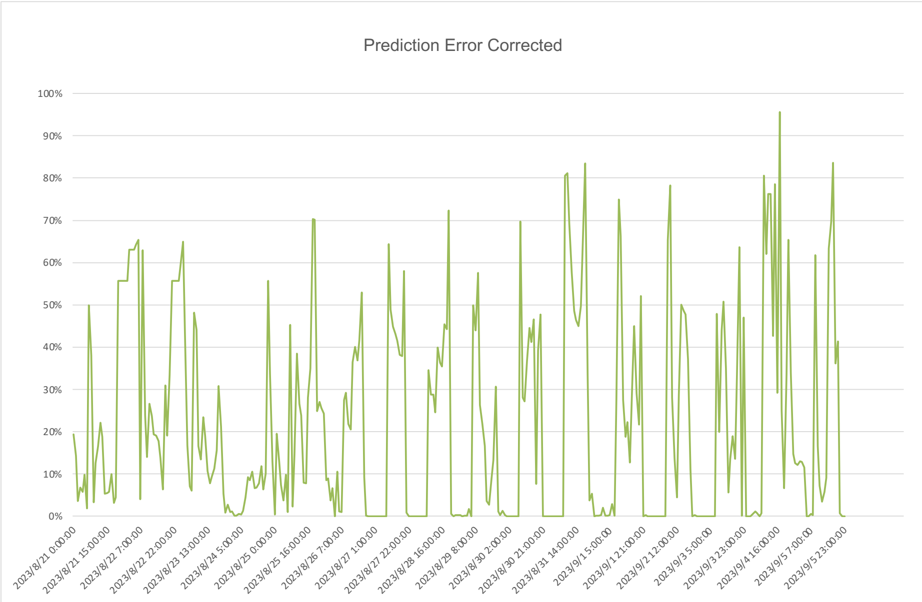


Figure 5.3-3 - Predictive Model Percentage Error chart corrected

Overall, the obtained results in the machine learning model are satisfactory and hopefully, in the future will have more accuracy and precision, lowering the error.

When analyzing the system's functionalities and main goal of automatically scheduling appliances it can be admitted that the system is fully functional with no errors, which was the main goal and challenge. The Web Application automation process is smoothly working, which is one of the most satisfying aspects of the whole dissertation process.

In conclusion, the results, both from the developed prediction model and system, are positive and satisfactory, meaning the main goals were successfully accomplished.

6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

Throughout the course of this extensive dissertation journey, countless fundamental aspects have been meticulously exploring the complex domain of smart home energy management. The *OptiSun* system, the main element of this research project, has been explored and exposed to detailed analysis, understanding its complexities to discover its elaborate functioning. Particularly, its ability to optimize energy usage, lower costs, and enhance user experiences has been strongly emphasized. This highlights its significant role in modern sustainable and eco-friendly living. Among the complexities of this study, it is essential to note that the data used for analysis was real energy consumption patterns, although the appliances themselves were simulated. This methodology facilitated the exploration of the SHEMS functionality, offering different perspectives to examine its efficacy and applicability. It is important to note that the aim is to integrate real appliances into the system.

Essential to this dissertation is the effort to use a predictive model based on supervised learning to forecast Active Power values. The model's success in providing energy production values highlights the game-changing potential of machine learning. This can modernize energy management strategies and lead to more informed decisions in energy utilization.

Notably, in this research work, it can be seen the visible evolution observed between the initial conceptual mockups and the eventual developed platform. The notable differences that appeared in the course of this process were not inconsistencies, but rather intentional enhancements that shaped the step-by-step design and development process that leads to a better and more polished user experience.

As this dissertation work concludes, a deep sense of satisfaction is present. This dissertation, which contributed to an exploration of technology, sustainability, and innovation, has led to an understanding of smart home energy management. With consistent commitment, it dove into complicated details and overcame challenges. As this academic thesis concludes, it's evident that this work not only contributes valuably to knowledge but also resonates as a call to explore limitless possibilities in sustainable and intelligent energy management, seamlessly integrated into modern homes.

6.2 Future Works

As this dissertation concludes, it is clear that the exploration of smart home energy management points to promising future research. This chapter envisions additional functionalities to the work made that could reshape energy-efficient living and home automation. By diving into unfamiliar areas, it inspires new research to innovate within the vast world of smart homes.

This chapter explores possible future works that could easily elevate this dissertation work and make it an even better addition to the smart homes' realm. Ultimately, this chapter promotes efforts to shape a future where smart homes seamlessly integrate efficiency, comfort, and environmental responsibility.

The first functionality that can be easily implemented in the future is to add a current weather widget to allow the user to know what the current weather conditions in the location of the photovoltaic panel are. This can be done by fetching values from an API by choosing the latitude and longitude. By adding this functionality the user would have access to the current and future weather conditions directly on *OptiSun*, without having to check any other apps or websites, facilitating the manual scheduling of the home appliances.

An important update that should be made is to connect real appliances to the web application platform and not only simulate them. Even though the data values are real, this addition would fully connect the system to the real world and allow the use to utilize the platform developed for actual control of home appliances.

To simplify the manual scheduling in the Calendar page, once a time slot is selected, it would be a great addition to already have checked the appliances that are previously scheduled for that time slot, as it was mentioned in chapter 4.3.7 – Calendar. Besides the boxes being checked, it would be great to also have the information of the predicted power production inside the modal of the selected time slot and, once an appliance is selected, the predicted value would be updated by subtracting the consumption value of the selected appliance, from the predicted value, allowing the user to better understand whether it makes sense to schedule the desired appliances manually.

A functionality that would elevate the platform would be to add notifications and warnings as explained and designed in chapter 3.1 - Requirements Identification and Verification. There could also be warnings to let the user know whether there is going to be produced enough energy when scheduling manually either in real time or for future scheduling.

It is also worth noting that there is potential for the predictive model concerning Active Power values to achieve a higher level of precision, enhancing the users experience, and improving the energy management in an even more conscious manner.

Lastly, it would be interesting to create a machine learning model that can analyze user routines and autonomously schedule appliances based on not only the routines themselves, but also seasonal variations. This innovation would empower users to seamlessly delegate the task of energy optimization, allowing them to remain hands-off if desired, while their appliances' scheduling intelligently adapt to their daily routines across different seasons.

References

Al-Ali, A. R., El-Hag, A., Bahadiri, M., Harbaji, M., & Ali El Haj, Y. (2011). Smart home renewable energy management system. *Energy Procedia*, *12*, 120–126. https://doi.org/10.1016/j.egypro.2011.10.017

Alaa, M., Zaidan, A. A., Zaidan, B. B., Talal, M., & Kiah, M. L. M. (2017). A review of smart home applications based on Internet of Things. In *Journal of Network and Computer Applications* (Vol. 97, pp. 48–65). Academic Press. https://doi.org/10.1016/j.jnca.2017.08.017

Alblawi, A., Elkholy, M. H., & Talaat, M. (2019). ANN for assessment of energy consumption of 4 kW PV modules over a year considering the impacts of temperature and irradiance. *Sustainability (Switzerland)*, *11*(23). https://doi.org/10.3390/su11236802

Alharin, A., Doan, T. N., & Sartipi, M. (2020). Reinforcement learning interpretation methods: A survey. *IEEE Access*, *8*, 171058–171077. https://doi.org/10.1109/ACCESS.2020.3023394

Alilou, M., Tousi, B., & Shayeghi, H. (2020). Home energy management in a residential smart micro grid under stochastic penetration of solar panels and electric vehicles. *Solar Energy*, *212*, 6–18. https://doi.org/10.1016/J.SOLENER.2020.10.063

ArcGIS Pro 3.0. (n.d.). *How Extra trees classification and regression algorithm works*. Retrieved August 8, 2023, from https://pro.arcgis.com/en/pro-app/3.0/tool-reference/geoai/how-extra-tree-classification-and-regression-works.htm

Avramenko, Y., Nyström, L., & Kraslawski, A. (2002). Selection of Internals for Reactive Distillation Column—Case-based Reasoning Approach. *Computer Aided Chemical Engineering*, *10*(C), 157–162. https://doi.org/10.1016/S1570-7946(02)80054-2

Banoula, M. (2023). *Machine Learning Steps: A Complete Guide!* https://www.simplilearn.com/tutorials/machine-learning-tutorial/machine-learning-steps

Bertoldi, P., Serrenho, T., & European Commission. Joint Research Centre. (2019). *Smart home and appliances : state of the art : energy, communications, protocols, standards.*

Boldyrev, A. A., & Herasymenko, N. V. (2020). *The Usage of Solar Panels Advantages and Disadvantages*. http://tehnonovator.com.ua/ua/energy-ua/sun-batteryua/types-sun-battery-ua.htm

Breiman, L. (University of C. B. (2001). Random Forests. *International Journal of Advanced Computer Science and Applications*. https://doi.org/10.14569/ijacsa.2016.070603

Chauhan, R. K., Chauhan, K., & Badar, A. Q. H. (2022). Optimization of electrical energy waste in house using smart appliances management System-A case study. *Journal of Building Engineering*, *46*. https://doi.org/10.1016/j.jobe.2021.103595

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 13-17-Augu*, 785–794. https://doi.org/10.1145/2939672.2939785

Conceição, M. (2021). *Previsão de Produção de Energia Renovável para uma comunidade Energética*.

