



TÉCNICO
LISBOA

Power Share



AI Energy Sensor Interface for Google Home/Assistant

Francisco Santos Fialho

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Duarte Nuno Jardim Nunes
Dr. Sabrina Scuri

Examination Committee

Chairperson: Prof. António Manuel Ferreira Rito da Silva
Supervisor: Prof. Duarte Nuno Jardim Nunes
Member of the Committee: Dr. Amâncio Lucas de Sousa Pereira

November 2022

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

I would like to thank my dissertation supervisors, Duarte Nuno Jardim Nunes and Sabrina Scuri for the guidance and support that made this thesis possible.

My grateful thanks are also extended to the SMILE project team from M-ITI for their contribution to this project, especially Eng. Wilson Santos.

To my parents who always supported and encouraged me to do my best throughout these five years. Without their support, this thesis would not be possible. I am deeply grateful for believing in me and helping me through a really hard time in my life. To my sister Silvana for always being available to listen to me, distract me, and make me laugh.

Thank you to my family members who also showed me their support. Thank you for always being by my side.

Last but surely not least, to all my friends and colleagues, especially my closest friend group in Lisbon and my hometown.

To Ana, Bruna, Margarida, Maria, Laura, and Sara for always showing their support and celebrating my victories. For all the random plans that allowed me to distract and relax during this time.

To Afonso, Carolina, Inês, Tiago and Margarida who made part of my academic life, for pushing me and believing in me even when I didn't. Thank you, without your support couldn't make it through all this.

I want to especially thank Carolina, Bruna, and Margarida for also helping me through a hard time in my life. Without you, I wouldn't be where I am.

I want to particularly thank Carolina who was always by my side since day one five years ago. Thank you for helping and supporting me and for always being available to listen to me.

Finally, I want to thank all the participants in this study.

To each and every one of you – Thank you.

Abstract

The climate crisis is pushing the requirements for more efficient energy management systems and increasing the introduction of renewable energies. These energy management systems are able to control the consumption and production of distributed households. For instance, some of these systems can measure the energy produced by solar panels, as well as the energy consumed by the multiple appliances used on a daily basis providing feedback and guidance. These systems are usually composed of smart meters, plug-level sensors, and other components that communicate with each other to provide reliable and real-time information about household energy. The project described here builds and extends on a previous system to control and provide feedback on a house's energy consumption and production, a small factory, and a restaurant. Here we improve on the previous system by introducing a scenario of a small-industry, and restaurant setting while also exploring how natural language assistants can improve the interaction with such systems. Given that, this project will integrate Google Assistant to deliver, through a natural language interface, all the information that the user needs about the energy consumption and production in their place. This solution will then be evaluated by a set of families in order to simulate the day to day life of a family, as well as in a restaurant and factory environment, where it will be evaluated qualitatively in terms of its performance and efficiency.

Keywords

Energy Management System, Google Assistant, Artificial intelligence, IoT, NILM.

Resumo

A crise climática está a exigir sistemas de gestão de energia mais eficientes e a aumentar a introdução de energias renováveis. Estes sistemas de gestão de energia são capazes de controlar o consumo e a produção de energia numa casa. São capazes de medir a energia produzida pelos painéis solares, por exemplo, bem como a energia consumida pelos vários dispositivos que são utilizados no dia a dia, fornecendo ainda *feedback*. Estes sistemas são geralmente compostos por *smart meters*, sensores *plug-level*, entre outros componentes, que comunicam entre si para fornecer informação fidedigna e em tempo real sobre a energia de uma casa. O projeto descrito aqui tem por base e vai estender um sistema anterior, para controlar e fornecer feedback sobre o consumo e produção de energia de uma casa, de uma pequena fábrica e de um restaurante. Pretendemos melhorar o sistema anterior, introduzindo um cenário de um ambiente de uma pequena indústria e restaurante e, ao mesmo tempo, exploramos como os assistentes de linguagem natural podem melhorar a interação com esse sistema. Tendo isso em conta, este projeto integrará o Google Assistant para apresentar, através de uma interface de linguagem natural, todas as informações de que o utilizador necessita sobre o consumo e produção de energia na sua casa. Esta solução será depois avaliada por um conjunto de famílias por forma a simular o dia a dia de uma família, bem como num ambiente de restauração e industrial, onde será avaliada qualitativamente ao nível do seu desempenho e eficiência.

Palavras Chave

Sistema de Gestão de Energia, Google Assistant, Inteligência Artificial, IoT, NILM.

Contents

1	Introduction	1
1.1	Energy Monitoring	3
1.2	Smart Metering Market	3
1.3	Internet of Things	4
1.4	Home Assistants	5
1.5	Objectives	5
1.6	Proposed Approach	6
1.7	Organization of the Document	6
2	Related work and Theoretical Background	7
2.1	Power Share Project	9
	A – Version 1	9
	B – Version 2	10
	C – Other Versions	12
2.2	Blockchain in Energy Trading Platforms	14
2.3	Internet of Things and Virtual Assistants	15
3	System design and Implementation	19
3.1	Introduction	21
	3.1.1 Smart Meter Solution	21
	A – Emporia's smart meter	22
	B – Sense's smart meter	22
	C – Snappee's smart meter	23
	3.1.2 Final smart meter solution	23
3.2	Design and User Commands	25
3.3	Google Assistant	33
	3.3.1 Google Assistant's process	33
	A – Google Assistant's detailed process.	34
3.4	Implementation	36

3.4.1	Smart Meters	36
3.4.2	Power Share Application	38
3.4.2.A	Google Actions Application	38
3.4.2.B	Web Application	40
3.4.2.C	Server	40
4	Evaluation	43
4.1	Evaluation Process	45
4.1.1	Qualitative Assessment	45
4.1.2	Quantitative Assessment	46
4.1.3	Process	46
4.1.4	Results	46
4.1.4.A	Factory with Domestic	46
4.1.4.B	Restaurant	49
4.1.4.C	Domestic	50
5	Conclusion	53
5.1	Conclusions	55
5.2	Limitations	56
5.3	Future Work	57
A	Power Share Structure	63

List of Figures

2.1	Power Share Android app home screen tabs	10
2.2	Power Share Android app	10
2.3	New version of the Power Share Android app	11
2.4	Power Share Hyperledger schema	11
2.5	Power Share interface improvements	12
2.6	Power Share Gamified version	13
2.7	Energy trading schema	14
2.8	Smart home automation schema	16
2.9	Circuit schema	17
2.10	Google Assistant Application	17
2.11	End-user side schema	18
2.12	Cloud side schema	18
3.1	Carlo Gavazzi EM340 triphasic meter	23
3.2	Strato Pi [29]	24
3.3	Circuit Diagram (using resources from flaticon.com)	24
3.4	Power Share Illustration (using resources from storyset.com)	25
3.5	Home page (with factory environment)	27
3.6	Complete Dashboard (with energy production)	27
3.7	Latest Dashboard (with energy production)	27
3.8	House Daily Dashboard (with energy production)	28
3.9	Email about House Daily Dashboard (with energy production)	29
3.10	House Weekly Dashboard (without energy production)	29
3.11	House Monthly Dashboard (without energy production)	29
3.12	Factory Daily Dashboard (without energy production)	30
3.13	House Energy Production Prediction Dashboard (for facilities without energy production)	30
3.14	Email about energy production	31

3.15 Meals Dashboard (with energy production)	32
3.16 Help Page (with factory environment)	32
3.17 User commands [32]	34
3.18 Conversation process [33]	34
3.19 Scene execution lifecycle [35]	35
3.20 Intent matching scenario [37]	36
3.21 Smart meter solution in the electrical panel	37
3.22 Meals Dashboard Notification	39
4.1 Interface ranking in Pre-Test	47
4.2 Factory Consumption Differences	48
4.3 Interface ranking in Restaurant	50
4.4 Interface ranking in Houses in Madeira	51
A.1 Power Share Structure	64

List of Tables

3.1 Pros and Cons of Smart Meters	22
---------------------------------------------	----

Acronyms

API	Application Program Interface
EF	Eco-Feedback
ET	Energy Trading
ETMS	Energy Trading Management System
EU	European Union
EV	Electric Vehicle
HCI	Human Computer Interaction
IAM	Individual Appliance Monitor
IGMS	Infrastructure and Geophysical Monitoring Systems
IoT	Internet of Things
NILM	Non-Intrusive Load Monitoring
NLP	Natural Language Processing
NLU	Natural Language Understanding
P2P	Peer-to-Peer
PV	Photovoltaic
RES	Renewable Energy Sources
SAF	Set-and-Forget
SMILE	SMart IsLand Energy systems
TMY	Typical Meteorological Year
TPC	Transiently Powered Computers
UI	User Interface
UX	User eXperience

1

Introduction

Contents

1.1 Energy Monitoring	3
1.2 Smart Metering Market	3
1.3 Internet of Things	4
1.4 Home Assistants	5
1.5 Objectives	5
1.6 Proposed Approach	6
1.7 Organization of the Document	6

The climate crisis and the ways to face it are a very debated subject nowadays. Are one of the biggest priorities of every country, and many solutions have arisen from many areas, such as engineering with systems that aim to increase energy consumption and production efficiency [1].

Countries, namely Australia, and Canada, among others, implemented regulatory mechanisms for the parties involved to meet the energy-saving targets [2]. These targets and rules make significant changes in the energy market, from implementing energy certifications on buildings, regular maintenance of the air conditioning and its certification for energy performance, to even the creation of financial incentives for energy efficiency in buildings [2]. On a big scale, smart cities, "high-tech intensive and advanced city that connects people, information, and city elements using new technologies in order to create a sustainable, greener city" [3], are implementing complex systems using new technologies into their smart buildings in order to measure and optimize the energy consumption and production.

These systems are part of smart-city projects that tend to incorporate sensors in order to get information from the electrical infrastructure. These sensors read and communicate through integrated apps, thus web dashboards, or even mobile applications to make this information available to the user.

1.1 Energy Monitoring

There are many ways to retrieve information about energy consumption and production. The two most mentioned ones are the Individual Appliance Monitor (IAM), and the Non-Intrusive Load Monitoring (NILM) [4].

IAM is an often-used technique to get information from appliances and consists of having, as the name says, individual monitors for each appliance.

The NILM technique is now beginning to be more explored, and its measurement algorithms are being improved, as stated in some research [4, 5]. NILM is usually located in a central place and can provide information in real-time about various individual appliances, such as lights and machines, among others, using voltage measurements. Thus, this last approach has many advantages, for instance, with only one device is possible to get information from various electric devices about energy consumption and like so, and it helps people decide the matter to reduce energy costs.

1.2 Smart Metering Market

The urge to integrate smart meters is becoming very common as more people install solar panels in their homes. In this regard, some smart meters in the market do the job, by monitoring the energy consumption production. The majority of the meters solutions available provide an integrated system composed of the smart meter itself and an application, whether a mobile application or even a web

platform, where the user can check their energy usage.

As detailed in the smart meter solution section 3.1.1, just a few meters make their Application Program Interface (API) available to developers create their solutions, since most of them already have their own integrated system. The best smart meters solutions available in the market are from the following companies: Sense [6], Emporia [7], and Smappee [8].

All of them can measure energy consumption and production, but they stand out from each other by their unique characteristics, as pointed out in the table 3.1. The most sophisticated and innovative of these three meters is the Sense smart meter system [6]. Sense provides a complete system that integrates not just the smart meter but also many ways to visualize the information from their devices. Besides the ability to check information through their mobile application and even through their web app, Sense also integrates stand-out services, such as Google Assistant, Alexa from Amazon, and even Philips Hue, among others, that facilitate the user's day to day life of controlling their devices around the house. These advanced systems integrates an Eco-Feedback (EF) functionality that "provides feedback on individual or group behaviors with a goal of reducing environmental impact" [9].

1.3 Internet of Things

The urge to create ways to be more conscious of the environment is increasing in parallel with the Internet of Things (IoT) integration. "It is estimated that the numbers of IoT users will improve massively where the economy grows significantly" [10].

The IoT stands for Internet of Things. These "things" or devices are designed to be convenient, easy to use, and seamlessly integrated into people's lives. Since old technology is decreasing in price without decrease its potential, it is becoming easier and more common to implement IoT devices. This technology makes day-to-day life easier, and its use is practical since it requires almost no human interaction [11] and some cases no human interaction at all, which is the case of the automation between IoT devices.

The IoT devices used in such projects are categorized as Set-and-Forget (SAF) devices in the IoT taxonomy [12] and are characterized by a battery life of several years, so they do not require much user intervention. The IoT taxonomy is divided into four categories. The first category includes wearable devices. The next one, is composed of the so-called SAF, followed by the Infrastructure and Geophysical Monitoring Systems (IGMS). The last category includes Transiently Powered Computers (TPC), such as smartcards.

1.4 Home Assistants

With the evolution of technology, in the virtual assistant field, many home assistant solutions are being developed to bring people closer to this new technology, creating products that can be integrated into their homes. Home assistants have a large number of functionalities that allow users to automate their homes using smart devices, ultimately helping the user in their daily life. The most famous home assistants on the market are from the top three tech companies, Amazon, Google, and Apple. Each of them has its own virtual assistant, Alexa, Google Assistant, and Siri, respectively.

The home assistant market is continuously growing and is expected that this year, 2022, "123.5 million US adults will use voice assistants at least once per month, and that base will continue to increase over the next few years" [13]. This represents a great opportunity for systems to grow and extend their functionalities to home assistants.

Different projects emerged from multiple areas and explored the benefits of virtual assistants in fulfilling their objectives. In section 2.3 are described examples of project that implemented IoT and virtual assistants.

1.5 Objectives

One of the main problems stated in studies [14] is the availability of the user in their schedule to check data and its accessibility.

To tackle this problem is important to take into account the different ways to conveniently present information to the user and also use new technologies that are becoming part of day to day life of people. As mentioned above, IoT is nowadays a popular concept that is being applied to objects to automate tasks and make people's lives easier. When combined with virtual assistants, another popular technology, it makes every process much more smoothly, seamlessly and increases the interactivity with the users' houses.

This project aims to strengthen people's awareness of renewable energy solutions as well as the energy usage in their households by achieving the following goals:

- Develop a system composed of a smart metering solution and a Google Assistant application to check information about energy consumption and production in facilities;
- Test the Google Assistant application in domestic, small manufacturing, and restaurant scenarios and understand whether this type of application is more engaging.

With this approach is expected to fulfill the objectives of being easy to use, convenient, and also to be incorporated into the user's daily basis.

1.6 Proposed Approach

Given the work objectives established, the proposed solution encompasses the development of an energy management system with interfaces and voice interactions for Google Assistant. For this purpose, will be developed a Google Assistant application and a web application to provide custom interfaces to the user.

The solution will be composed of a Google Action application, a Web application, and a Nodejs server. The Google Action application will be responsible for supporting voice commands and retrieving the Web application interfaces and will be the application with which the user will interact. As for the Web application it will integrate all the custom interfaces to show to the user and will be directly connected to the Google Assistant application and the server application. Finally, the server application will be responsible for the requests needed by the functionalities of the Web app and the connection to the SMart IsLand Energy systems (SMILE) API.

One of the test environments of this project has the particularity of composing not just a household but also a small factory. This factory is focused on producing handbags, among other fashion products. In this way it has many machines that require a large power supply, supported by a triphasic power setup. It counts with more than eighteen machines, three fans, and a large amount of led lights. As for the house, it has a triphasic power setup, as well.