Da Silva, N. G. (2022). *O USO DE PAINÉIS SOLARES NAS EDIFICAÇÕES*.

El-Azab, R. (2021). Smart homes: Potentials and challenges. In *Clean Energy* (Vol. 5, Issue 2, pp. 302–315). Oxford University Press. https://doi.org/10.1093/ce/zkab010

El Naqa, I., & Murphy, M. J. (2015). *What Is Machine Learning? BT - Machine Learning in Radiation Oncology: Theory and Applications* (I. El Naqa, R. Li, & M. J. Murphy (Eds.); pp. 3–11). Springer International Publishing. https://doi.org/10.1007/978-3-319-18305-3_1

Elkholy, M. H., Senjyu, T., Lotfy, M. E., Elgarhy, A., Ali, N. S., & Gaafar, T. S. (2022). Design and Implementation of a Real-Time Smart Home Management System Considering Energy Saving. *Sustainability*, *14*(21), 13840. https://doi.org/10.3390/su142113840

Fumo, D. (2017). *Types of Machine Learning Algorithms You Should Know*. Towards Data Science. https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861

Gunge, V. S., & Yalagi, P. S. (2016). Smart Home Automation: A Literature Review. In *International Journal of Computer Applications*.

Gupta, A. (2021). *Steps to Complete a Machine Learning Project*.

H. Friedman, J. (2001). Greedy Function Approximation: a Gradient Boosting Machine. *Annals of Statistics*, *2*, 34. http://www.autonlab.org/tutorials/infogain.html%0Ahttp://www.jstor.org/stable/10.2307/2699986

Jakhongir Turakul ugli, T. (2019). *International Journal of Engineering and Information Systems (IJEAIS) The Importance of Alternative Solar Energy Sources and the Advantages and Disadvantages of Using Solar Panels in this Process*. www.ijeais.org

John Wiley & Sons. (2022). *Machine Learning and the City: Applications in Architecture and Urban Design*.

Kannan, N., & Vakeesan, D. (2016). Solar energy for future world: - A review. In *Renewable and Sustainable Energy Reviews* (Vol. 62, pp. 1092–1105). Elsevier Ltd. https://doi.org/10.1016/j.rser.2016.05.022

Kavafia, H. (2023, March). *The best smart home apps for your Android phone*. Android Police. https://www.androidpolice.com/best-smart-home-apps-for-android/

Kharwal, A. (2021, June 1). *Explained Variance in Machine Learning*. https://thecleverprogrammer.com/2021/06/25/explained-variance-in-machine-learning/

Kumar, A., Kim, H., & Hancke, G. P. (2013). Environmental monitoring systems: A review. *IEEE Sensors Journal*, *13*(4), 1329–1339. https://doi.org/10.1109/JSEN.2012.2233469

Lei, M., Shiyan, L., Chuanwen, J., Hongling, L., & Yan, Z. (2009). A review on the forecasting of wind speed and generated power. In *Renewable and Sustainable Energy Reviews* (Vol. 13, Issue 4, pp. 915–920). https://doi.org/10.1016/j.rser.2008.02.002

Li, Y. F., & Liang, D. M. (2019). Safe semi-supervised learning: a brief introduction. In *Frontiers of Computer Science* (Vol. 13, Issue 4, pp. 669–676). Higher Education Press. https://doi.org/10.1007/s11704-019-8452-2

Lobaccaro, G., Carlucci, S., & Löfström, E. (2016). A review of systems and technologies for smart homes and smart grids. In *Energies* (Vol. 9, Issue 5). MDPI AG. https://doi.org/10.3390/en9050348

Maghami, M. R., Hizam, H., Gomes, C., Radzi, M. A., Rezadad, M. I., & Hajighorbani, S. (2016). Power loss due to soiling on solar panel: A review. In *Renewable and Sustainable Energy Reviews* (Vol. 59, pp. 1307–1316). Elsevier Ltd. https://doi.org/10.1016/j.rser.2016.01.044

Mansouri-Samani, M., & Sloman, M. (1993). Monitoring Distributed Systems: A Functional Model of Monitoring in Terms of the Generation, Processing, Dissemination, and Presentation of Information can Help Determine the Facilities Needed to Design and Construct Distributed Systems. *IEEE Network*, *7*(6), 20–30. https://doi.org/10.1109/65.244791

Md, S., Moh, K. H., Md, M. I., Md, M. A., Md, F. A., & Yeong, M. J. (2020). *An Overview of AI-Enabled Remote Smart-HomeMonitoring System Using LoRa*.

Mohamed, M. A., Eltamaly, A. M., Farh, H. M., & Alolah, A. I. (2015). *Energy Management and Renewable Energy Integration in Smart Grid System*.

Moore, J. (2018). Knowledge-based Systems (KBS). *TechTarget*. https://www.techtarget.com/searchcio/definition/knowledge-based-systems-KBS

Moparthi, N. R., Mukesh, C., & Sagar, P. V. (2018). *Water Quality Monitoring System Using IOT*.

Nfaoui, M., & El-Hami, K. (2018). Extracting the maximum energy from solar panels. *Energy Reports*, *4*, 536–545. https://doi.org/10.1016/j.egyr.2018.05.002

Petersson, D., & Gillis, A. S. (2023). *Supervised Learning*. https://www.techtarget.com/searchenterpriseai/definition/supervised-learning

Photovoltaic Geographical Information System. (n.d.). 2023. Retrieved January 19, 2023, from https://re.jrc.ec.europa.eu/pvg_tools/en/

Pratt, M. K., & Gillis, A. S. (2023). *Unsupervised Learning*. https://www.techtarget.com/searchenterpriseai/definition/unsupervised-learning

Pros and Cons of Solar Battery Storage – Is It Worth It? (n.d.). Retrieved January 19, 2023, from https://instylesolar.com/blog/pros-and-cons-of-installing-a-solar-battery

Rahim, S., Javaid, N., Ahmad, A., Khan, S. A., Khan, Z. A., Alrajeh, N., & Qasim, U. (2016). Exploiting heuristic algorithms to efficiently utilize energy management controllers with renewable energy sources. *Energy and Buildings*, *129*, 452–470. https://doi.org/10.1016/j.enbuild.2016.08.008

Reis, P. (2019). *What is the price of batteries for a solar photovoltaic system?* https://www.portal-energia.com/preco-baterias-sistema-solar-fotovoltaico-147012/

Robles, R. J., & Kim, T.-H. (2010). Applications, Systems and Methods in Smart Home Technology: A Review. In *International Journal of Advanced Science and Technology* (Vol. 15).

Singh, A., Thakur, N., & Sharma, A. (2016). *A Review of Supervised Machine Learning Algorithms*.

Startup Info Staff. (2022). *Most Common Problems with Smart Home Technology Systems*. https://startup.info/most-common-problems-with-smart-home-technology-systems/

Swaminathan, V. (n.d.). The Conundrum of Using Rule-Base vs. Machine Learning Systems. *Zuci Systems*. Retrieved January 27, 2023, from https://www.zucisystems.com/blog/the-conundrum-of-using-rule-based-vs-machine-learning-systems/

Thomas Fielding, R. (University of C. (2000). *Architectural Styles and the Design of Network-based Software Architectures* (Issue 1645).

tray.io. (n.d.). *How do APIs work?* Retrieved August 23, 2023, from https://tray.io/blog/how-do-apis-work

Tsujioka, Y., Akmal, S., Takada, Y., Kawai, H., & Batres, R. (2012). Semantic similarity for case-based reasoning in the context of GMP. *Computer Aided Chemical Engineering*, *31*, 830–834. https://doi.org/10.1016/B978-0-444-59507-2.50158-X

Usama, M., Qadir, J., Raza, A., Arif, H., Yau, K. L. A., Elkhatib, Y., Hussain, A., & Al-Fuqaha, A. (2019). Unsupervised Machine Learning for Networking: Techniques, Applications and Research Challenges. *IEEE Access*, *7*, 65579–65615. https://doi.org/10.1109/ACCESS.2019.2916648

Vashist, S. K. (2013). Continuous Glucose Monitoring Systems: A Review. In *Diagnostics* (Vol. 3, Issue 4, pp. 385–412). MDPI. https://doi.org/10.3390/diagnostics3040385

What is a rules based system in AI? (n.d.). Retrieved January 27, 2023, from https://www.engati.com/glossary/rule-based-system

Zanikolas, S., & Sakellariou, R. (2005). A taxonomy of grid monitoring systems. *Future Generation Computer Systems*, *21*(1), 163–188. https://doi.org/10.1016/J.FUTURE.2004.07.002

Zhu, J., Lin, Y., Lei, W., Liu, Y., & Tao, M. (2019). Optimal household appliances scheduling of multiple smart homes using an improved cooperative algorithm. *Energy*, *171*, 944–955. https://doi.org/10.1016/j.energy.2019.01.025

A

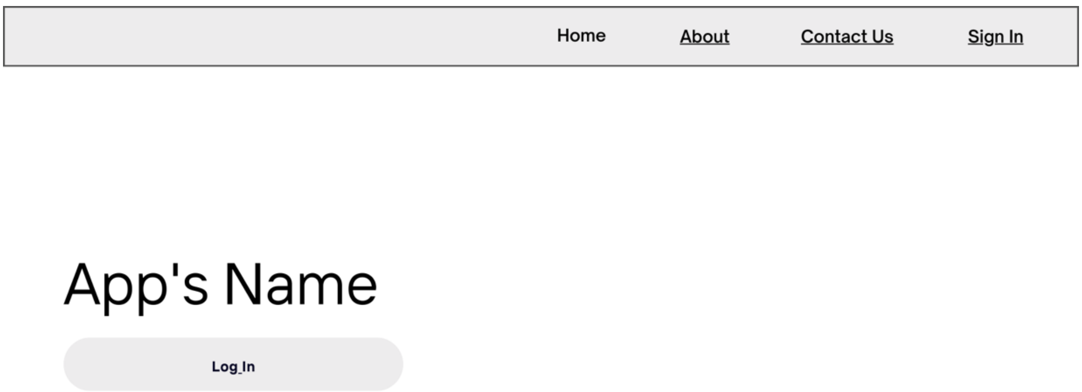
Web Application Appendix

A.1 Web Application Mockups

In this appendix chapter, it is presented a comprehensive collection of application mockups that visually represent the user interface and interaction design of the system. These mockups serve as illustrative representations of the proposed software, showcasing the layout, features, and user flow in a perceptible way. The mockups provide a preview of how users will engage with the application and offer insights into the overall user experience.

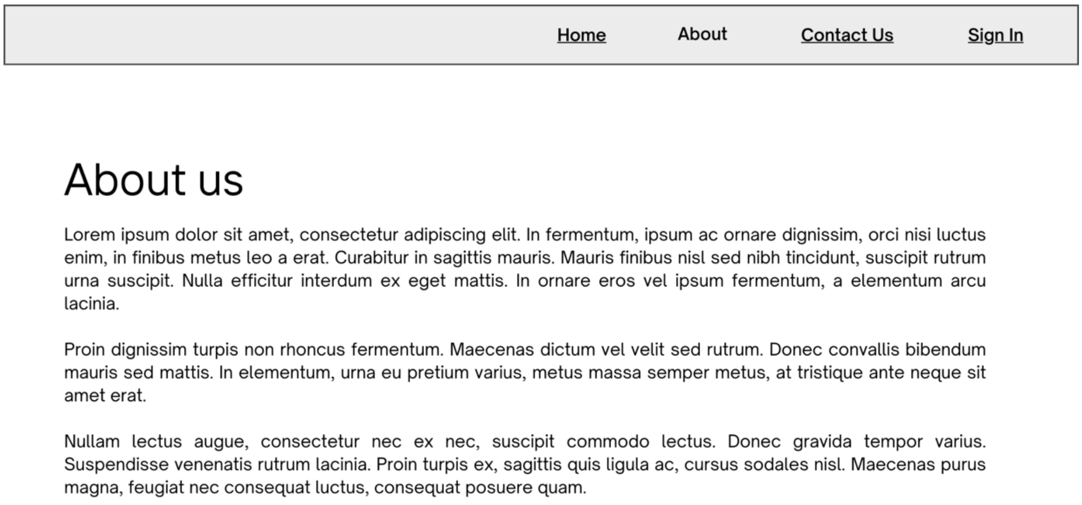
This appendix is complementary to chapter 3.1 - Requirements Elicitation for Software Development.

Once the user accesses the web application platform, the initial home page is presented. This page allows the user to log into their account as can be seen in the figure below.

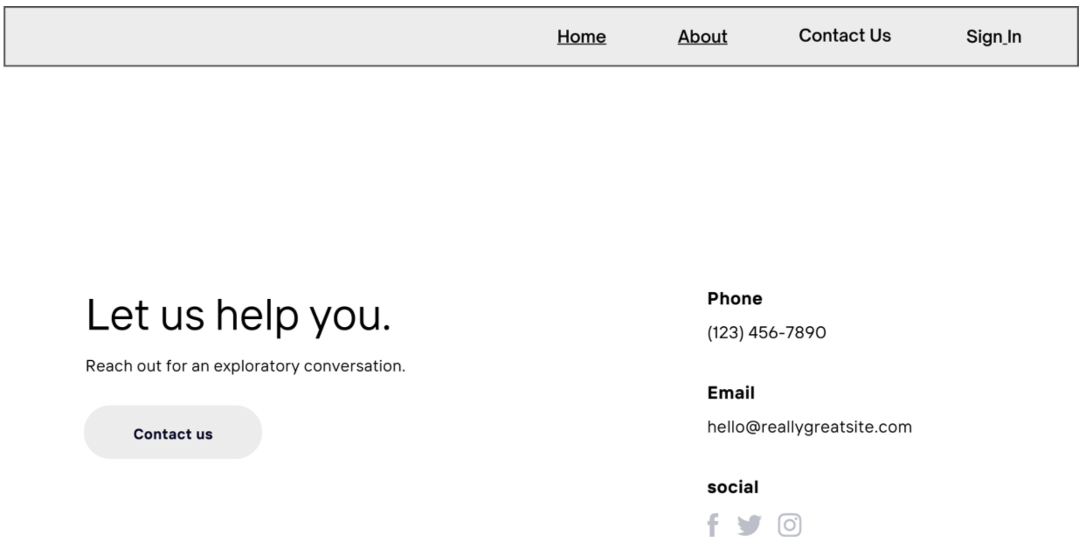


Appendix Figure 1 - *Home Page* logged off mockup

There was also designed a page dedicated to understanding more about the web application platform in question, presented in Appendix Figure 2 and another page where anyone can contact the team to ask questions or even explain problems they might be having with the system presented in Appendix Figure 3.

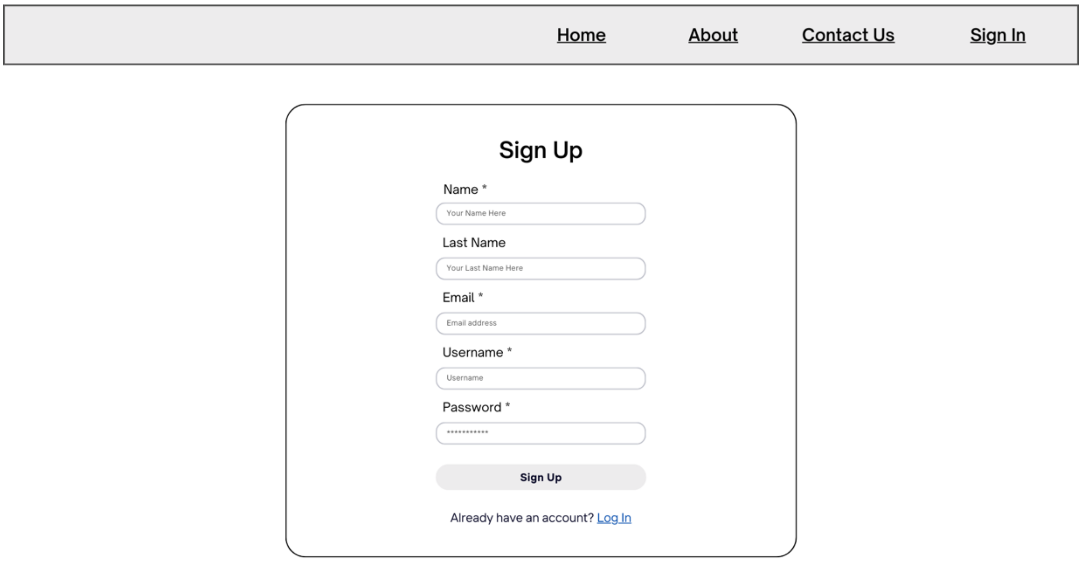


Appendix Figure 2 - *About Us* page mockup



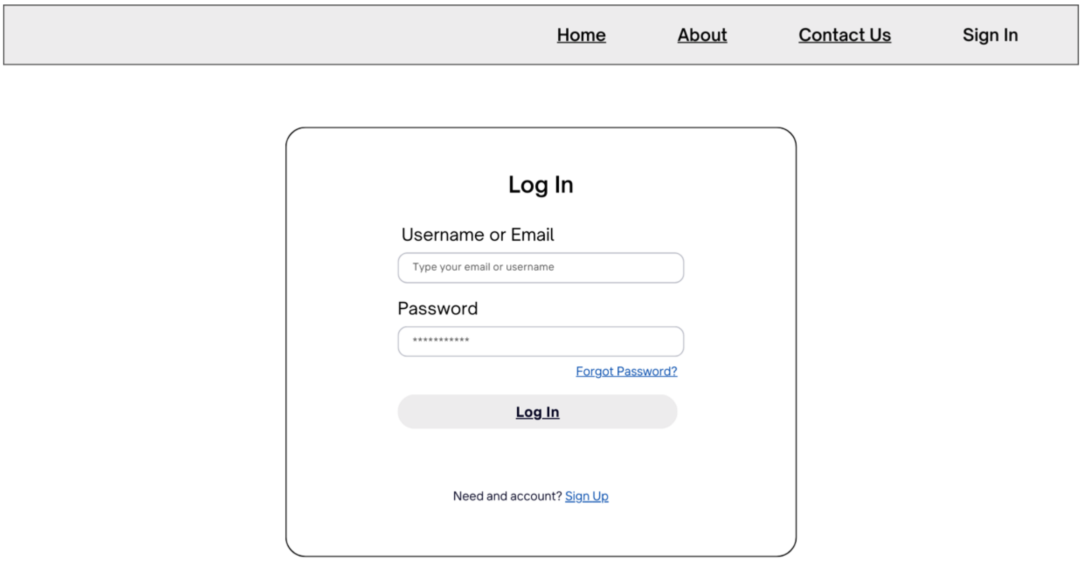
Appendix Figure 3 - *Contact Us* page mockup

To start using the platform the user must create an account by signing up, as presented in Appendix Figure 4.