The evaluation procedure of this solution encompasses a set of families that will test the system by incorporating it into their daily life. In addition, the solution will be tested in a restaurant, as well. The results of these tests will help answer some questions raised relating virtual assistants, and this type of application, among others.

1.7 Organization of the Document

The document is organized as follows: Chapter 1 introduces to the project, Chapter 2 presents the related work and the state of art with projects using the latest technology, such as blockchain and IoT among others. The Chapter 3 described the proposed solution with all the technologies used, as well as all the steps to create the final solution. The strategy and evaluation of this project are presented in Chapter 4. Finally the chapter 5 states the conclusions taken from the project as well as, the future work.

2

Related work and Theoretical Background

Contents

2.1 Power Share Project	9
2.2 Blockchain in Energy Trading Platforms	14
2.3 Internet of Things and Virtual Assistants	15

In this section, is presented the related work on blockchain as well as IoT and previous projects. The related work is divided in subsections, namely (a) 2.1 Power Share Project, (b) 2.2 Blockchain in Energy Trading Platforms, (c) 2.3 IoT and Virtual Assistants.

2.1 Power Share Project

The previous projects' solutions, like most solutions in this field, were built with blockchain, a new technology with the objective to decentralize the process of transactions as we know [1].

A – Version 1 The first project aimed to test business models for the SMILE project (<https://www.h2020smile.eu>). The SMILE project involves three islands, one in Portugal, one in Denmark, and one in the United Kingdom. The main goal is to test technological solutions related to renewable energy and further replicate them for other parts of Europe. The SMILE consortium is composed of 19 partners from 6 European Union (EU) countries.

The proposed solution in this project was to inform people about their energy consumption and allow prosumers - "i.e., households that function as both energy producers and consumers" [15] - to exchange their surplus energy with their neighborhood.

In order to achieve this goal, was created a mobile application (Android) for Energy Trading (ET) using blockchain as a payment method and the Power Share Energy Trading Management System (ETMS) to manage users' accounts. With this application was intended to raise awareness about energy usage, allow prosumers to sell their surplus energy and to engage the share of Renewable Energy Sources (RES), by providing EF and tools for ET [1]. The payment process combines blockchain as the payment method, with IOTA technology [16].

The main advantages of integrating this technology were the solution of scalability problems and the elimination of transaction fees.

This application is composed of many different screens and functionalities that allow the user to check their energy consumption and production, make transactions and configure the criteria to buy and sell energy. The functionalities are divided into four interfaces, home, historical, transactions and ranking, as Figure 2.2 shows. On the home page is presented the current energy consumption and production and a menu bar with three different options. This menu bar, represented in Figure 2.1, shows the battery level for the amount of energy stored, the current transactions, and a section with EF for the share of RES.



Figure 2.1: Power Share Android app home screen tabs

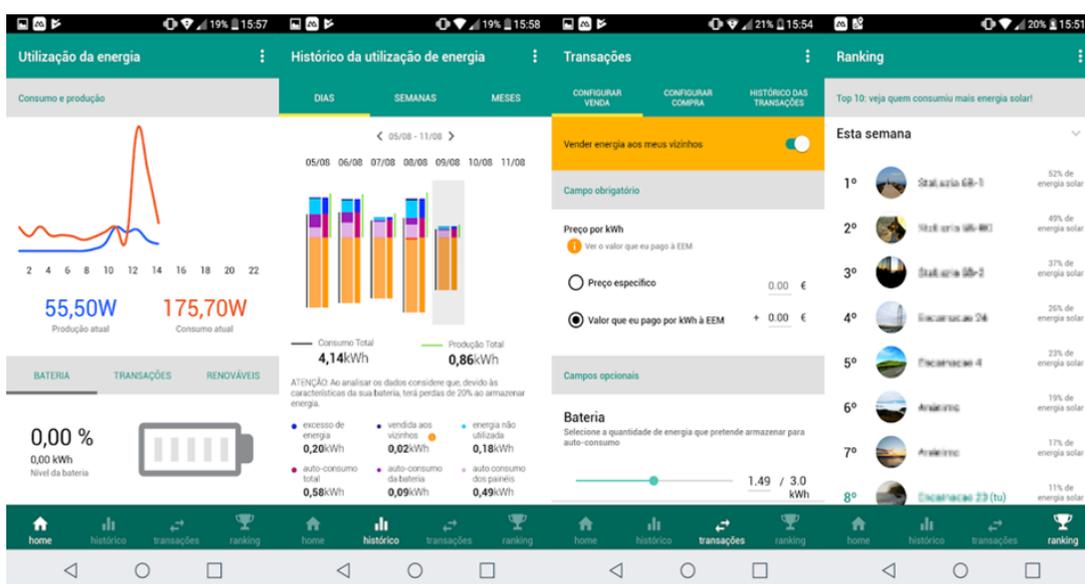


Figure 2.2: Power Share Android app

The historical page shows all the past energy use, with detailed information. On the transaction page the user sees the energy purchase and sale settings and their historical transactions. The user can change the battery level, price parameters and choose whether they want to sell/buy energy to/from the neighbors. In addition, the author found it interesting to create a page where users could "compete" between them to engage them to use the app. Finally, the last page presents the neighbor's ranking based on their consumption of renewable energies.

B – Version 2 The latter project solution is an iteration of the previous one. It aimed to create a new version of the mobile app and even improve the project performance by integrating a new blockchain system [17].

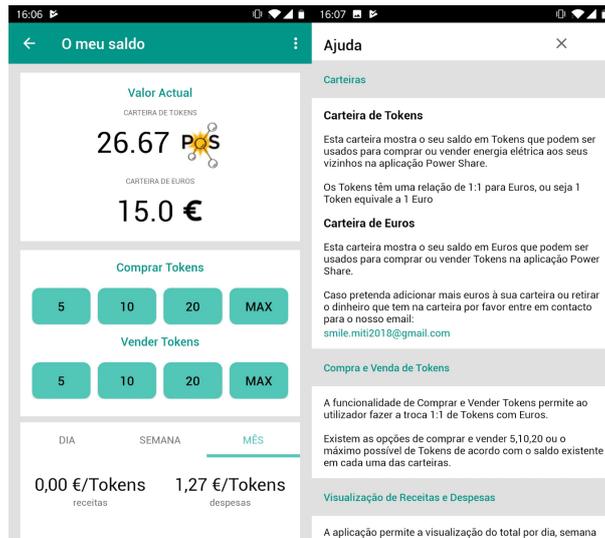


Figure 2.3: New version of the Power Share Android app

This improvement was directed towards the integration of a new technology, the Hyperledger Fabric Blockchain Server. This network allows prosumers, consumers, electric vehicles, and utility companies to exchange energy [17]. The new version benefits from the main characteristics of Hyperledger by using a private blockchain for confidential transactions and smart contracts to run code in blockchain to automate the model, giving stability to the whole transactions model.

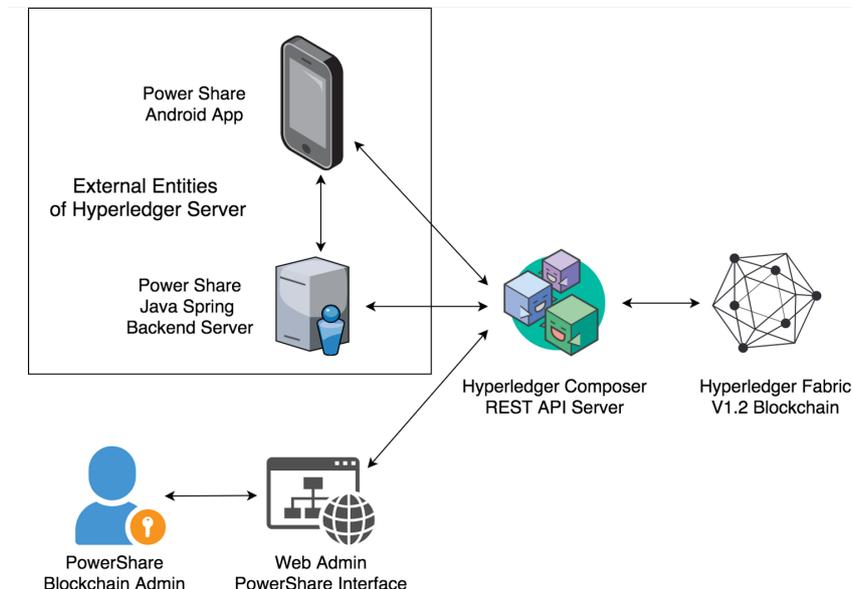


Figure 2.4: Power Share Hyperledger schema

The system schema is represented in Figure 2.4. The system is composed of an Android app and

a Spring server that represents the external entities that communicate with the REST API. It has an Angular web app, in which the company's administrator manages the participants, such as Residents (energy prosumers and consumers), Electric Vehicle (EV)s, among others.

In conclusion, the transition from IOTA to Hyperledger Fabric was beneficial as it gives even more stability and the transactions run much more smoothly since it improved its performance. This new version of the Power Share project, as stated before, focused on increasing the performance of the transactions and the whole process while also trying to improve user engagement.

C – Other Versions Besides these projects, other studies were made regarding the Power Share project, namely a study that aimed to develop and test Power Share in a Human Computer Interaction (HCI) perspective [14] and a study that created a gamified version of this previous one [15].

The Power Share Peer-to-Peer (P2P) Energy trading project [14] started by choosing the technology to use in the platform, specifically for the energy trading system. The technology frequently referenced as revolutionary and a good choice for this purpose is blockchain. However, it also has some intriguing aspects that made the authors debate. One of those aspects is the trust given by the users since people tend to trust an entity to perform transactions. So, compared to a decentralized system, where there is no third-party handler involved, they choose not to trust. This was a very interesting aspect since the study is focused on the HCI perspective, besides the implementation of the platform.

It is possible to see the improvements made in the interfaces compared to the previous solution 2.2 in Figure 2.5.

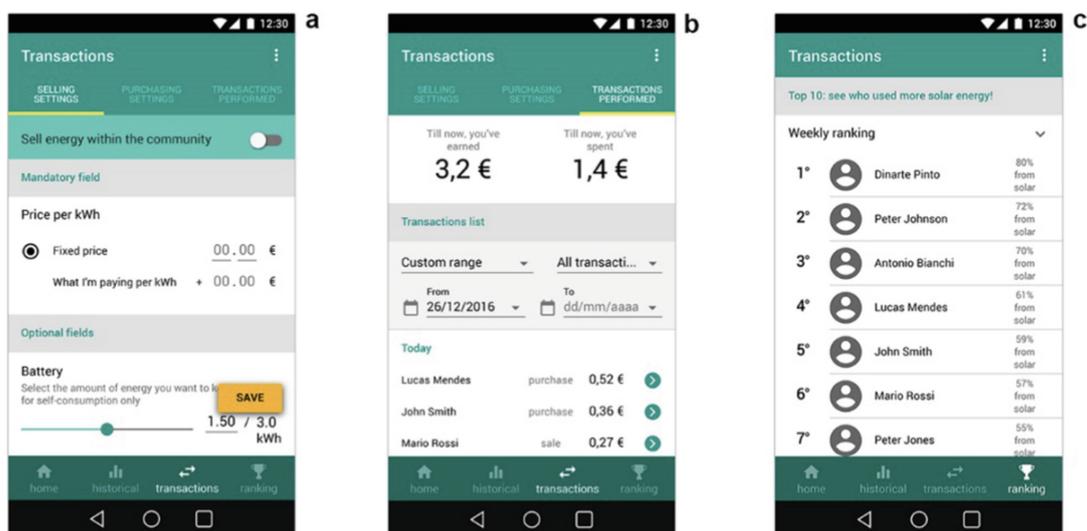


Figure 2.5: Power Share interface improvements

Some important results noted throughout their investigation were the impact of the economic part of this matter, the transparency of the information about the billing, the sense of community and social

comparison that led to the next iteration [15], and the trust that the majority of the users gave in this decentralized system.

The authors concluded that users were becoming more aware of their energy consumption and that it also created an interest in them. A particular feature that the users were looking forward to was the ranking, where they can see their position compared to the rest of the neighbors regarding "effective in fostering sustainable behaviors" [14]. The users reported that they checked the ranking regularly, and it motivated them to compete with each other.

This was an important aspect for the authors since the users felt motivated to be at the top of the ranking and also felt engaged to use the application even more.

Thus, this brings us to the next iteration of this project: Power Share 2.0 [15], the gamified version. This project aimed to explore further the game strand of the previous solution by creating challenges for users to surpass. The three main challenges consist of reducing energy consumption within the community, keeping the consumption below a threshold, and finally increasing the number of consumers of RES. Whenever the community completes the challenges, they unlock a new level, and the app also helps users by giving them tips about ways to be energy efficient.

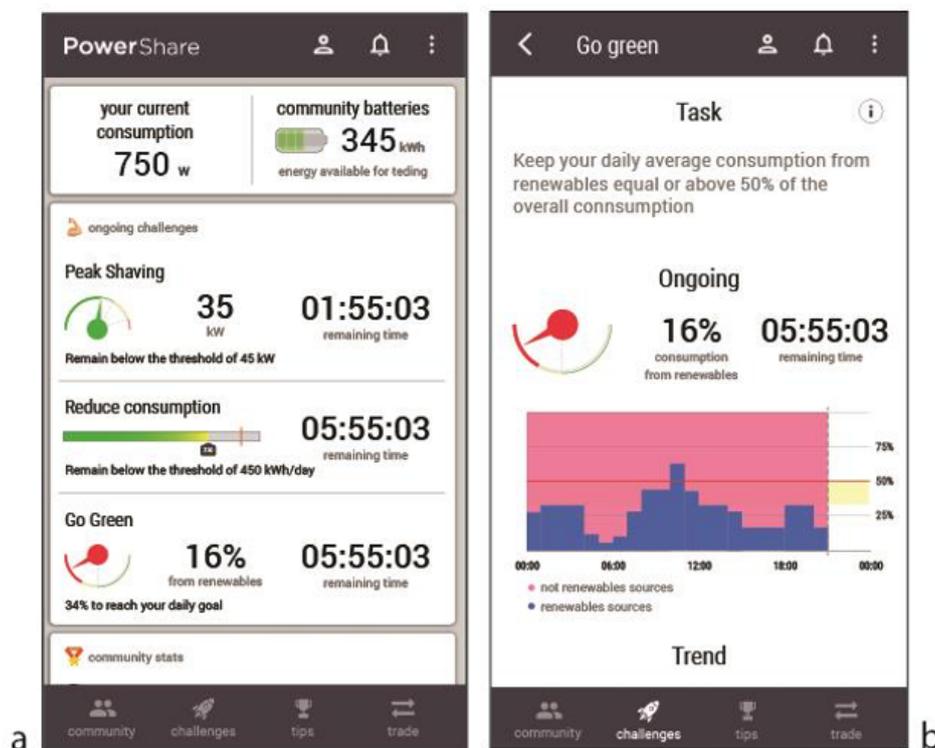


Figure 2.6: Power Share Gamified version

As a result of this study, they concluded that the design experience was very effective once the users were more engaged and felt that they were united in favor of a common objective. They also noted that

users were much more interested in their energy consumption and production, and it gave them a new perspective on the environment and renewable energy matter.