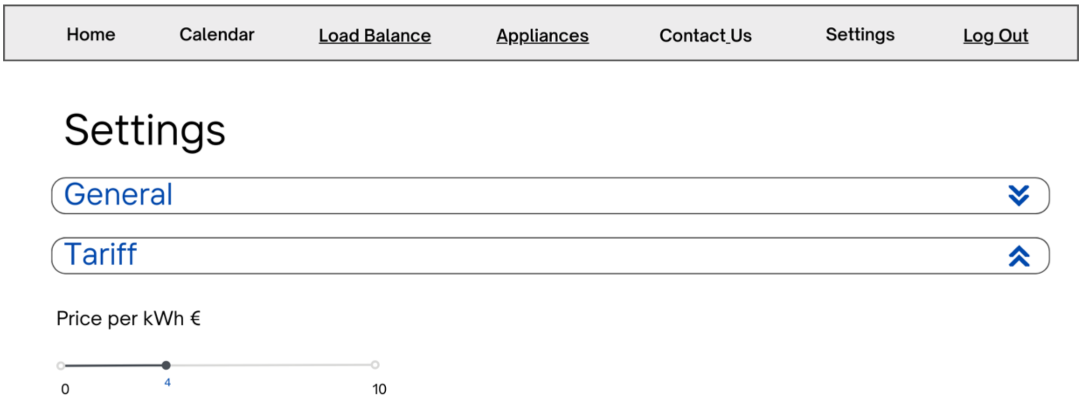
Appendix Figure 4 - *Sign Up* page mockup

Once the account is created the user can log in.



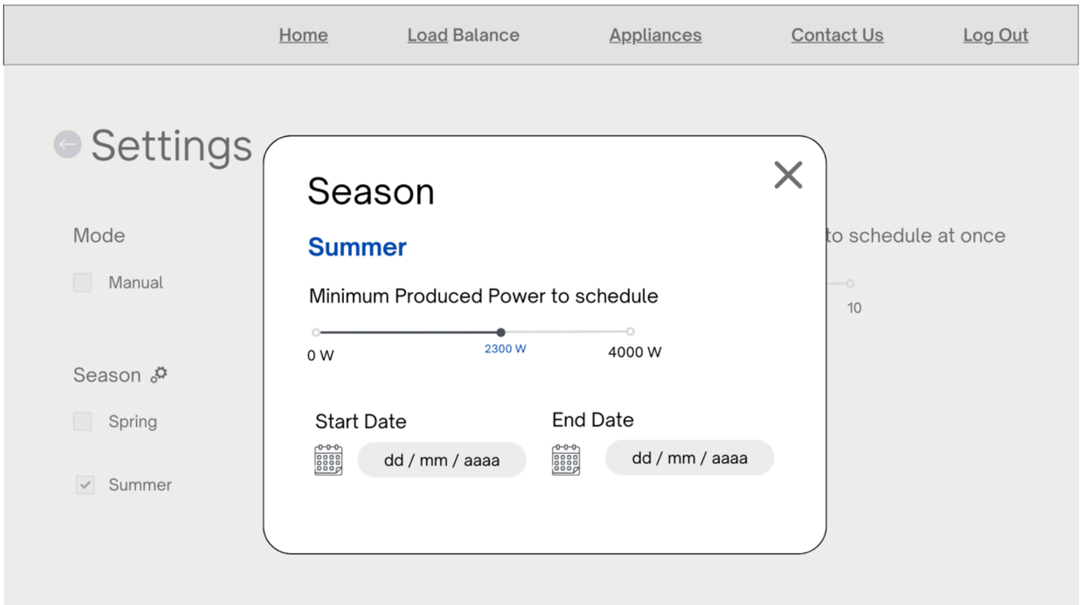
Appendix Figure 5 - *Log In* page mockup

As mentioned in chapter 3.1 the user once logged in can change the system settings. The *Tariff* settings are presented in Appendix Figure 6, as can be seen below.



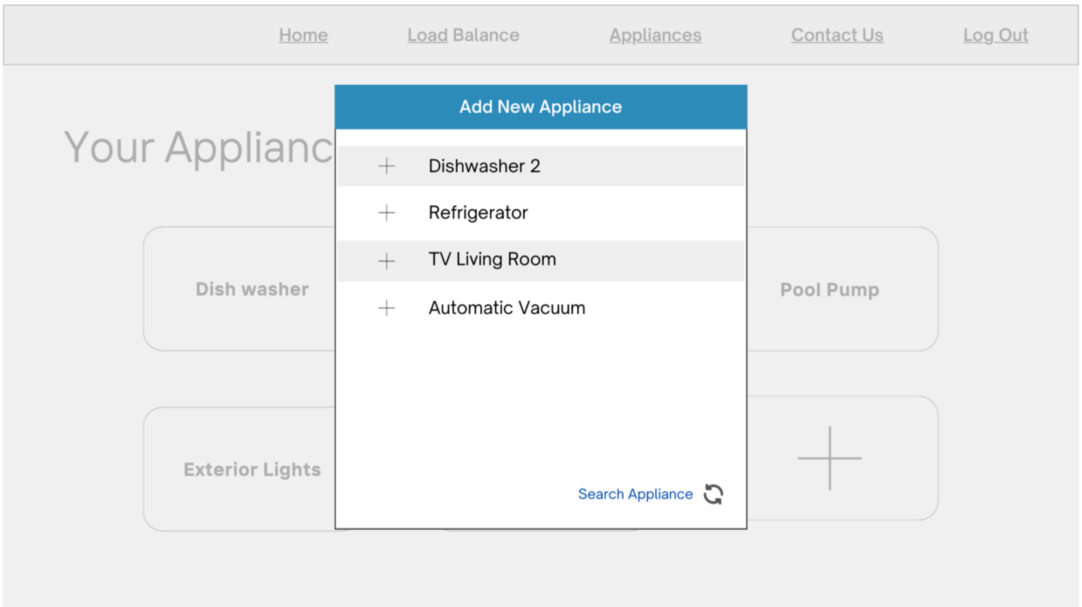
Appendix Figure 6 - *Tariff Settings* page mockup

The *General Settings* have a *Season Settings* parameter, in which the user can define when each season starts and ends, and the minimum power being produced to schedule during each specific season. The *Season Settings* is represented in Appendix Figure 7, below.



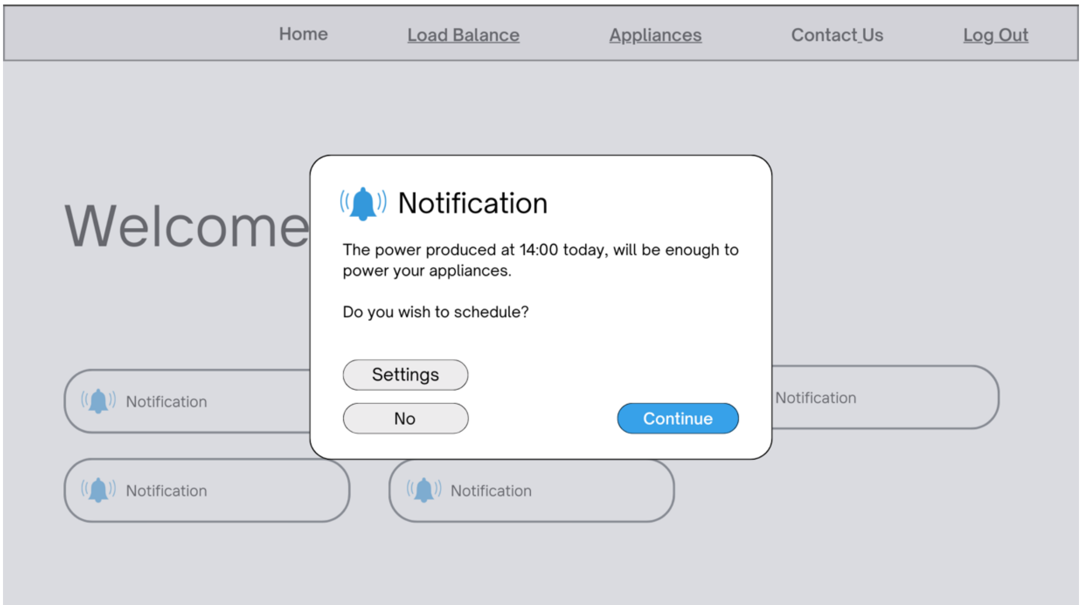
Appendix Figure 7 - *Season Settings* mockup

To add an appliance, the user should be able to connect an appliance to the system, just like it is represented in Appendix Figure 8.

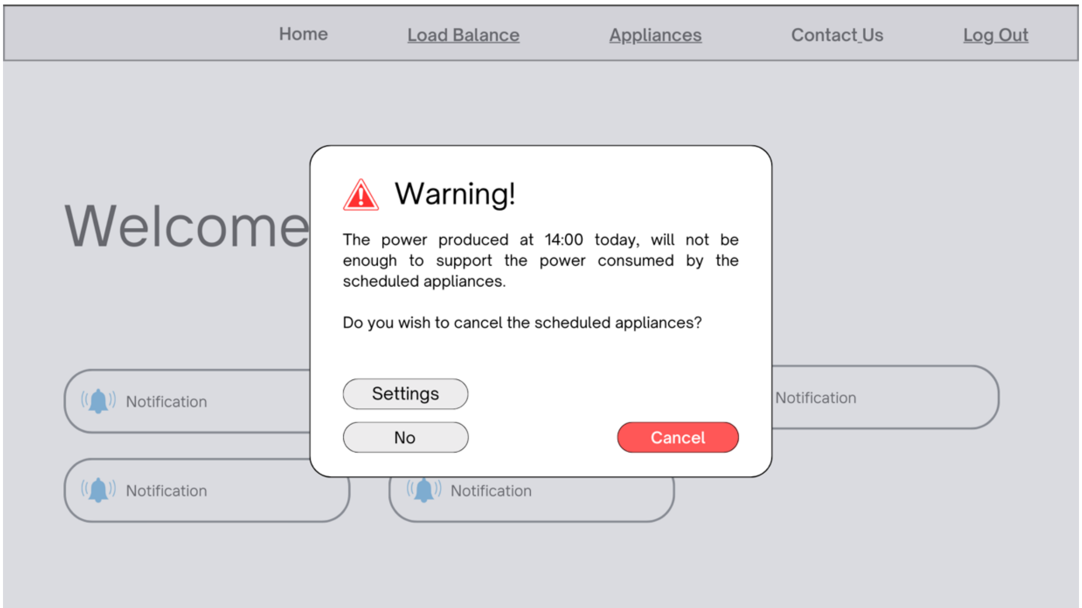


Appendix Figure 8 - *Add New Appliance* mockup

Finally, the notification and warning buttons give important details to the user referring to the scheduled appliances and current power production, as can be analyzed in Appendix Figures 9 and 10.

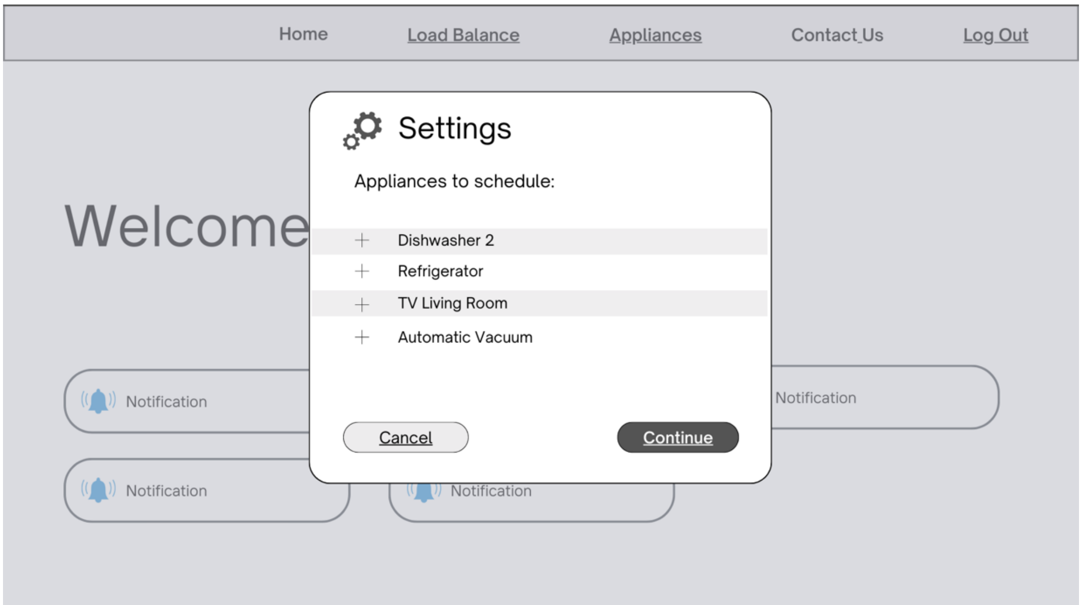


Appendix Figure 9 - *Notification* mockup

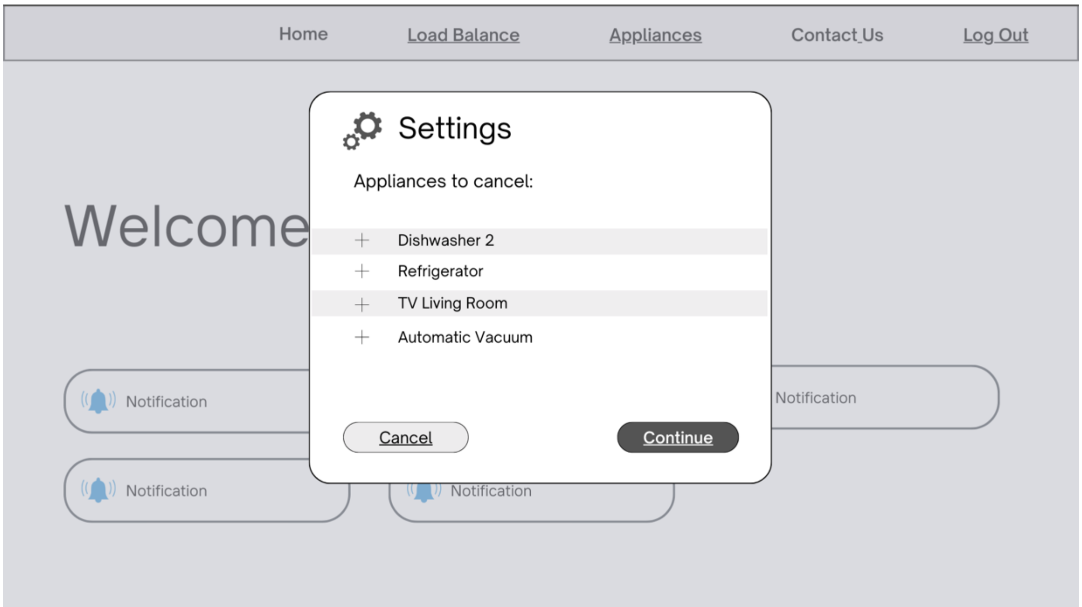


Appendix Figure 10 - *Warning* mockup

The user must decide between scheduling new appliance if a new notification appears or cancel a schedule if a warning appears. The scheduling and canceling process can be customizable for the user's needs as can be seen in Appendix Figures 11 and 12.