2.2 Blockchain in Energy Trading Platforms

Blockchain is now becoming a prestigious technology and evolving substantially lately since people are becoming more aware of its benefits when applied in projects such as energy management systems.

Blockchain is a distributed and immutable ledger that helps in transactions by easing its process, and tracking assets, for instance cars, copyrights, and many other valuable possessions [18]. This technology offers security, for example, in payments, it automates processes by using smart contracts and also ensures data integrity, among other features.

Given this, energy management systems, such as the Power Share project [1], described in the previous section 2.1, benefit from using blockchain. In addition to this project, blockchain technology is a success in many sectors. In the financial industry the blockchain is used in crypto services platforms and in the logistic sector in the digital ledger of shipments. The health care sector uses smart contracts platforms, such as BurstIQ company [19], to transfer sensitive medical information from doctor to patient and vice-versa.

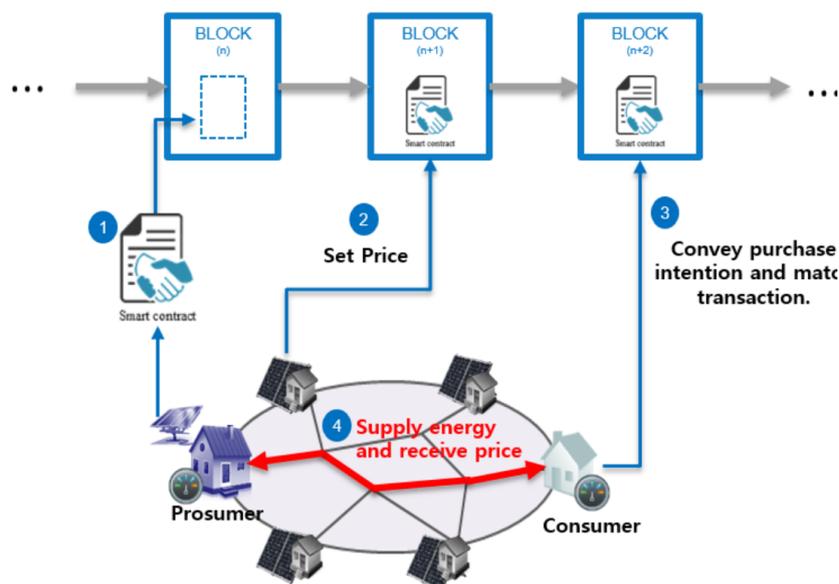


Figure 2.7: Energy trading schema

Kang et al. [20] also agreed that the use of blockchain was a good choice, taking into account its properties that help them to create an energy trading platform combined with IoT. Their approach of combining blockchain as part of the trading model with the IoT as part of the communication system,

was based on retrieving information from the IoT devices in houses, for instance, light bulbs, and using Ethereum's smart contracts to manage the energy transactions between consumers and prosumers.

Their energy trading diagram shows the process behind it, with stakeholders as prosumers, the neighbors, and the consumer, as shown in Figure 2.7. With this implementation, they were able to create, what is commonly named, a smart grid, which consists of having a "widely distributed automated energy network" [20] between consumers' houses.

Following a similar path, Baig et al. [11] also found it interesting to apply blockchain in their peer-to-peer energy trading platform. However, they have a slightly different approach with the IoT devices. The project was also built using Ethereum's smart contracts to manage and automate the energy trading between consumers. The IoT part was composed of four hardware devices, such as a power supply, a sensor to get current drawn by the load, an Arduino board, and a led. Using the Node-RED flow, a visual programming platform, they managed to program the Arduino to whenever a transaction was initiated the led would light up and then turn off as the transaction was over.

From this project, they have concluded that they were able to develop a platform that provides helpful functionalities, namely energy transferring and money transferring between users [11]. They also pointed out that it stands out from other similar platforms by being user-friendly, efficient, and profitable.

2.3 Internet of Things and Virtual Assistants

Recently IoT is getting introduced into platforms that use blockchain, like the ones stated before. In the "IOT Based Smart Energy Meter for Efficient Energy Utilization in Smart Grid" [21] study, the authors had an interesting and a bit complex approach to energy-efficient systems for a proposed system. Considering all the benefits of IoT, the proposed system consists of a smart energy meter that controls energy consumption and helps detect power theft. The components that composed this solution were a Wifi module to manage the operations through the system, an OLED display that shows the operations' information and energy consumption, an energy meter, an optocoupler, and a current sensor.

With this solution, they were able to create an energy consumption pattern with the information retrieved, detect power theft, conserved energy, and have automatic control of the energy meter.

Vishwakarma et al. [22] developed, using IoT, a home automation to efficiently control the energy around the house. The system involved an Arduino using NodeMcu, Adafruit, and IFTTT [23], a web-based assistant, and Google Assistant, as represented in the schema in Figure 2.8. The authors highlighted the evolution and importance of the new technologies in the IoT field, which now allows older people and people with some disabilities to interact with systems with the help of virtual assistants such as Google Assistant.

Keeping this in mind, they developed a prototype model with three light bulbs controlled by Google

Assistant with voice controls and a web-based service. One of the reasons why it is possible to manage the smart home via a web-based service is because voice commands may not be well processed in the presence of background noises, which is one of the disadvantages of virtual assistants using voice commands.

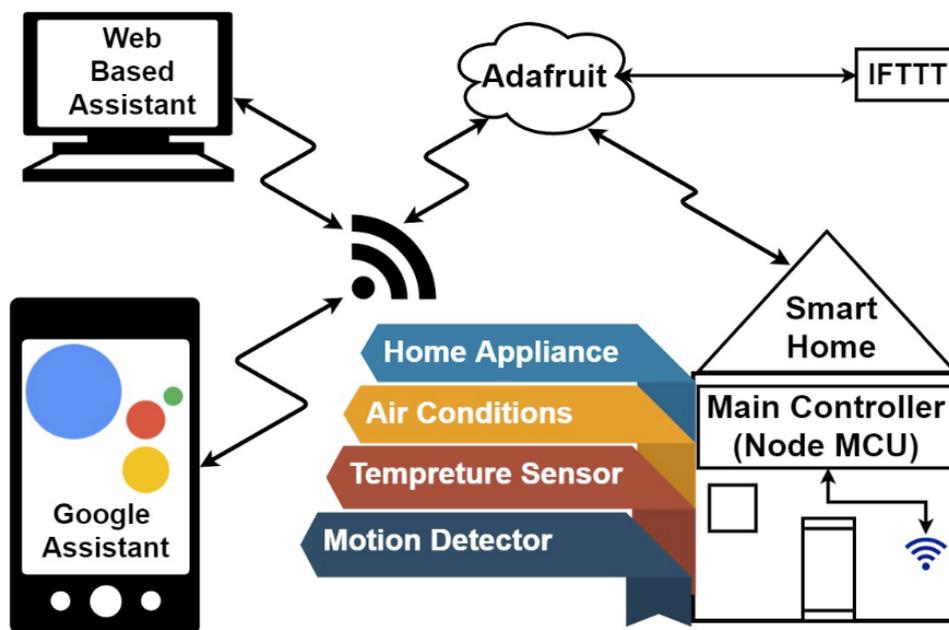


Figure 2.8: Smart home automation schema

With a similar perspective, Isyanto et al. [10] implemented a system for people with disabilities using Google’s virtual assistant as well, where they can control the entire home simply with their voice. With this system, users were able to control the fans, lights, and even the TV using Google Assistant. Figure 2.9 represents the circuit diagram the authors implemented for this solution. Google makes its virtual assistant available on many devices, from its range of Google Nest products to even smartphones, whether they are Android or IOS. This was a very interesting aspect pointed out by the authors since it covers a great number of users, not restricting IOS devices, representing 42.7% of US smartphone users [10]. As an example, Figure 2.10 shows how the system works.

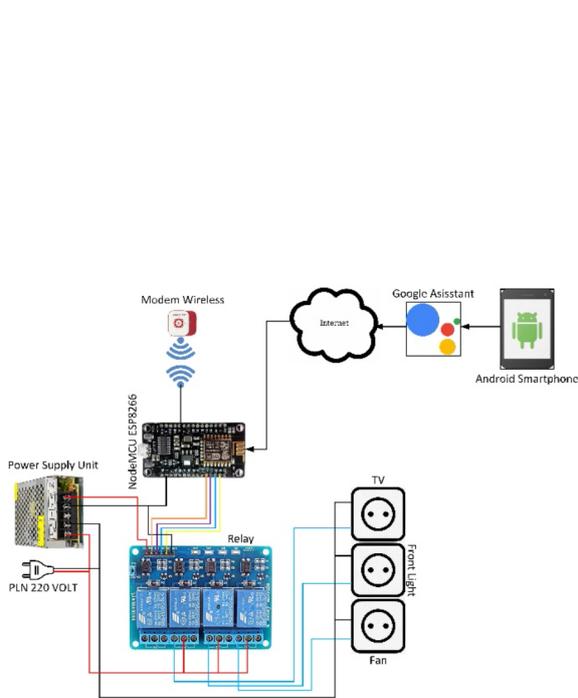


Figure 2.9: Circuit schema

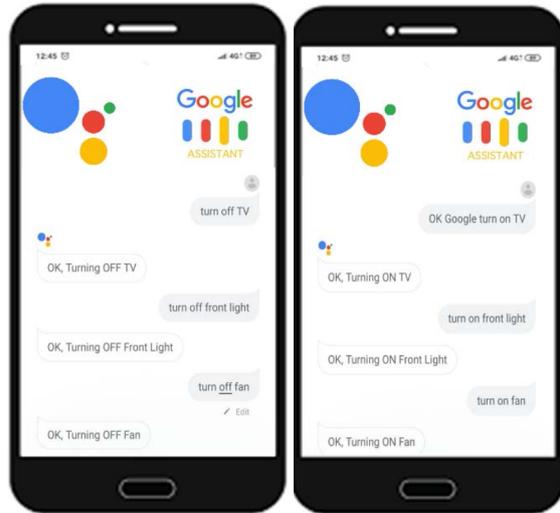


Figure 2.10: Google Assistant Application

Many projects aim to control a certain place using just voice commands. Another example of that is the project of Poongothai et al. [24], which consists of automating a laboratory by using Google Assistant to control devices, such as lights, projectors, and AC among others. Their solution was based on a raspberry pi using NodeMcu, the usage of Blynk API, "a real-time state sensing IoT platform that helps in interfacing IoT hardware and Android or an iOS device" [24], also a Heroku Server and Google Assistant. This proposed system not only controls the multiple devices in the laboratory but also monitors, continuously their energy consumption, presenting the data to the users.

A closer solution to this document's proposed approach is the one described in the "Voice-Based Monitoring And Control System of Electronic Appliance Using Dialog Flow API Via Google Assistant" paper [25]. With the resource of Google's technology, such as Google Assistant and its development platform, the authors were able to create a system based on the voice that can control and monitor electronic appliances.

Google offers with their virtual assistant an API (Dialog Flow API [26]) in which developers can create their applications with Google Assistant. Google Assistant incorporates Automated Speech Recognition and Text-to-Speech all in one, which facilitates developers' process of converting human language to information with their API.

All of these tools helped the authors develop an application and tests that controls the aquaponic. The user is, by default, greeted by Google Assistant saying "Hello, how can I help?" [25]. Then the user makes a request, for instance, "How about the temp?" and the assistant captures the voice request, communicates with the different components such as database, and responds with "Current temp is "\$temp" degree Celsius" [25]. This was one of the experiments that they did with their application with successful results.

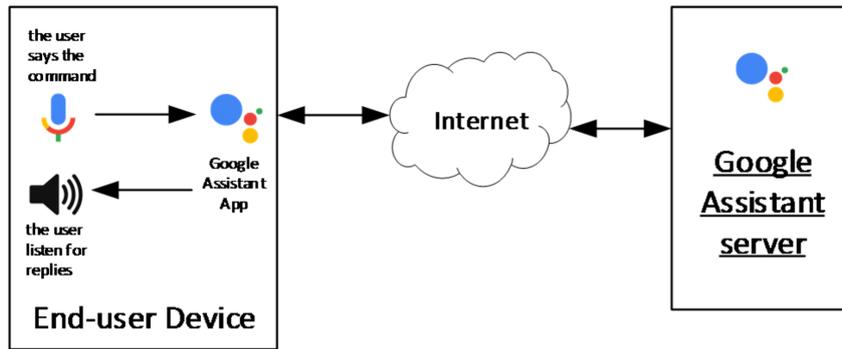


Figure 2.11: End-user side schema

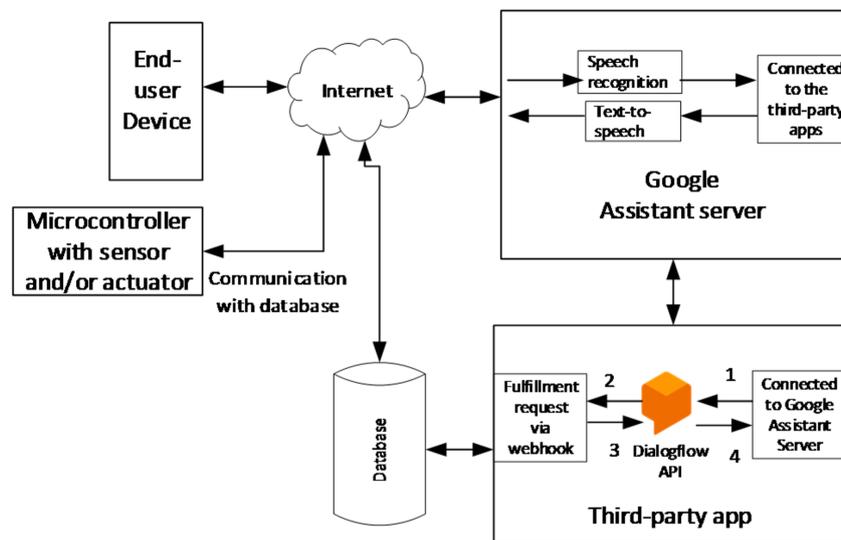


Figure 2.12: Cloud side schema

Google Assistant has many advantages, among them the authors highlighted that it is not required memory to run such applications since all the process is made on the Google Assistant server's side. They also mentioned that the user experience is much more seamless since they don't need to interact with the application in the same way it is done in other types of applications.

With their solution, they stated that "the test outcomes demonstrating that creating electronic gadget application on checking and control dependent on voice had incredible potential. The accomplishment pace of talk affirmation structures is 75% of success rate" [25], which is a very positive point to proceed with developments in this area.

3

System design and Implementation

Contents

3.1 Introduction	21
3.2 Design and User Commands	25
3.3 Google Assistant	33
3.4 Implementation	36

This section describes the process of developing this solution, as well as the pros and cons of each decision. This section has the following structure: (a) 3.1 Introduction, where is introduced the different components that compose the new Power Share app, and the decisions made to get the right solutions; (b) 3.2 Design and User Commands, where are defined the User Interface (UI) following the User eXperience (UX) guidelines and all the user commands to navigate through the app; (c) 3.3 Google Assistant, where Google Assistant is introduced and the process behind every Google Action application; (d) 3.4 Implementation, where is described the development process of each part of the Power Share app. This section is in turn, divided into four subsections, which describe the four components of this solution.

3.1 Introduction

The main goal of this project is to develop a system composed of a smart metering solution and a Google Assistant application to inform the user about their energy consumption and production in their home or facility, bringing users closer to virtual assistants while making their interaction seamless and convenient daily. And finally, test the Google Assistant application in domestic, small manufacturing and restaurant scenarios.