Appendix Figure 11 – *Notification's Settings* mockup

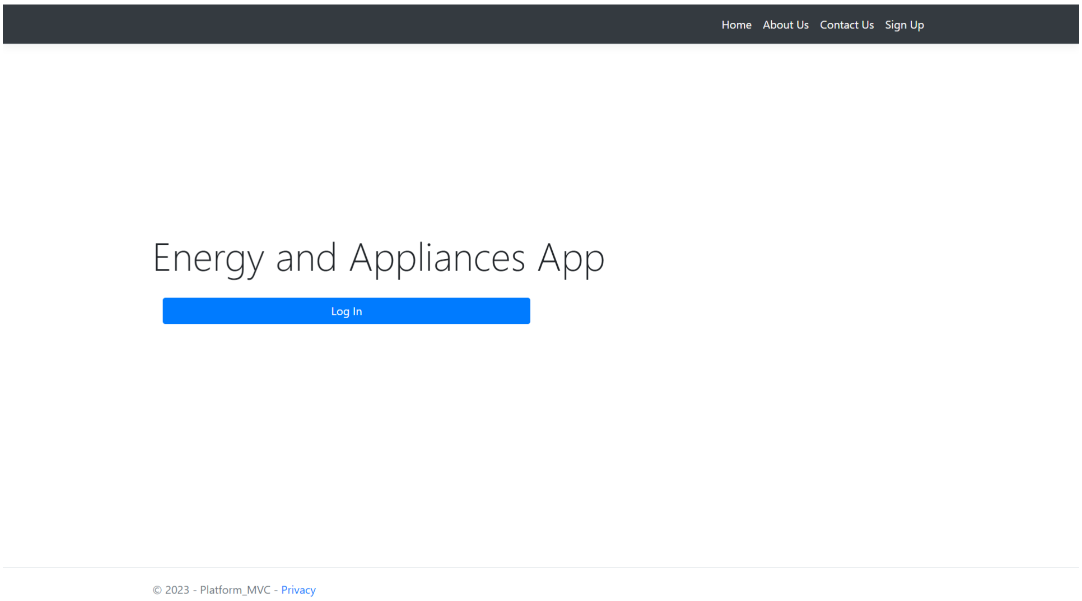


Appendix Figure 12 – *Warning's Settings* mockup

A.2 Web Application Platform

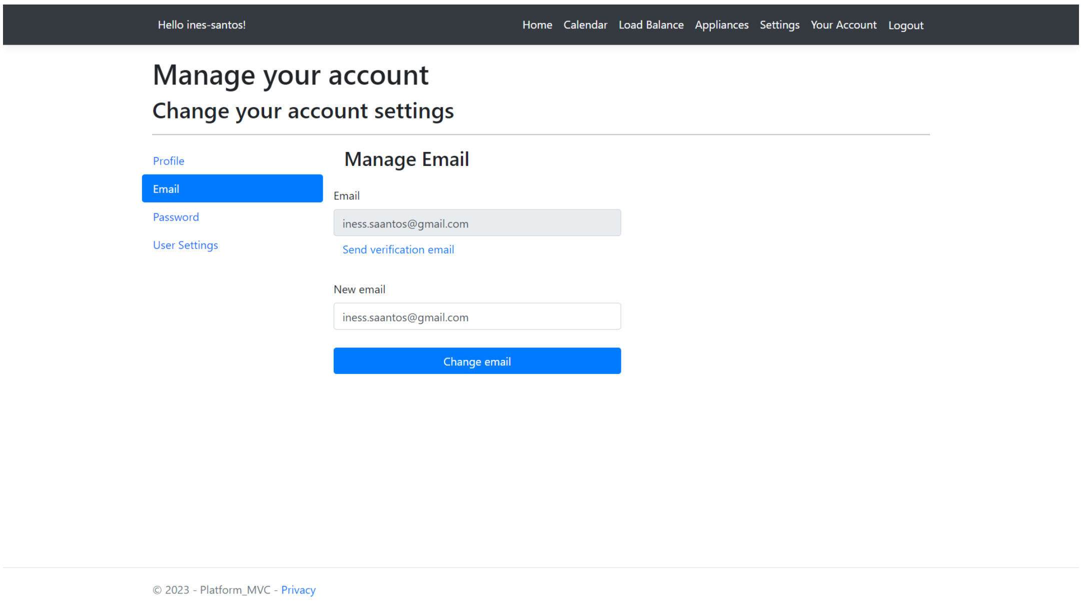
In this section, an extended analysis of the web application platform is provided through a series of screenshots that supplement the information covered in Chapter 5.2 - Web Application Platform Scenario. The screenshots presented here showcase various aspects and functionalities of the web application that were not thoroughly covered in the chapter.

Once a user accesses the Web Application Platform, the *Home Page* logged out is presented, as can be seen in Appendix Figure 13.

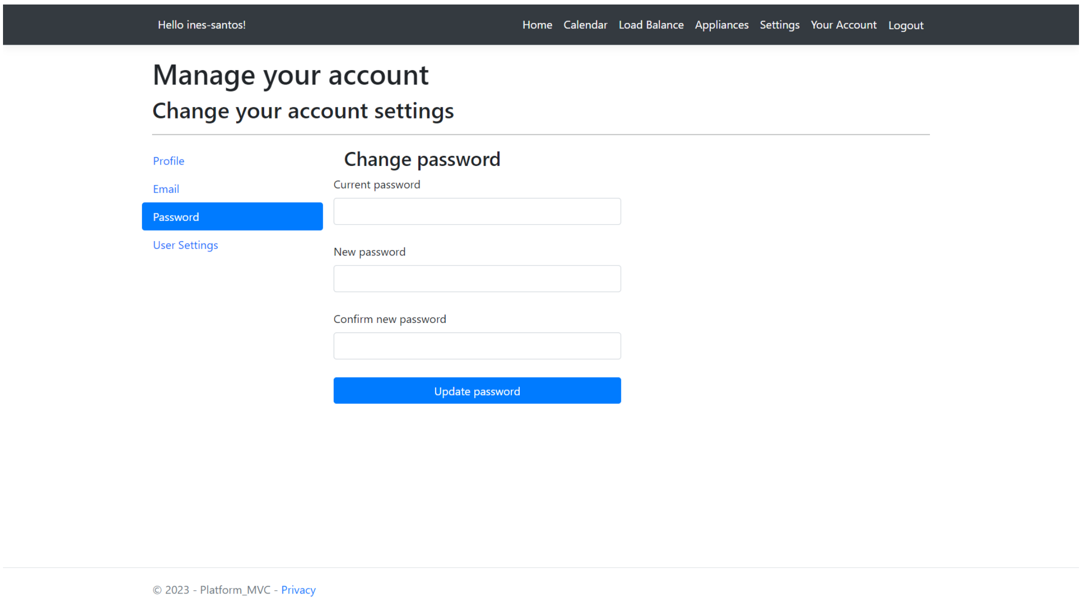


Appendix Figure 13 - *Home Page* logged out

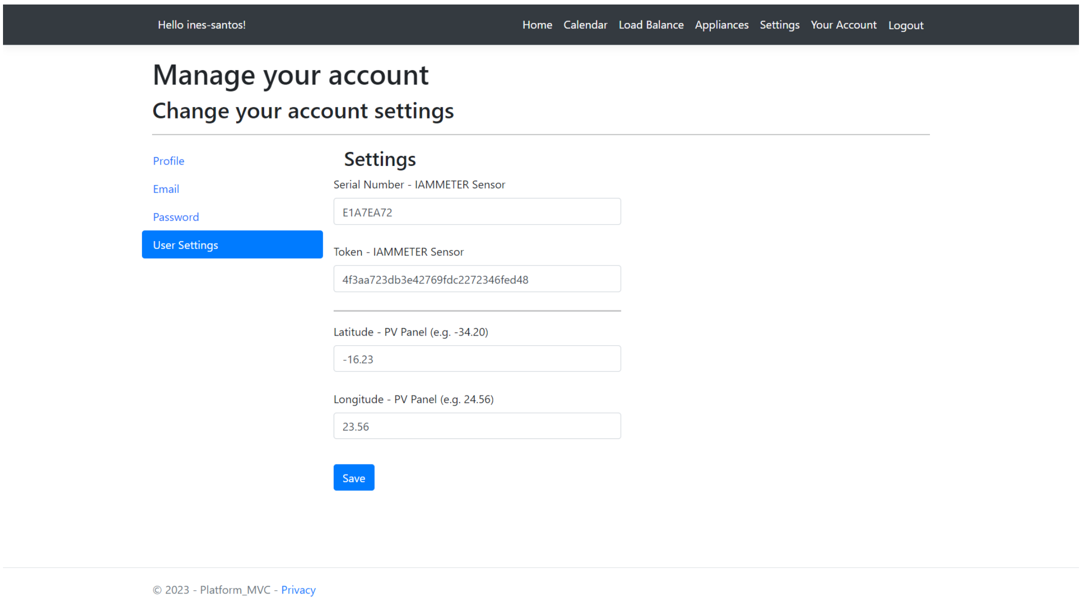
Once logged in, the user can change a few account settings such as password, email, latitude, and longitude and the token and serial number of the IAMMETER sensor as can be seen in the following images. The profile settings were previously covered in chapter 5.2.



Appendix Figure 14 - Account Settings Email

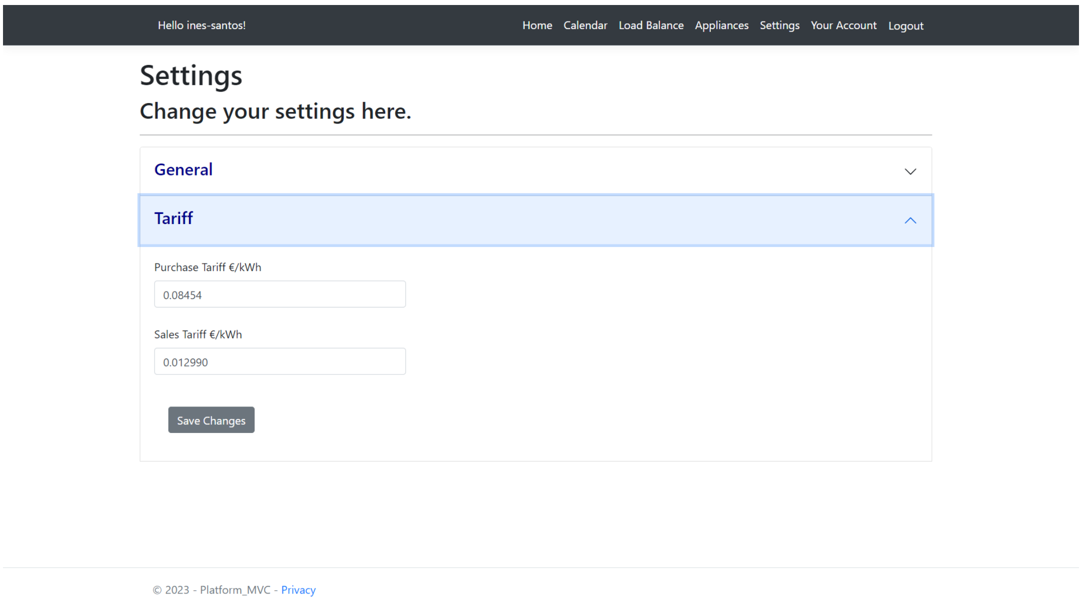


Appendix Figure 15 - Account Settings Password



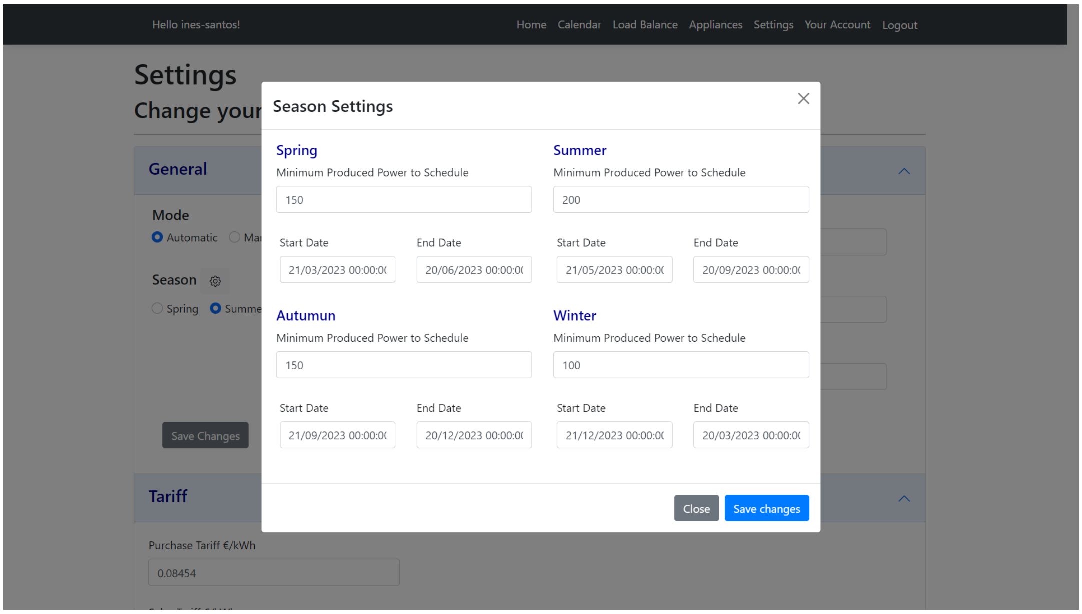
Appendix Figure 16 - Account Settings, User Settings

On the continuation of Chapter 5.2, the *Settings Tariff* page is presented in the figure below, where can be seen both the *Purchase Tariff €/kWh* and *Sales Tariff €/kWh* parameters.



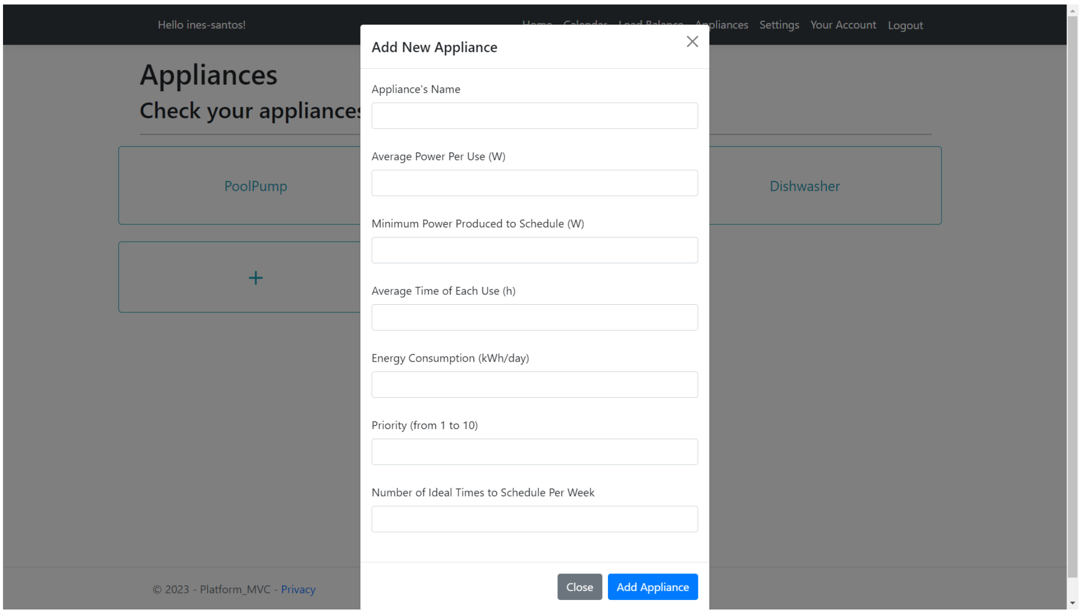
Appendix Figure 17 - *Settings* page, *Tariff Settings*

On the other hand, once selecting the *General Settings* dropdown menu and the *Season Settings* button, the user will be presented with a modal pop up to alter each season's settings, just like it can be seen in Appendix Figure 18.

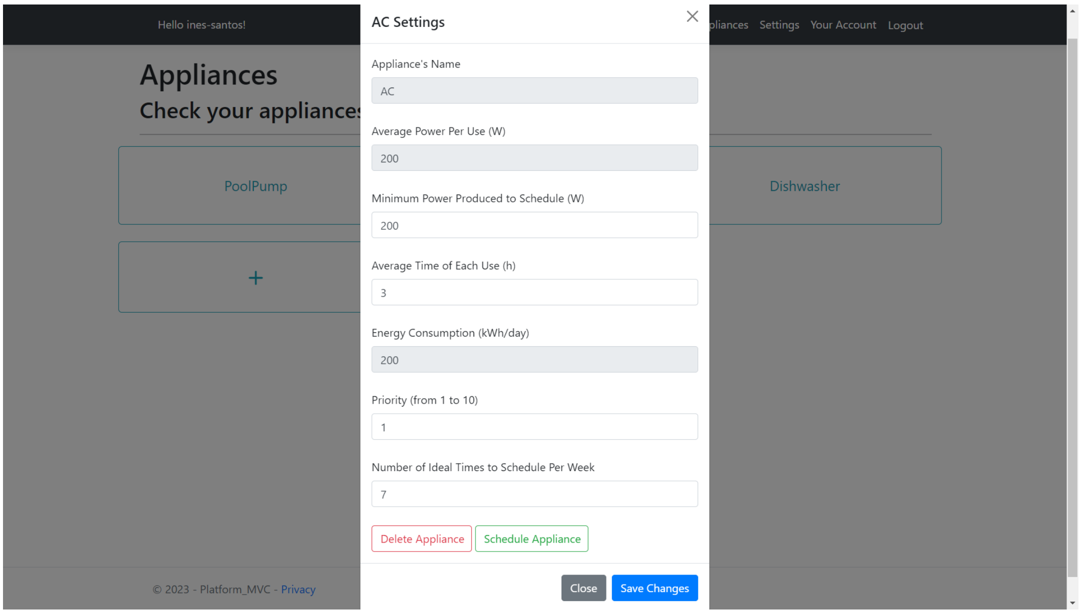


Appendix Figure 18 - *Saeason Settings* modal

When in the *Appliances* page the user can either create a new appliance or make changes to the already existing appliances' settings. Both situations are presented below in Appendix Figures 19 and 20.