One of the environments of this project is very particular, as it includes a house and a factory nearby. This factory is focused on producing handbags, among other fashion products. This way, it has many machines that require a large power supply, supported by a triphasic power setup. It counts more than eighteen machines, three fans, and a large amount of led lights. As for the house, it has a triphasic power setup, as well. This environment expands the initial functionalities thought for the system. It made us think about different environments that may exist and where this system could fit. Showing information regarding energy consumption, and in some cases, about energy production, among other helpful information.

Thus, this project counts on different test environments, namely the environment previously described, houses, and even restaurants. In this way, the system is thought to handle multiple types of facilities and energy configurations.

To develop this solution, we needed to divide it into two parts, the smart meters, and the application. The first part covers the installation of smart meters and raspberry pi in the electrical panel and their configuration. The last part describes the development of the Power Share app, including the Google Assistant application and its connection to the meters.

3.1.1 Smart Meter Solution

First and foremost, it is important to understand which technologies fit best in this solution, and smart meters are a crucial part of this project. There are some smart meter solutions available in the market.

However, just a few of them do what is envisioned in this project. Those eligible ones are able not only to monitor real-time energy consumption but also to alert users about relevant events on specific devices.

The three leading smart meters with those characteristics are from Sense [6], Emporia [7] and Smappee [8]. These devices have some functionalities in common that are relevant for this project, such as monitoring in real-time the energy usage, more specifically consumption and production, and making this information available to developers to explore.

The following table, 3.1, represents the different pros and cons of each meter solution, which helped to decide what was the best solution for this particular project.

	Pros	Cons
Smappee	It has an official API	Requires too much equipment and they are expensive.
	It doesn't implement features that are proposed in this solution	Their API usage and Data Usage License has a time limit and it requires a subscription to keep it.
	It offers an online dashboard	Limited API calls (1 per hour)
Sense	It has an official API	Already does parts of the proposed solution
	The data is available at any time	It may be impossible to integrate in the European electrical panels, there are no certainties.
	It doesn't require too much equipment	
Emporia	It offers an online dashboard and a mobile app to manage the different devices	Doesn't offer an official API
	The price is reasonable and it doesn't require too much equipment	

Table 3.1: Pros and Cons of Smart Meters

After analyzing the characteristics of each solution, it was possible to conclude which is the best option since each pro and cons have different weights.

A – Emporia’s smart meter The biggest con that excludes Emporia’s smart meter is that it does not provide an official API to make data from the electrical panel available to developers. Although it has attractive features, such as web and mobile apps with real-time information, not having an API makes it impossible to work with this meter.

B – Sense’s smart meter Sense’s smart meter has many advantages, making it one of the best choices to implement in this solution. In fact, the meter is very well developed, and it explores more strands than any other mentioned, namely IoT integration with Google Assistant and Alexa from Amazon and many more services.

Given this statement, this solution already implements some functionalities meant to be developed in this work, which is not a point in favor. But the major con is that Sense is an American company, and their devices are designed for North American electrical panels, which differ in voltages from European

electrical panels. This constraint makes it impossible to install the meters.

C – Smappee’s smart meter Last but not least, the only meter left from table 3.1 is the Smappee one. The main advantage of this device is providing an official API to work with, as Sense does. It also offers an online dashboard to consult the data and does not incorporate features from the proposed solution.

The dark side of Smappee is the API and data availability. That is, the API access and the Data Usage License is limited, and even the API calls are limited to 1 per hour. Besides these restrictions, it is required too much equipment, and its price is too high.

3.1.2 Final smart meter solution

Since all previous devices had some major cons that made their implementation difficult in this project, a different approach was taken to solve this problem.

Given the previous successful experience with the Carlo Gavazzi smart meters [27], in the SMILE project [28], we chose the EM340 electricity meter (Figure 3.1) to read the energy usage from the electrical panel. This meter is specific for triphasic power setups, which is the one present in the environment described in 3.1.



Figure 3.1: Carlo Gavazzi EM340 triphasic meter

The smart meters will be integrated into a system that also includes a Strato Pi [29], that inside has a Raspberry Pi 4. The Strato Pi will be responsible for processing the information received from the smart meters, and to send it to the SMILE API.



Figure 3.2: Strato Pi [29]

The final composition of this system can be seen in the following figure.

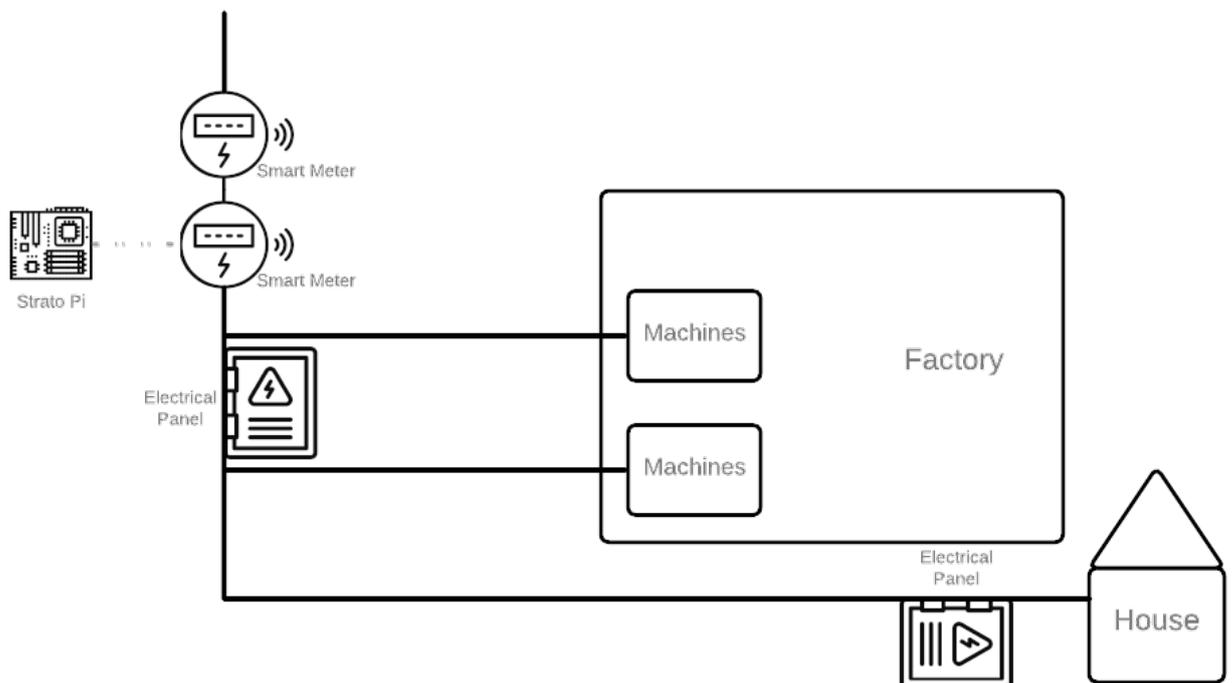


Figure 3.3: Circuit Diagram (using resources from flaticon.com)

The smart meters are placed in the triphasic energy circuit both in the source, where it will measure the factory and house's power. With the meters in place, the next device to be integrated into this circuit

is the Strato Pi. This device will collect information about energy consumption and production from the smart meters and then will make it available through an API to be fetched.

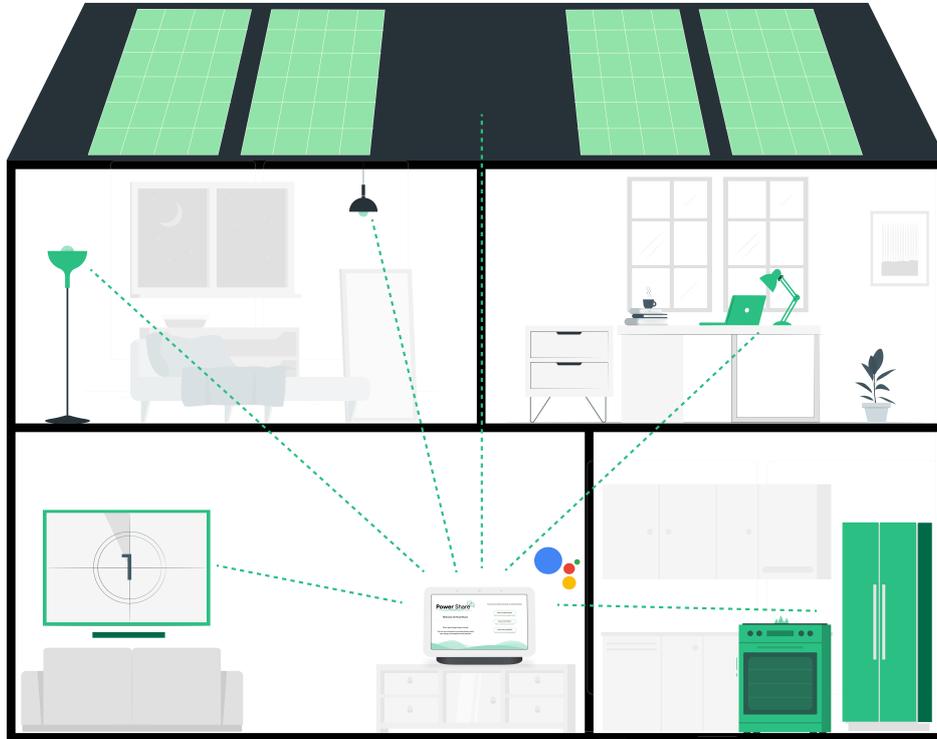


Figure 3.4: Power Share Illustration (using resources from storyset.com)

3.2 Design and User Commands

From the solution designed for this project, we started to structure the model of the application, specifying the user commands and the interfaces that are shown to the user. In this process, we took into account the user needs, the helpful information that we can take from the smart meters, the information shown on previous projects [1, 17], and also how to conveniently show it to the user through the interfaces.

With this process, we identified interfaces that were crucial to be implemented:

- **Home:** introduces the application to the user with a description. Suggests user commands to navigate through the app.

Each of the following interfaces encompasses two dashboards. That is, in each interface the user can choose to see the dashboard regarding the current or previous day/week/month, respectively.

- **Daily Dashboard:** shows information about today's or yesterday's energy consumption and production. It also presents a results card describing the peak and low values and feedback for both

production and consumption. The feedback for this dashboard is made by comparing today or yesterday with the same weekday of the previous week;

- **Weekly Dashboard:** presents this week's or last week's energy consumption and production. Similarly to the previous interface, here is presented a results card with the same type of information, but the feedback is made by comparing this week with last week and vice versa;
- **Monthly Dashboard:** provides an energy use overview throughout this month or last month. It also has a results card, but here the feedback for consumption and production is done by comparing this month with last month and vice versa.

After defining these interfaces, we conclude that to provide a summary of all that information to the user, and for them to check it quickly, we needed to create two more interfaces. These two interfaces contemplate a summary of the ones mentioned above:

- **Complete Dashboard:** provides a complete summary of the daily, weekly, and monthly dashboards;
- **Latest Dashboard:** shows an overview of the dashboard of yesterday and the dashboard of last week.

Once this process was completed, we discussed the visual aspect of the interfaces, by reviewing and establishing new UI/UX goals.

The interfaces needed to be simple, with clear information, and also appealing to the user's eyes. To create them, we needed to take into consideration two aspects, the screen size of the smart home device (Google Hub) and the fact that this type of application is intended to be driven through the user's voice and not so much through the touch, like usual smartphone apps. This last aspect sets some constraints when designing interfaces. Google has guidelines with best practices on how to design them, which emphasize: "design with voice-first in mind" and also "focus on one touchpoint at a time" [30].

The design of the UI started by collecting the screens from the previous master thesis [1, 17], filtering the most important ones, and selecting the components related to this project. From those screens, we highlighted the energy usage graph, the feedback information, and the different types of views of the energy graph, from days to months.

First and foremost, we started to create the home interface. Taking into account the best practices pointed out before, on this screen is displayed the logo, a short description of the app, and three user commands as buttons. Below these buttons are phrases of how to request it by voice and to follow the UX guideline, "adapt voice output for the screen" [30], the app suggests, through the Assistant's voice, other commands, and how to request it with voice;

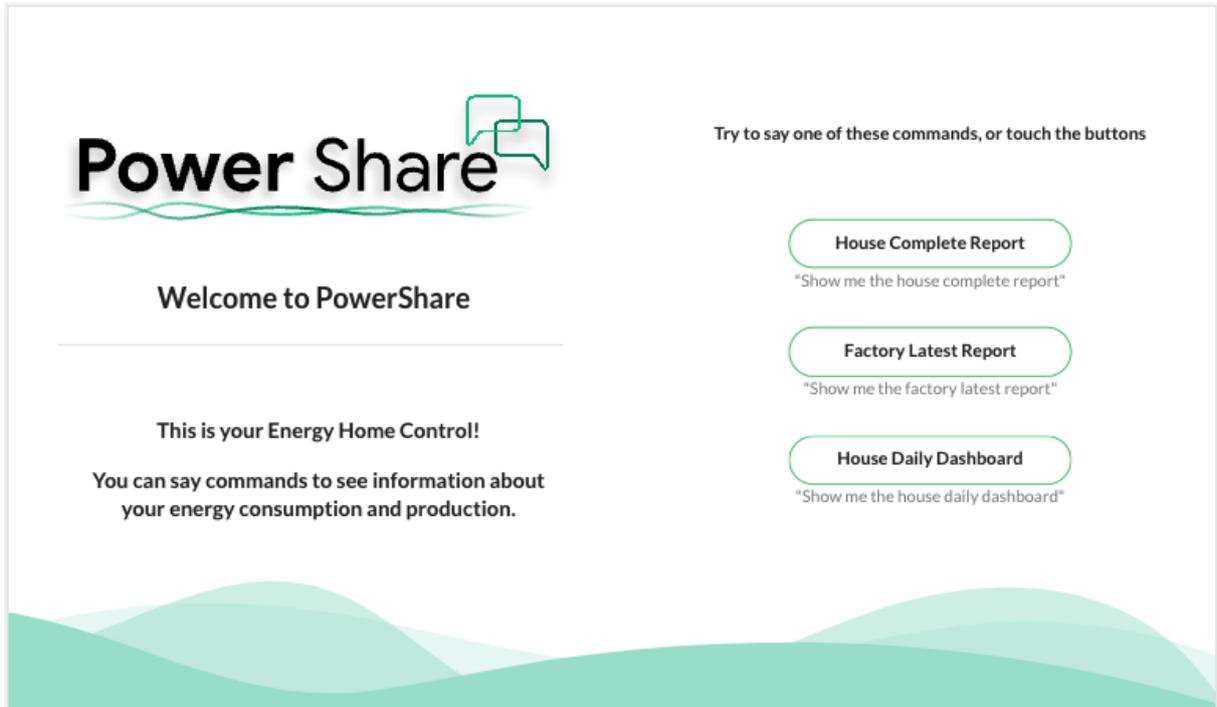


Figure 3.5: Home page (with factory environment)

After successfully creating the home interface, we moved on to the next set of interfaces, the summary interfaces, which include the complete dashboard and the latest dashboard. Since these interfaces are just a resume of the overall activity, we wanted to simplify it by showing only the necessary information.

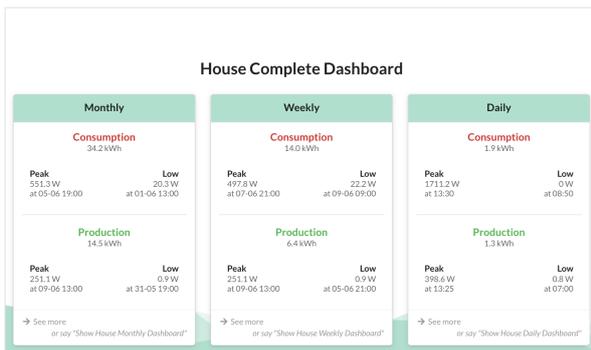


Figure 3.6: Complete Dashboard (with energy production)

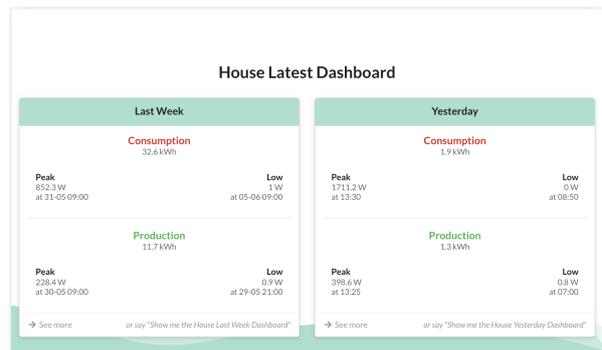


Figure 3.7: Latest Dashboard (with energy production)

As Figures 3.6 and 3.7 show the user can easily and quickly view the most important information of each detailed dashboard through these interfaces. Both screens show the peak and low values for consumption and production, when each occurred, and the overall consumption and production value in kWh. To effortlessly go directly to a detailed dashboard shown on these interfaces, we added a button

on the card that leads to that specific dashboard. Not only that, but it also presents a phrase for the user to say rather than clicking the button.

Once these were done, we started thinking about the detailed dashboards. For these types of dashboards, we wanted to show more helpful information to the user. Considering the relevant components from the previous project's interfaces, we decided to incorporate a graph representing energy consumption and production and a results card for the specific period. To provide more information and for the user to have an insight into how much they consumed and produced compared to the same weekday of the previous week, the previous week, or month (respectively, for each dashboard), we added a feedback section to the results card.

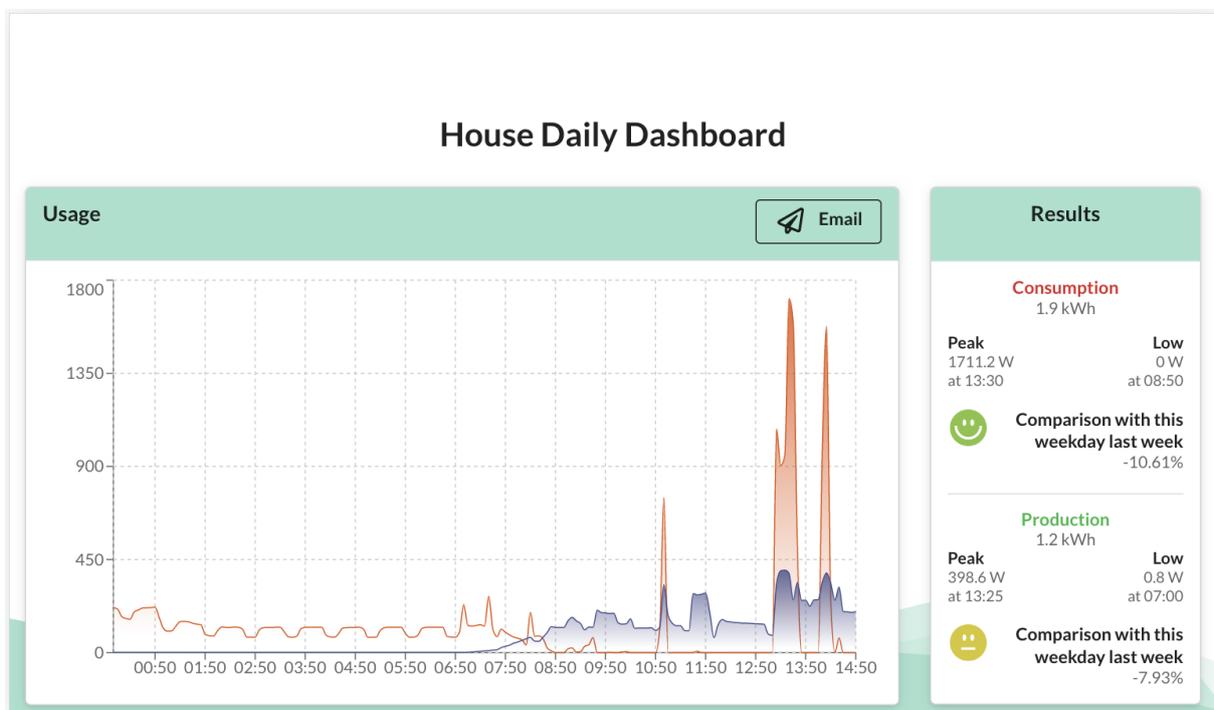


Figure 3.8: House Daily Dashboard (with energy production)

As a result of a preliminary test made with one family, specified in Section 4, a new feature, initially implemented in a different interface, was also added to these interfaces. This feature has the purpose for the user to save this information and have it at any time. With the user's consent, an email is sent with information regarding the current dashboard. This email presents not only the results of the energy usage for the specific dashboard but also links to articles on how to optimize energy consumption and a screenshot of the interface. The email sent to the users is represented in Figure 3.9.

Hello Francisco, this is Power Share!



As requested, here you have more information related to your House Daily Dashboard.

This is your energy consumption and production results

Consumption

- Overall: 1.9 kWh
- Peak: 1711.2 W at 13:30
- Low: 0 W at 08:50
- Feedback: -10.61% (Comparison with this weekday last week)

Production

- Overall: 1.2 kWh
- Peak: 398.6 W at 13:25
- Low: 0.8 W at 07:00
- Feedback: -7.93% (Comparison with this weekday last week)

If you want to know more on how to optimize your energy use, check the following links:

- [E-Redes - Good practices](#)
- [DECO Proteste - How to save money on energy and water consumption](#)

You can see your current House Daily Dashboard in the image in the attachments.

House Daily Dashboard



Figure 3.9: Email about House Daily Dashboard (with energy production)

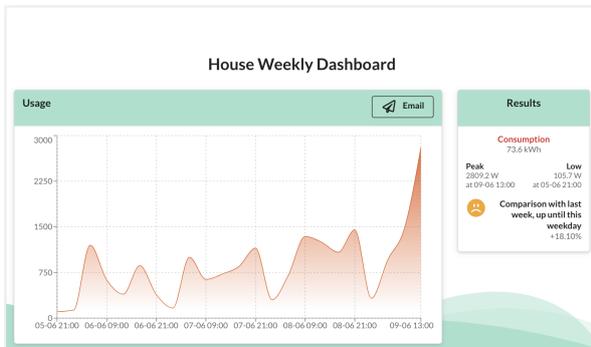


Figure 3.10: House Weekly Dashboard (without energy production)

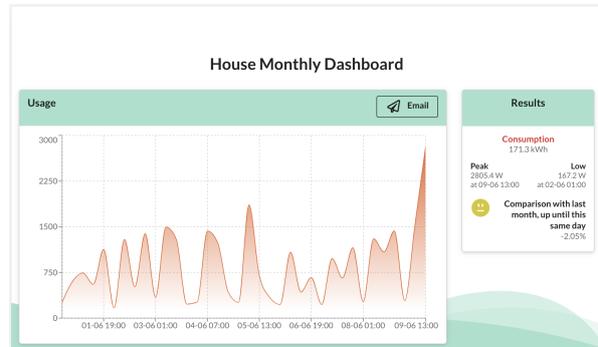


Figure 3.11: House Monthly Dashboard (without energy production)

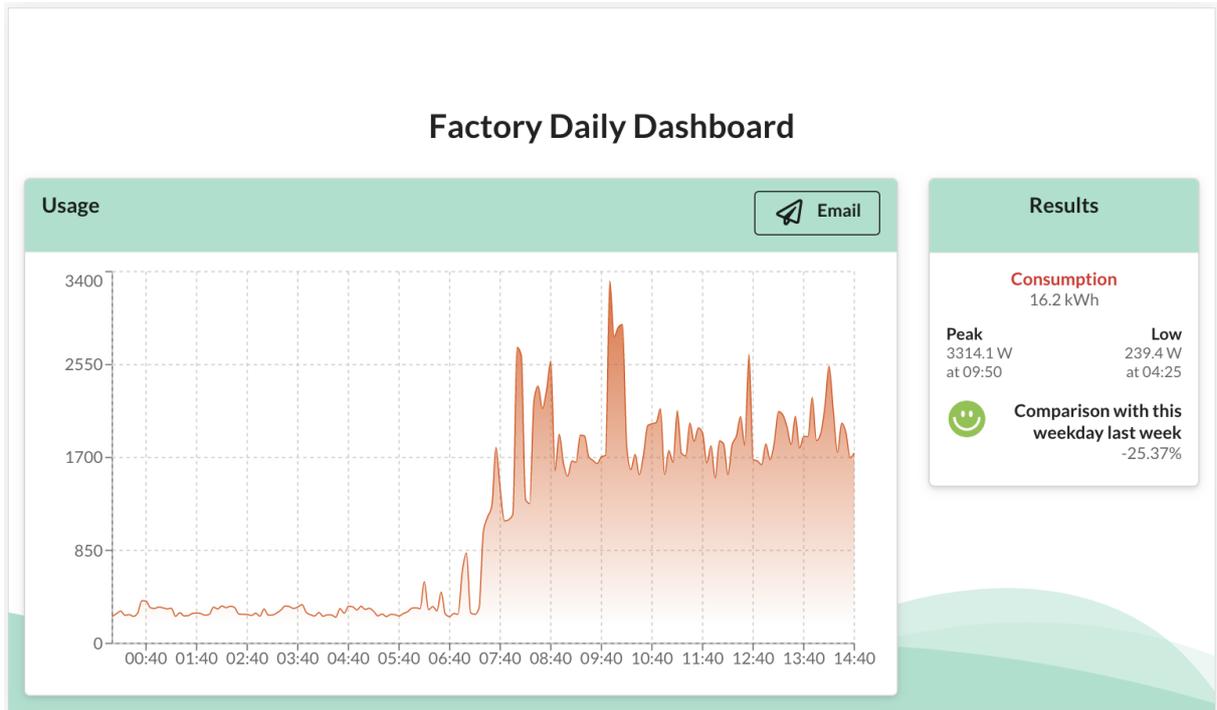


Figure 3.12: Factory Daily Dashboard (without energy production)

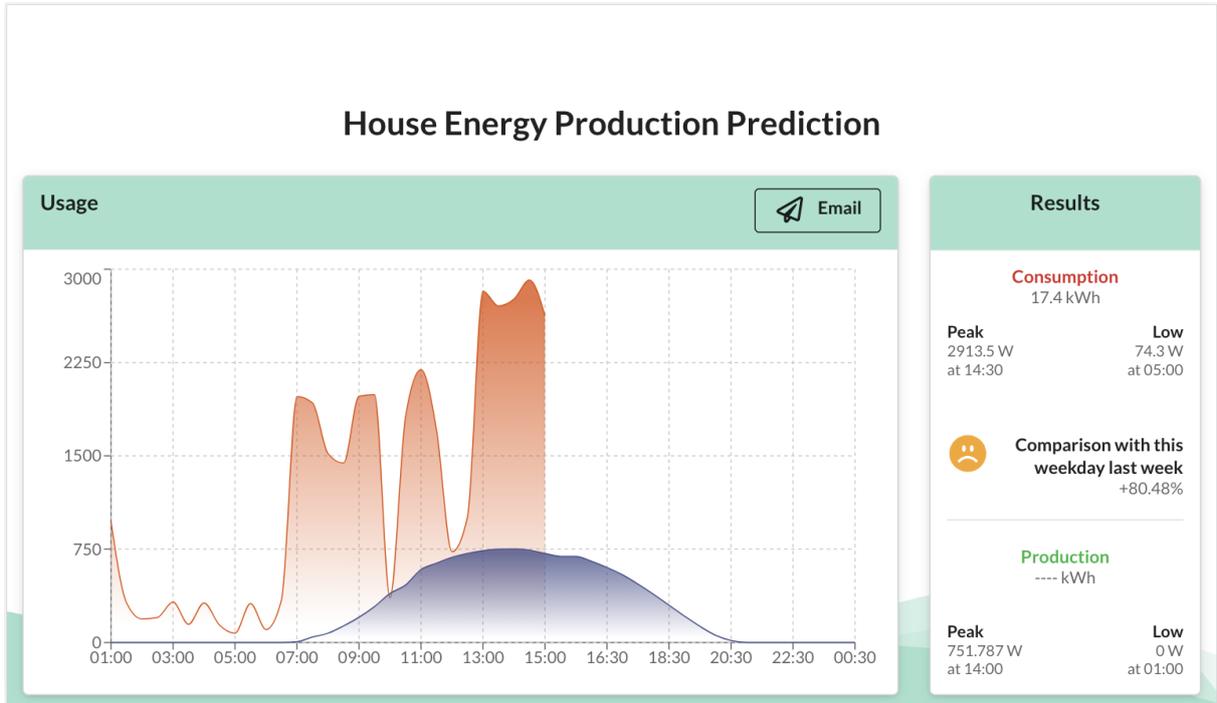


Figure 3.13: House Energy Production Prediction Dashboard (for facilities without energy production)

Finally, Figure 3.12 shows the special environment, that is, a factory. In this figure, it is possible to

see much higher consumption values compared to daily consumption at home.

Because in some environments there is no energy production, and the information presented to the user is only regarding energy consumption, we found it interesting to create a perspective of production, so the user would know how it could be in their house. For that matter, a special interface was designed to demonstrate a prediction of the energy production at the user's device location, together with the actual daily consumption.

As presented in Figure 3.13 when the user calls this dashboard, they will have an insight into the energy production if they install solar panels at their location. Similar to what users with energy production in their homes see in the daily dashboard, here are presented the peak and low values of the energy produced.

Since these users do not have solar panels, we introduced a new feature, latter added into the detailed dashboards as specified above, to inform the user about energy production. Similar to those dashboards, an email is sent with information regarding energy production and solar panels, as Figure 3.14 shows. In the email are presented direct links to solar panel solutions from the largest energy companies in Portugal and a link to an article from the Portuguese consumer's defense company, which helps to choose the best solution. Additionally, a screenshot of the interface is attached to the email, so that the user has a reference of the consumption and the expected energy production.