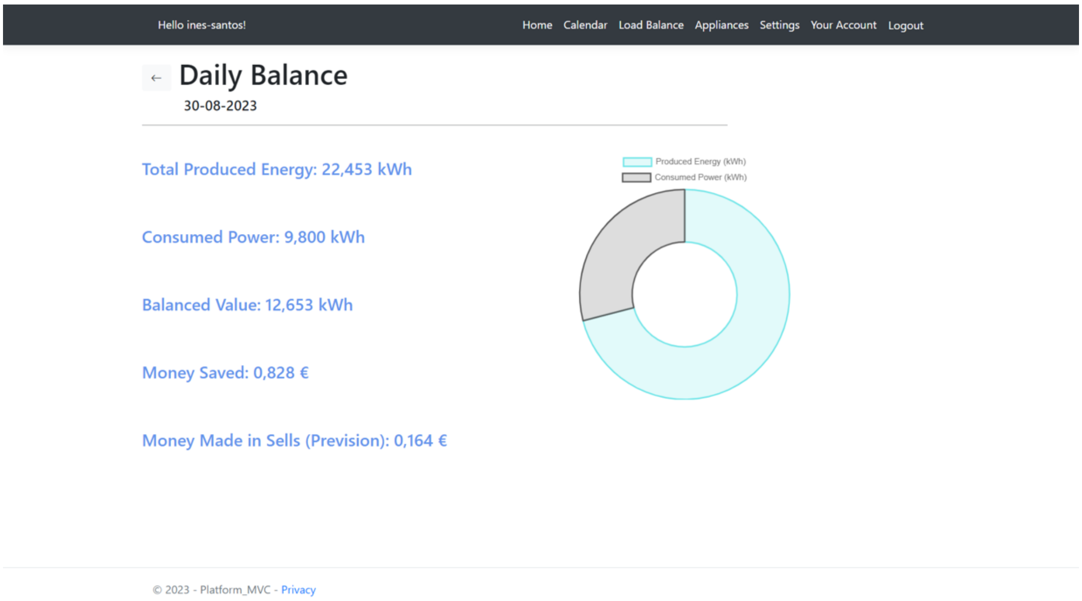


Appendix Figure 19 - *Add new Appliance* modal

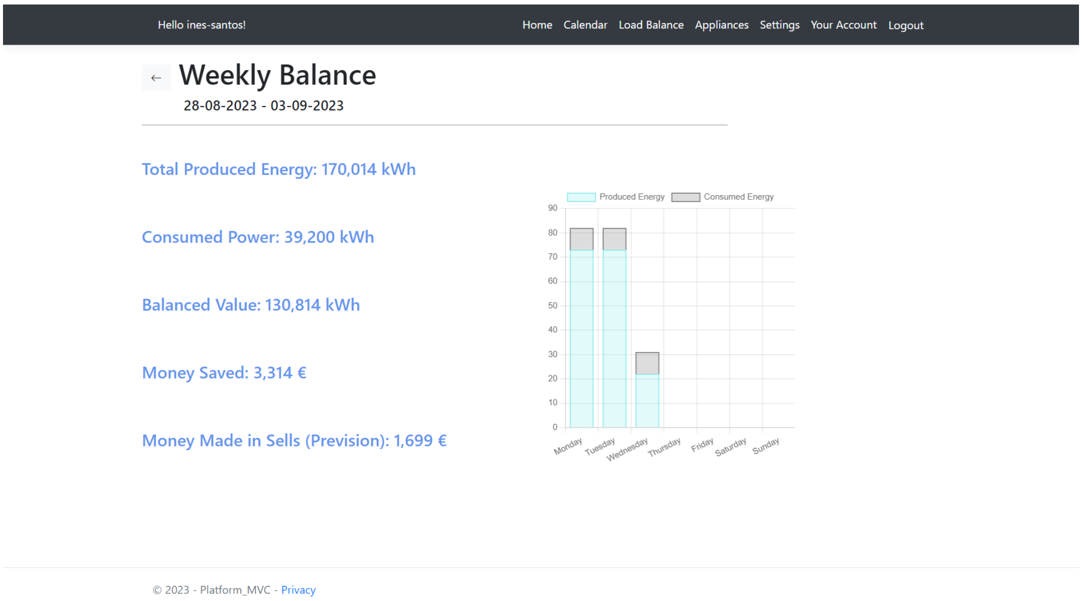


Appendix Figure 20 - Appliance's settings

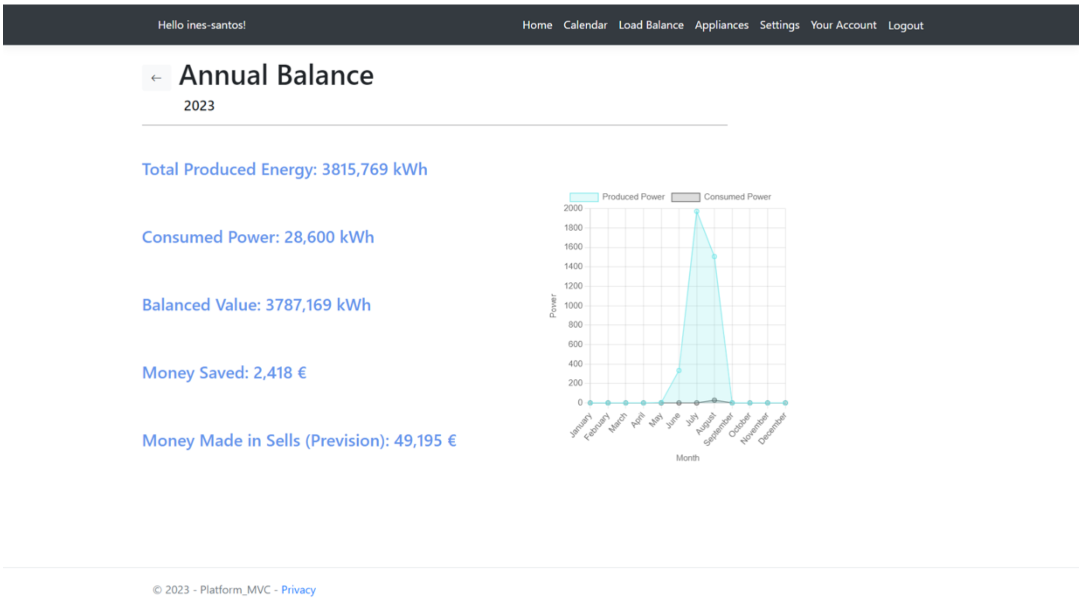
To better analyse the Loads Balance the user can also have information the daily, weekly and annual balances, as can be seen in the following figures, additionally to what was presented in chapter 5.2



Appendix Figure 21 - *Daily Balance* page

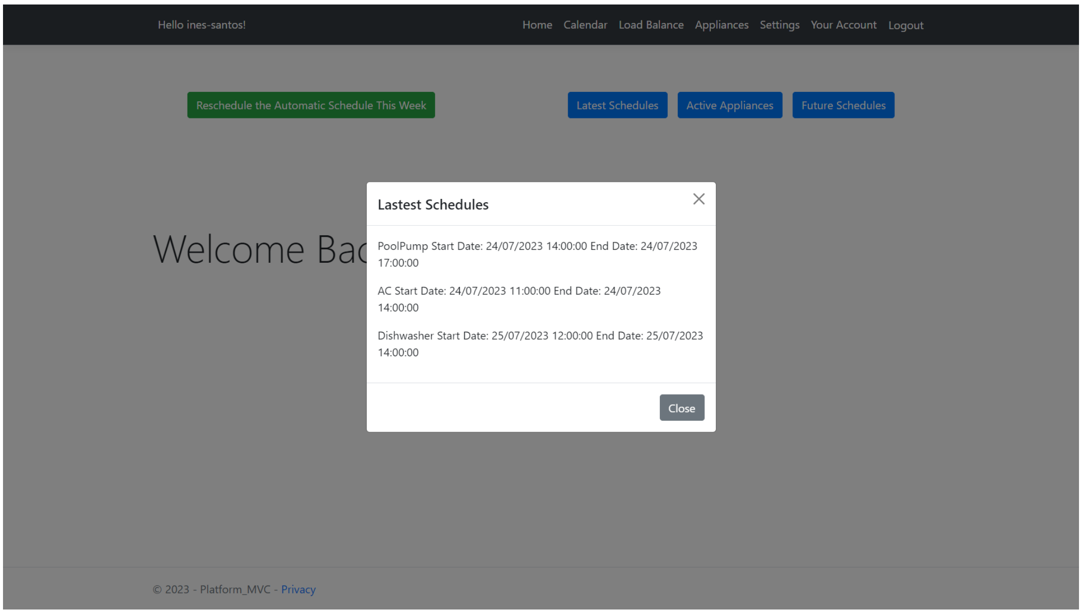


Appendix Figure 22 - *Weekly Balance* page

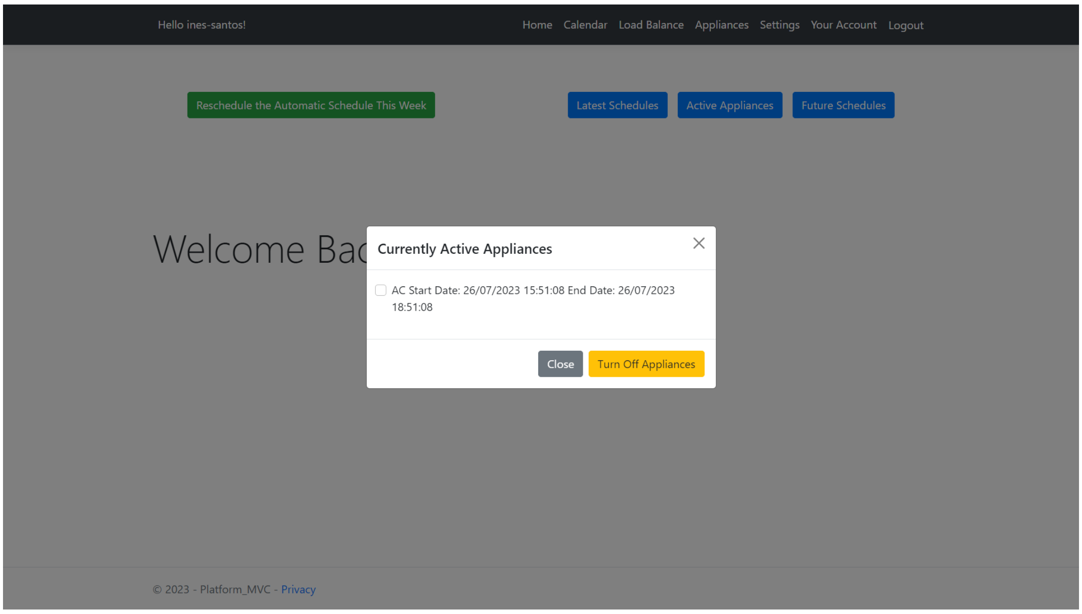


Appendix Figure 23 - *Annual Balance* page

Finally, as mentioned in Chapter 5.2, the user can also access the list of lastest schedules, and current active appliances, respesented in Appendix Figures 24 and 25.



Appendix Figure 24 - *Lastest Schedules* modal



Appendix Figure 25 - *Current Active Appliances* modal

2023 INÊS SANTOS OPTIMIZATION OF SOLAR ENERGY USING A MONITORING SYSTEM TO MANAGE THE USE OF HOME APPLIANCES

NOVA SCHOOL OF SCIENCE & TECHNOLOGY