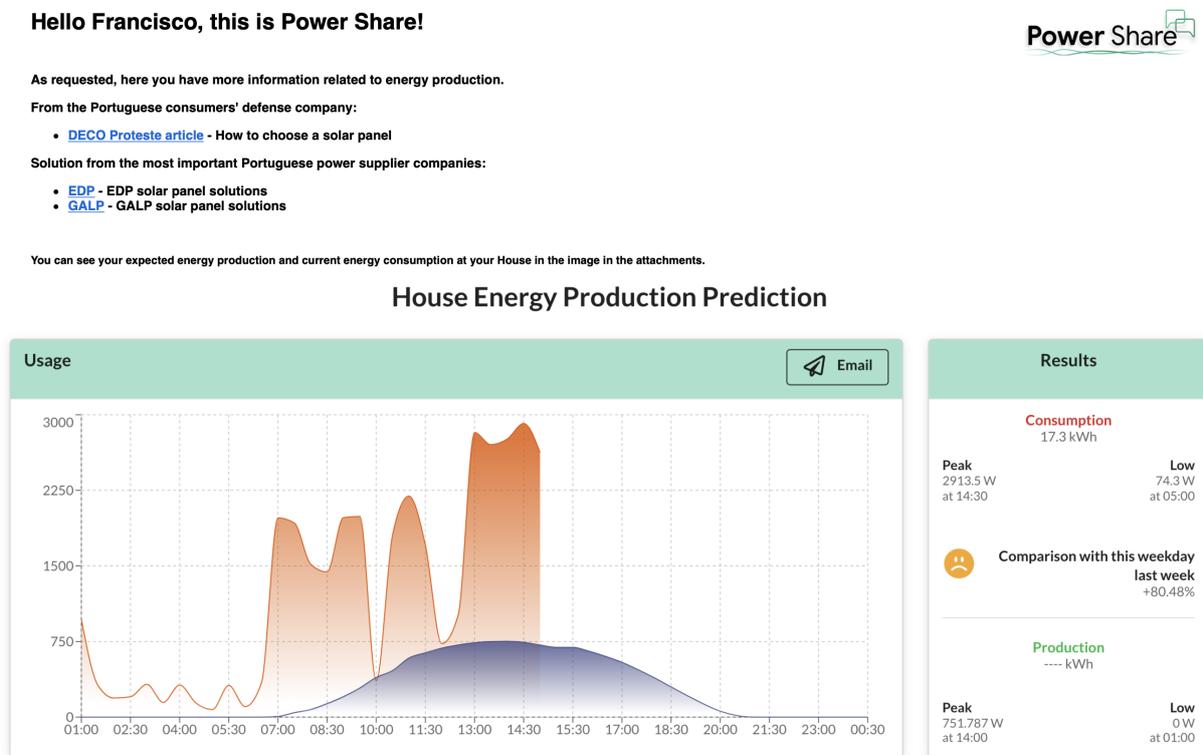
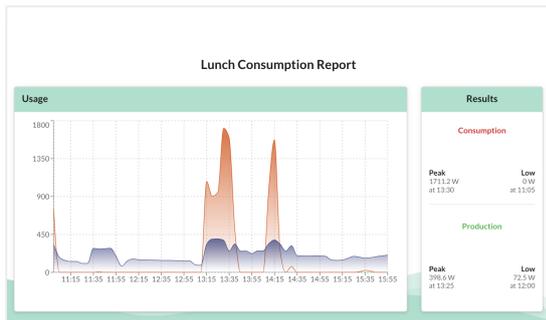


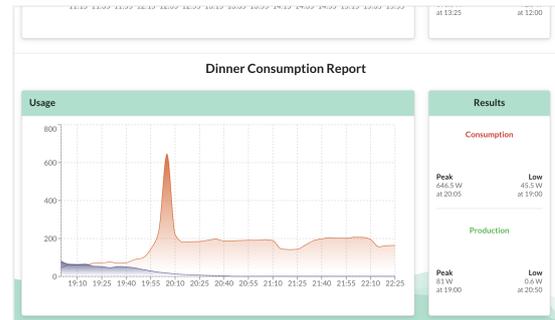
Figure 3.14: Email about energy production

Finally, the last energy-related interface was developed to provide helpful and focused information on the most significant parts of the day, lunch and dinner. Similarly to the daily dashboard, it is possible to see information about energy consumption and production but filtered to mealtime as presented in Figure 3.15. In the first graph, the time is filtered between 10:00 and 14:00 hours, showing the activity at lunchtime, and the time in the second graph is between 18:00 and 23:30 hours, representing dinnertime.

To complement it, we created a daily update notification feature. With this feature we intend to send daily notifications about the meals dashboard. Whenever the user calls this interface, they can decide whether they want to receive a daily update or not. If so, they can choose the hour to receive it.



(a): Lunch Dashboard (with energy production)



(b): Dinner Dashboard (with energy production)

Figure 3.15: Meals Dashboard (with energy production)

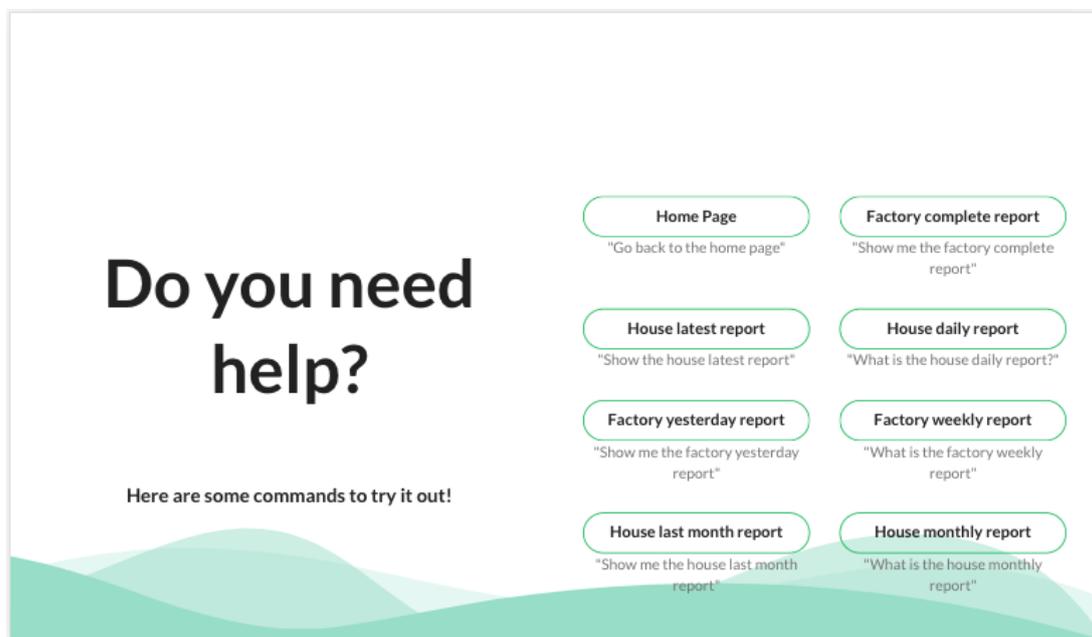


Figure 3.16: Help Page (with factory environment)

After completing the design of the main interfaces, we found it fundamental to have a help page

to guide the user if they need it. Not only that, but given that this application is voice-controlled, it emphasizes the need to help the user find the correct ways to navigate the app. As presented in Figure 3.16, the user is faced with several buttons to help them see the different commands available. Similarly to the home page 3.5, after each button, a possible sentence is shown for the user to say to go directly to the dashboard. Each of the previous interfaces is adapted to the environment in which it is being used.

3.3 Google Assistant

As referred to in the related work section 2.3, Google offers a platform called DialogFlow [26], which allows developers to create their own Google Assistant applications. Later, they developed a new conversation interface builder called Google Actions, which aims to improve the previous version, DialogFlow. This new platform contemplates better and simpler interfaces that help developers to have more insights from the application, for instance, the errors and warnings are now displayed on the screen. With this tool, it is possible to build an application supported not only on Google Nest products but also on any smartphone with the Google Assistant app installed, with some limitations for IOS devices [31]. Another limitation of this tool, that affects the system testing, is the fact that it is not possible to create a shortcut, for instance an icon or an always-on-display functionality, to have better access to the application. The user is forced to invoke the app using their voice, as explained in section 5.2.

3.3.1 Google Assistant's process

In every Google Assistant applications, the assistant follows the same process, from receiving a command from the user to its fulfillment. This type of application is commonly denominated Action, as is part of Google Actions, applications developed for Google Assistant. The flow of this process is illustrated in Figure 3.17.

The process starts with the user requesting a command to Google Assistant using their voice. The assistant then sends the request for Google Home, i.e., to local devices to execute the commands much faster. If the request fails, it will try to request via the cloud, which can be the default execution method, which minimizes the request failure. After the commands are processed either locally or via the cloud, the request is sent to the correspondent appliances and fulfilled.

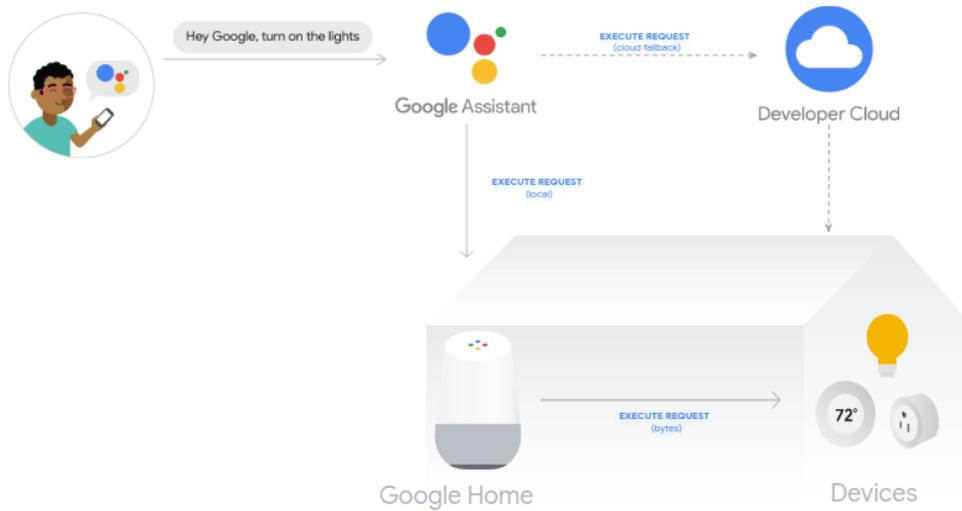


Figure 3.17: User commands [32]

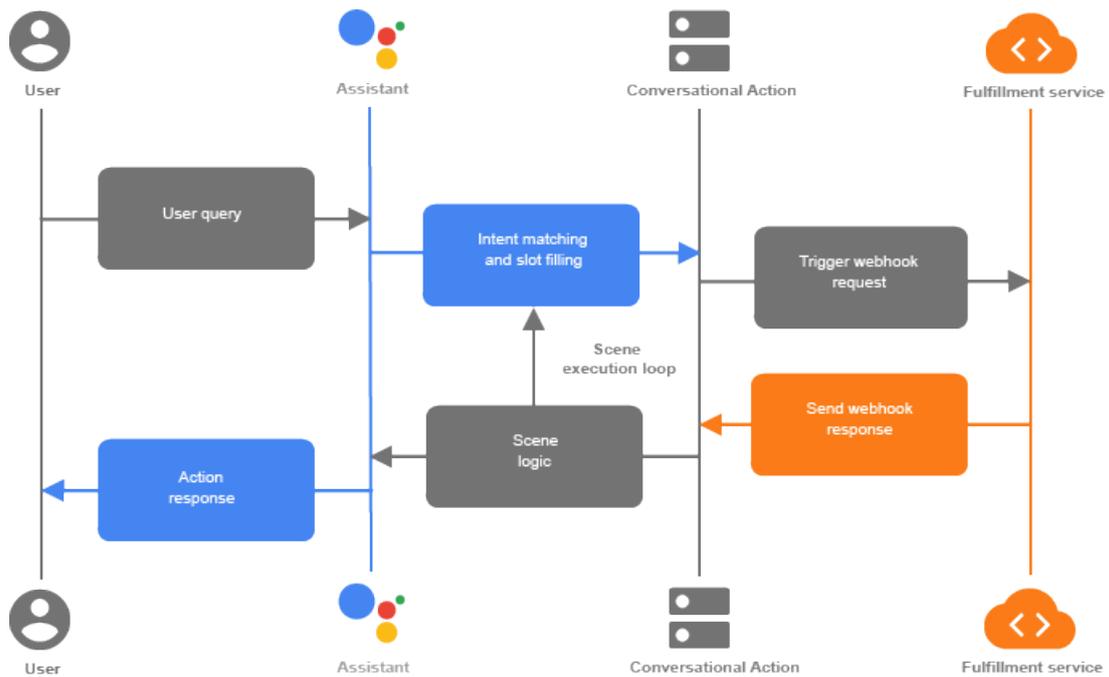


Figure 3.18: Conversation process [33]

In more detail, the process behind every interaction between the user and the application is illustrated in the previous figure.

A – Google Assistant’s detailed process. Google Assistant applications have three to four intermediates or entities. These entities are the user, the Assistant itself, the conversational action, and

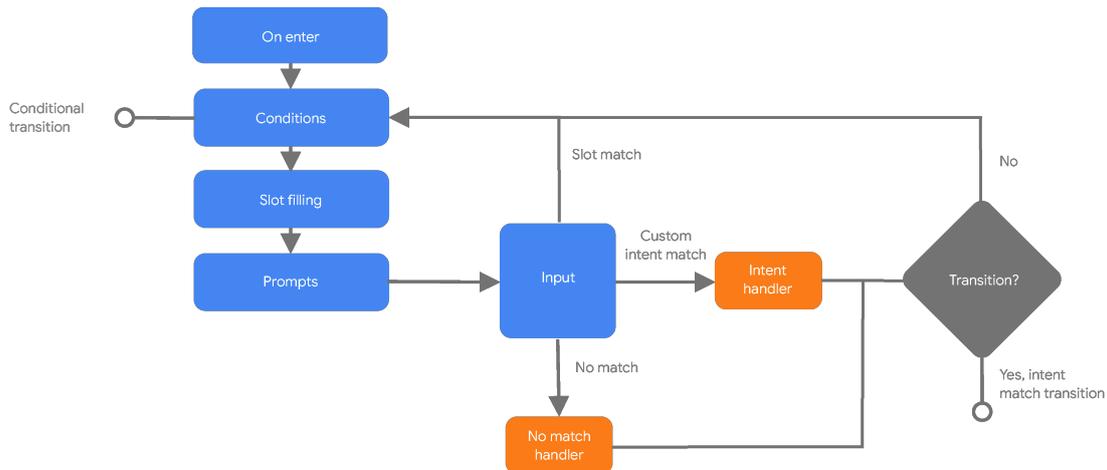


Figure 3.19: Scene execution lifecycle [35]

finally, for applications that require it, the fulfillment service, which is the case of applications with custom interfaces. This specific type of application uses the *Interactive Canvas* functionality.

These applications are composed of 4 core elements that interact with the previous entities:

- **Scenes:** One of the most important components of a conversational application. Represent the state of the application and maintain the logical aspect of the conversation, and each scene has its life cycle, as shown in Figure 3.19 [34]. In every scene is specified the conditions to be met, and the special type of words that are verified in slot filling, among other configurations. Ultimately, the scene shows the user a prompt or calls a webhook, represented by an interface.
- **Intents:** Intents are another important component in this type of application. An intent is a bridge between users' commands and the scene. To navigate from scene to scene, the user has to match the intent corresponding to the one that leads to the desired scene. Each intent has its training phrases, that is, possible commands that the user can say, and they are processed by Natural Language Processing (NLP) and Natural Language Understanding (NLU). The processing of these training phrases allows users to have word variations in their commands [36].

Intents are divided into system intents and custom intents:

- **System Intents:** Default tasks, defined by Google, that the action needs to fulfill, such as when the user says "yes";
- **Custom Intents:** Intents specific to the application, defined by the developer. Custom intents, in addition to training phrases, also have parameters. These parameters are typed data or specific words extracted from the user input by the NLU.

Finally, these intents can be divided into regular custom intents and global intents. While usually intents are only matched during a conversation, global intents have the particular

aspect of being possible to be matched whichever the application status is. The application can be in any scene or even turned off, and when the user invokes this intent, the application will process the request and fulfill it.

Figure 3.20 represents the application flow, using intents to transition from scene to scene.

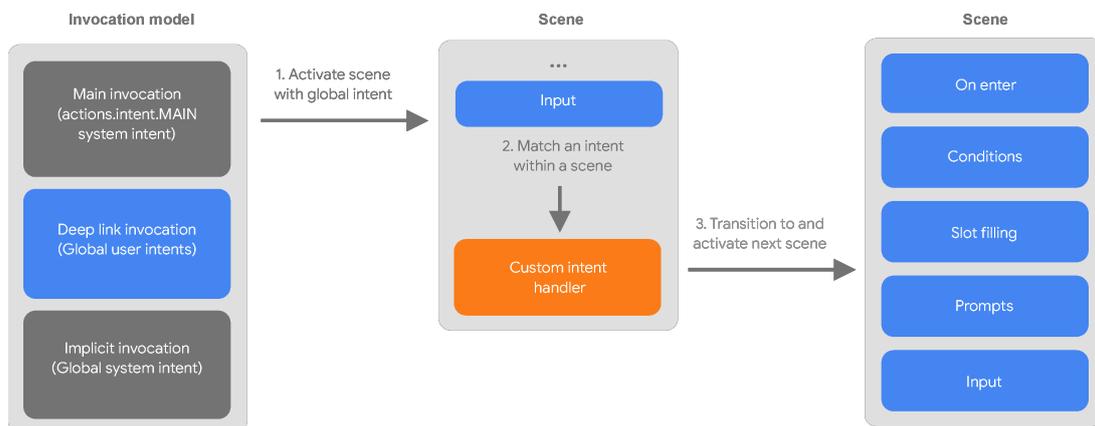


Figure 3.20: Intent matching scenario [37]

- **Types:** Structured data with a key/value structure. Allows, for example, to give a word as a key, and have word variations as values, processed by Assistant NLU. This way, the user can say different variations of the same word. This is also extensible to phrases [37].

It is possible to configure NLU to extract only the specified types from the user input or even accept unknown values [37]. The NLU will evaluate and extract these types in the scene's slot-filling stage and the scene's conditions stage.

3.4 Implementation

3.4.1 Smart Meters

Given the smart meter solution, composed of two smart meters and a Strato Pi represented in Figures 3.1 and 3.2, and the circuit described in Figure 3.3, we started configuring all the devices. The configuration of the smart meters was simple, as it only required changing the identification of one of the meters so that we could distinguish the one that measures the power of the factory from the one that measures the house.

As for the Strato pi, it needed more configurations. As mentioned in 3.1.2, inside the Strato pi is a Raspberry pi 4, responsible for processing the information received. First, was installed a fresh copy of the Raspberry pi OS image [38] on the Raspberry pi, so we could start configuring it with the additional

software needed to collect information from the meters, store it and send it to the SMILE API. With the installation completed, we passed the code implemented for the project SMILE to Raspberry, which reads the data from the meters. To this code, we changed some parts to adapt to the house/factory environment, in which both meters are used to read energy consumption from a triphasic power setup. After changing the code, we created a MySQL database so that the data read from the meters is stored locally to prevent data loss if the connection with the SMILE API fails.

At each second, the Strato Pi reads from the meters and stores the data in the local database. Parallel to this process, each minute, Strato reads from the database the data from the two meters, send it to the SMILE API, and writes the process on a log file for further analysis. After completing these configurations, the devices were connected and mounted into the electrical panel, and the system started to read and send data.

The setup of the devices in the electrical panel is represented in Figure 3.21. The smart meters are connected in parallel in the electrical panel, finally connecting to the Strato pi via the RS485 communication port.

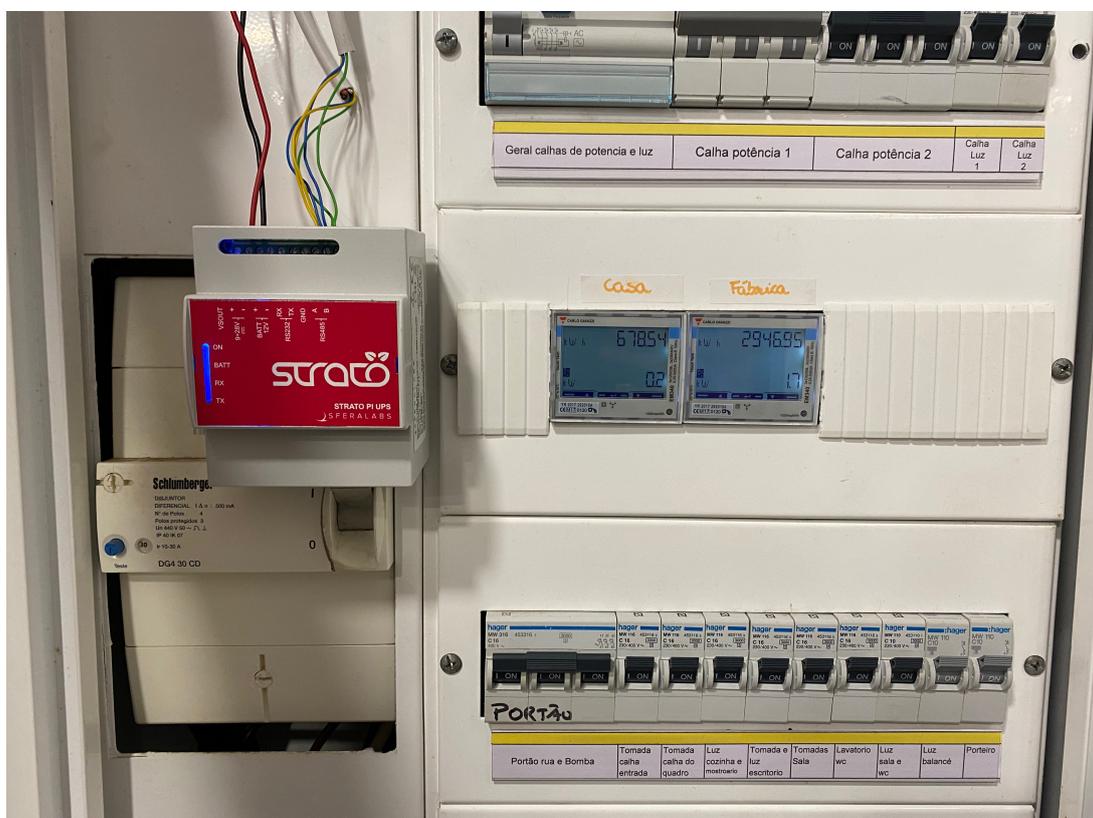


Figure 3.21: Smart meter solution in the electrical panel

3.4.2 Power Share Application

The Power Share application is designed to provide custom interfaces regarding energy use at the user's home. As described in Subsection A –, these applications are composed of an additional entity, the fulfillment service, in other words, a web application. With this said, Power Share is divided into two applications, a Google Actions application, and a web application.

3.4.2.A Google Actions Application

Given the knowledge of the structure of Google Actions applications described in Section 3.3, the development process started with the setup of the overall application. To create the application is firstly required to have a Google Firebase project [39] and a Google Cloud project [40]. These projects are connected since all the information gathered from the Google Firebase is stored on the Google Cloud buckets.

Google Firebase offers many services, from databases to application monitoring. But in this particular project, we used the hosting and functions services. The first service hosts the web application created to support the visual component of the Power Share application, and the latter supports the creation of the Google Actions application.

The Google Actions application is developed on Google Actions Console, as referred to in section 3.3. In this platform, we have created a project for the application and started configuring it, giving it a name, and choosing the main invocation, which is how the user invokes the app. From here, we followed the architecture bottom-up, starting from the types.

According to the design, we divided the first energy interfaces into summary and detailed dashboards, which means we had to create two types. In the first type are specified the complete and latest dashboards, in a key-value structure, where the values are other possible ways to refer to the key, as mentioned in Subsection A –. In the other type are specified the daily, yesterday, weekly, last week, monthly, and last month dashboards. Since this project has more than one test environment, we needed to create one more type referring to the three facility options, house, factory, and restaurant.

For each type of dashboard (detailed and summary), we created an intent with training phrases composed of the dashboard and facility type.

For example, "Show me the house daily dashboard", and "What is the restaurant complete dashboard". This composition allows the user to vary the commands as it is only required to match these two key values.

With the intents implementation completed, we created a scene for each of them. As described in Subsection A – in each scene we defined the conditions to show a specific interface. For instance, the conditions in the previous example phrases are that the facility type and the dashboard type must be present in the command given by the user.

In addition, we also defined the paths for when the conditions are met or not. If they were met, the scene calls a webhook from the web app and shows the desired interface. Otherwise, it calls the *Match Any* scene, which is visually represented by the help page since the user asked for something the assistant does not understand. Finally, to finish the development of these scenes, we specified the commands available to go from the current scene to another. We made the same development process for the production prediction dashboard, meals dashboard, help page, and home page development.

The production prediction dashboard and the meals dashboard required special attention during their conception since they both have features different from the other interfaces. For the production prediction dashboard, we need to have the device coordinates to present the expected energy production information. To get these coordinates, the user needs to create an account that will be linked automatically to their Google account. This process will make Google Assistant recognize the user by voice when invoking the app. With Google API integrated into the Google Action console and with the user's consent, it is possible to obtain the coordinates of the device on which their Google account is being used.

As for the meals dashboard, we intended to send notifications for the user to check this particular dashboard. Curiously this API has a dedicated part for user notifications and daily updates. The latter has the particular aspect of sending notifications at a given hour of the day, as presented in Figure 3.22. Even though the API offers such a feature, it is only supported by Android and IOS smartphones. That means that Google Hub, the home device used for this solution, does not support notifications. The users can set them on Google Hub, but they will receive them on their smartphones.

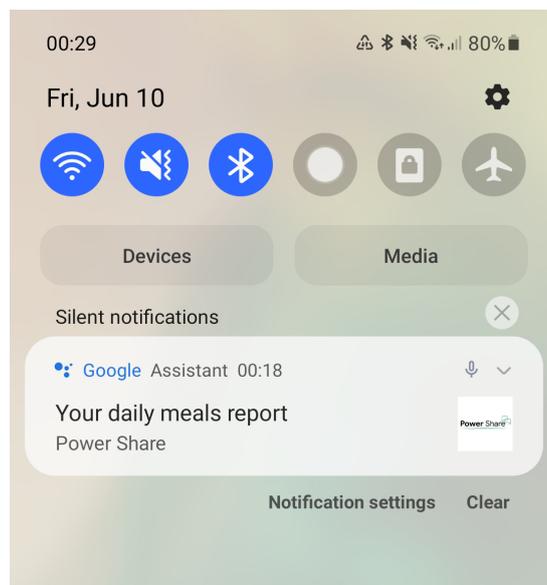


Figure 3.22: Meals Dashboard Notification

The structure of the Google Actions application is represented by Figure A.1 in appendix A. This

figure represents the various intents and scenes described previously, the main invocation, and the possible routes the user can take from a scene to a scene using voice commands.

3.4.2.B Web Application

As part of the Power Share application, a web app was developed to provide the interfaces designed previously for the user. This application was developed using JavaScript [41] and React [42] technologies, and the connection between the web app and the Google Actions app is made in this application.

The web application is composed of a control component and the components that represent the different interfaces. The control component, named Canvas, receives and processes the parameters from the user request and additional information provided by the Google API, such as the type of facility, the type of dashboard, and the user's email. In this application, we configured the screen size and rendered the components representing the desired interface.

The web app is also responsible for the rules of the Power Share application. It defines which users see each interface. For instance, users with energy production cannot see the production prediction interface, and users who do not own a factory cannot see information about this facility. To get the information needed to show in each dashboard, we created a Nodejs [43] server.

3.4.2.C Server

The server, an additional component of this solution, processes the information and makes it available for the web application to obtain. In this component, we defined the endpoints for each functionality, from the daily energy information collected by the meters to the prediction of energy production.

The daily, weekly, and monthly endpoints, and the meals endpoint, obtain information from the meters through the SMILE API.

On the other hand, the endpoint related to the prediction of energy production will make a request to the Solcast API. Solcast [44] is a company directed at developing tools for solar power systems. It provides many resources related to solar data, from historical and Typical Meteorological Year (TMY) to Photovoltaic (PV) live and forecast data. The Solcast resource chosen for this feature is the PV power, which provides live and forecast data for a given specific location in the world, which aligns perfectly with our goal. During the implementation of this endpoint, we had to consider the API access, which is limited to a certain number of daily calls.

To not exceed the call limit, we create a file with the live and forecast data after making a call to Solcast API. In subsequent calls, we verify whether there is a file with the data for that specific day and location. If it exists, the data is read from the file. Otherwise, a call to Solcast API is made, and the file is replaced. This way, we managed to save calls, as it only makes two calls per location per day. The first call gets the PV live data. Here, the time is limited from the beginning of the day to when the call was

made. The last call gets the PV forecast data. In turn, the time is limited from when the call was made to the end of the day.

As for the email endpoint, we created it with the help of SendGrid API [45]. SendGrid is an email sender API, which allows us to send an email to any recipient, and is compatible with Nodejs, using its npm module. To send a specific email with energy production information, we created an HTML preset. In this preset are presented direct links to solar panel solutions, an article that helps the user choose the right solution, and a screenshot of the interface, as Figure 3.14 shows. Similarly, we created an HTML preset for emails that present information about the current dashboard.

After making the requests in each endpoint to the correspondent API, the information is processed and parsed to be delivered as clean and structured as possible. The data is filtered to the corresponding date and hour range, the low and peak values are computed, and feedback is provided by calculating the difference between the previous and current values.

In more detail, this information is calculated for the precise moment when the user visualizes the interface. For example, in the daily dashboard, the feedback is computed by comparing today's consumption and production at the current hour with the same weekday of the previous week until the same hour. The same happens if the user asks for yesterday's dashboard. The feedback in the weekly and last week's dashboard is obtained by comparing this week's energy values with those of the previous week up to the same day of the week and hour. The feedback in the monthly and last month's dashboard is determined by comparing this month's consumption and production with those of the previous month on the same day and hour.

Finally, in every request made to SMILE API, the token and id of the user's facility are needed to get the correct information. So, we had to assign them to the user and use it in every request. Given that Google Assistant recognizes the user by their voice and makes the Google email available within the Power Share app, the server uses the email as a reference to the user to assign their correct facility tokens. Whenever a user interacts with the app, the server automatically uses the tokens to make requests correctly.

4

Evaluation

Contents

4.1 Evaluation Process	45
------------------------------	----

This section describes the evaluation methodology of the Power Share application and the results from the users' test.

4.1 Evaluation Process

The evaluation of this system is divided into three main goals, which include quantitative and qualitative data, to have a complete overview of the users' perspective on the system.

The goals are described as follows:

- **Evaluate the usability the system:** is intended to understand how users interact with the system;
- **Assessing the information of the system:** extract from the users' assessment what is the most helpful information to be presented in the system;
- **Evaluate the users acceptance to virtual assistants:** understand the user's perspective on virtual assistants, as well as their applications.

As mentioned, the system evaluation methodology will comprehend a qualitative and quantitative assessment. The quantitative assessment is made through the monitoring of the user's interactions with the system, with the help of Dashbot.io [46] and Google Actions Analytics, and the qualitative assessment is covered by semi-structured interviews. The duration of the evaluation varies between 2 weeks and a month.

4.1.1 Qualitative Assessment

The interviews made to users covered topics that were not possible to be assessed only using the interactions with the system.

The main topics are:

- **The use of the system:** how users proceeded to use the system;
- **The understand of the system:** did users understand the features available on the system, and which they used the most;
- **Feedback:** which features should be covered, which are not yet covered, and what is more important to show to users;
- **Willing to use:** understand if users are willing to use such an application afterwards;
- **The users acceptance of virtual assistants:** get to know if users are willing to accept and trust virtual assistant;
- **Future of automation:** users' feedback on similar systems being a default part of future homes;

4.1.2 Quantitative Assessment

Using both Google Actions Analytics and Dashbot.io [46], the interactions between users and the systems were recorded. Also, by comparing the energy consumption and production before and after using the system, it was possible to gather some interesting information during the pre-test. The recorded data let us know the following aspects:

- **The interactions with the system:** the aspects of the conversations between users and the app, such as duration, number of messages, etc.;
- **The interfaces called:** which interfaces were called the most and the least, and at which time the system is more used;
- **The energy usage:** differences on energy consumption and production when using the system (only in the pre-test);

4.1.3 Process

The evaluation process of the Power Share system was divided into two phases, a pre-test, and a final test.

The pre-test was intended to assess the system in the environment specified in section 3.1, which includes a house and a small factory. This test was performed by a family within one month. At the beginning of the month, the family was presented with a consent form and a general explanation of the system. To test the system, a Google Home Hub was provided since the participants did not own one.

The final test was performed by a family in Madeira, in a domestic scenario, and a restaurant in Lisbon. The research team assisted in the acquisition and installation of meters in this family's house and with the communication with the system. The family, and the restaurant, similar to the pre-test, received a Google Home Hub, instructions, and a consent form. This final test had the same duration as the pre-test, one month, divided into two weeks for the assessments in the restaurant in Lisbon and another two weeks for the family in Madeira.

By having these three distinguished types of environments, it is possible to understand the differences in how the users interact with the system and how it can be improved to fit best in each scenario.

4.1.4 Results

4.1.4.A Factory with Domestic

The data collected after this assessment showed that the most called interface was the detailed dashboards, specifically the *house daily dashboard*. This dashboard shows the current energy consumption with real-time feedback and, as expected, was the most requested during this test.

The second most visualized interface was the one that shows the expected energy production, which was quite interesting. This dashboard was specifically thought for houses that do not have energy production. The information displayed on this interface informs users about how it would be if they installed solar panels to produce energy, as it overlaps, in a graph, the expected energy production with the current daily consumption. It is reported that at least 54% of the time users called this interface, they used the email functionality. This means that 54% of the time when the Assistant asked users if they wanted to receive an email about energy production they said yes.

Following the ranking, the factory-related interfaces were the next most viewed, and the weekly and monthly dashboards related to the home and factory were the least requested. The ranking of interfaces is represented in Figure 4.1. This test had 367 interactions between the system and the users, having been most used around 12:00. These interactions, or so-called conversations, had an average duration of 1 minute and 47 seconds.

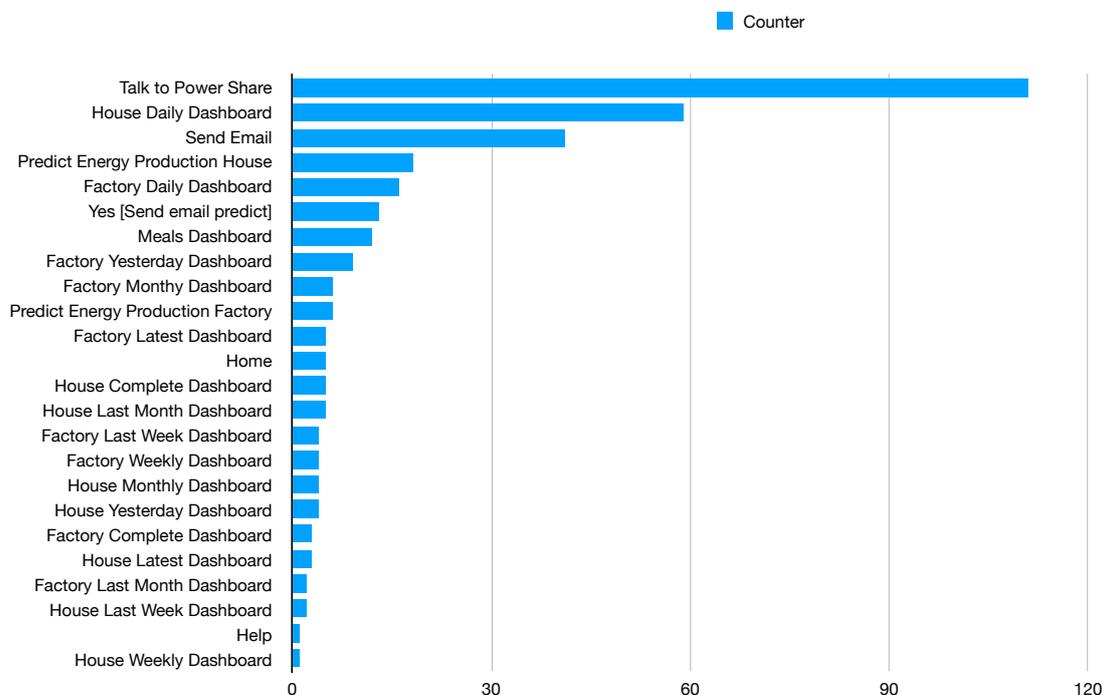


Figure 4.1: Interface ranking in Pre-Test

As mentioned in Subsection 3.4.2.A, whenever the user says a command that Google Assistant does not understand, the scene *Match Any* is called. During this test, the *Match Any* scene was called 25 times, which can be seen as a failure using the system. Thus, the success rate was 93.19%, as 342 interactions out of 367 were successfully fulfilled.

As specified in Subsection 4.1.2, one of the assessments made was the differences in energy consumption when using the system. As a result of this analysis, we found an interesting change in energy consumption during one weekend. During the test, the family studied their consumption habits and

thought about how to change them. As part of that, the connection between the electrical panel and the machines in the factory was turned off, as well as the air compressor machine. This decision resulted in a decrease of almost 1000W during a time when the factory was not working, as shown in Figure 4.2.

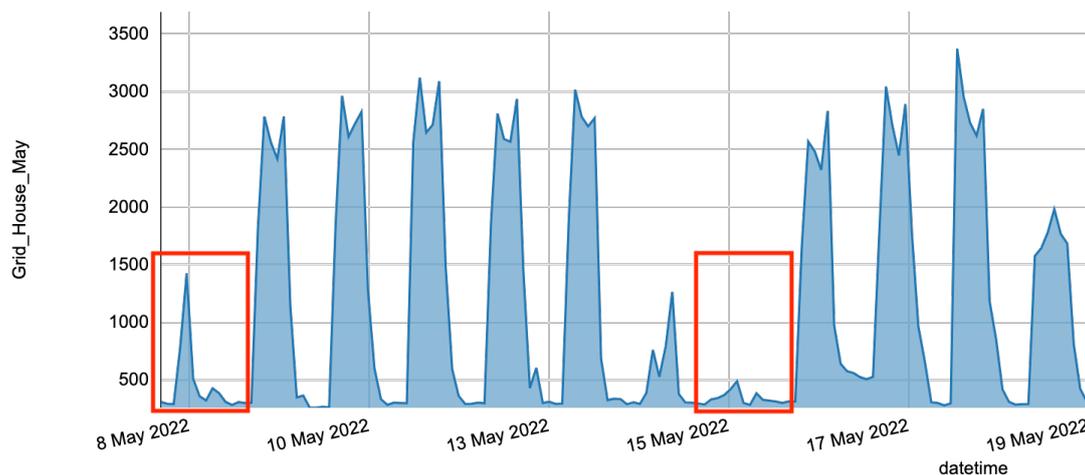


Figure 4.2: Factory Consumption Differences

Through the interview made to this family, it was possible to confirm the previous quantitative data and to conclude some aspects:

- **The use of the system:** Although users have used virtual assistants in the past, they have never interacted with such an application. They found the system easy to use, being the language processing the main difficulty as Google Assistant interpreted different words.
- **Feedback:** The feedback and email features were positively mentioned by the interviewees. They found the feedback about their energy consumption to be helpful to understand their energy consumption habits, and the email useful and informative. Additionally, they suggested that the email feature could be integrated into the detailed dashboard interfaces.

Users emphasized that factory owners could benefit considerably from using this system. Compared to their experience, where they turned off the connection to the machines during the weekend, the users said that with this system, the production line of a factory could be reviewed and changed to have better energy consumption values.

Finally, the users highlighted the benefits of using the system. During the test, they found ways to implement measures to reduce energy consumption by comparing the factory energy use between weeks. After the test, they continued to apply the same procedure every weekend and noticed a

considerable difference in the energy bill. Furthermore, having used the energy production prediction feature, users started considering installing solar panels. They began finding solutions for their specific environment, which includes a house and a factory with a high energy consumption.

- **Future:** Showing interest in using the system in the future, the interviewees even see the potential for similar systems to be integrated into future houses to control energy throughout the house.

4.1.4.B Restaurant

The restaurant differentiates itself from the other two environments by being only a workplace. This difference is reflected in the data collected during the system tests, for example, the interfaces that were called to show information about energy consumption. And most importantly, in fewer interactions with the system due to user availability during working hours.

It is essential to take this last statement into account because in professional environments, it is especially challenging to have time to perform tests with users, and therefore it influences the data collected.

Nevertheless, the results showed that the most called interface was the *restaurant complete dashboard*. In this interface, the user can see the energy consumption peaks and lows during the month, the week, and even the day, all at once.

As expected, the *restaurant daily dashboard* was one of the most visualized since this dashboard details the consumption throughout the day. The data displayed here also inform the user about their consumption on the same weekday in the previous week by comparing it with the current day. This information can be relevant for this specific environment as the number of clients and working machines varies daily.

Continuing the ranking, the *restaurant weekly dashboard* and the *restaurant monthly dashboard* were the following most preferred dashboards. These two interfaces can be particularly helpful for tracking energy usage variations and habits in the restaurant since they present the data from a more summarized perspective. Furthermore, the user can even compare the data to the previous week and month.

It is reported that the other functionalities were not used, suggesting a lack of availability to explore the rest of the system. This fact is justified by the type of establishment in which the test was performed. Interestingly, some of the not-accessed interfaces require, in fact, some time for the user to interact with the system and fully understand the given results. Moreover, the most called interface is the one that shows all the information in a summarized form to be easily visualized. With the data collected in this test, it is possible to conclude that the user was more interested in seeing information in real-time than information about the last days, weeks, and months.

The results of this test show that the interactions between the user and the system had an average duration of 1 minute and 7 seconds and were made around 14:00, which is when the restaurant closes after serving lunch. The success rate, measured by the number of times the scene *Match Any* was called, was 88.57%.

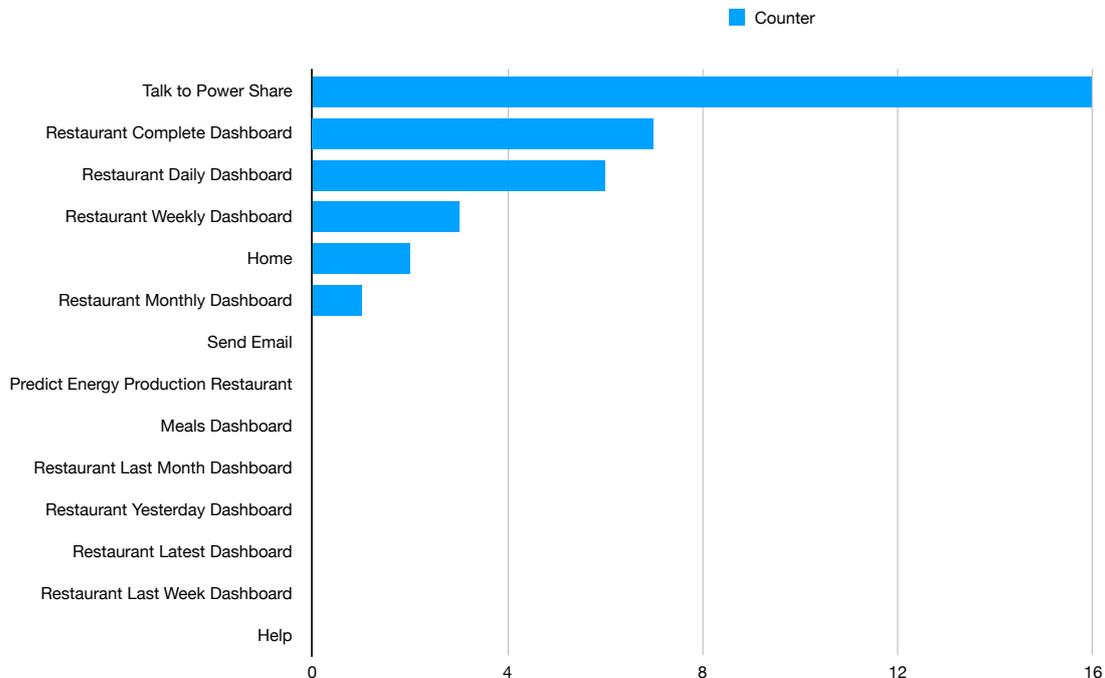


Figure 4.3: Interface ranking in Restaurant

4.1.4.C Domestic

Following the same procedure as the previous assessments, the data collected by Dashbot.io allowed having a clear perspective of the most used features. The dashboard that showed the most interest on the part of the user was, as expected, the *house daily dashboard*. As mentioned above, this interface shows real-time energy consumption and production. It also provides continuously updated feedback by comparing the data with those of the same day in the last week. This feature is commonly the most desired one, as it gives the perception of the current energy values, which users often do not know.

Interestingly, the second most asked dashboard was the *house latest dashboard*. This action can be seen as a complement to the previous feature, as this dashboard gives summarized information regarding the last day and last week's energy consumption and production.

The next data reveals a surprise since this specific functionality was suggested by the other family in the pre-test. In the pre-test, the users frequently asked the assistant to show the prediction of the energy production at their location, and when they got the data, they had the possibility to receive it in their email. The email feature was so well received that the family suggested amplifying it to the other

dashboards. As a result, this feature was included in the detailed dashboards and was made available during the final test. And it was one of the most used features during this family assessment.

While the user was very interested in seeing the information from a closer perspective, that is, at the day level, they also explored other interfaces where the data is displayed differently. They chose to consult the *house complete dashboard* and the *house weekly dashboard*. The *house complete dashboard* condenses the daily, weekly, and monthly energy consumption and production values into a single interface. The *house weekly dashboard* presents the current week's energy values in a much more detailed form, similar to the *house daily dashboard*.

The last called interface was the *meals dashboard*. This feature presents the energy data during lunch and dinner time by filtering it to the respective hours. It is interesting to realize that only the families of this test explored this interface.

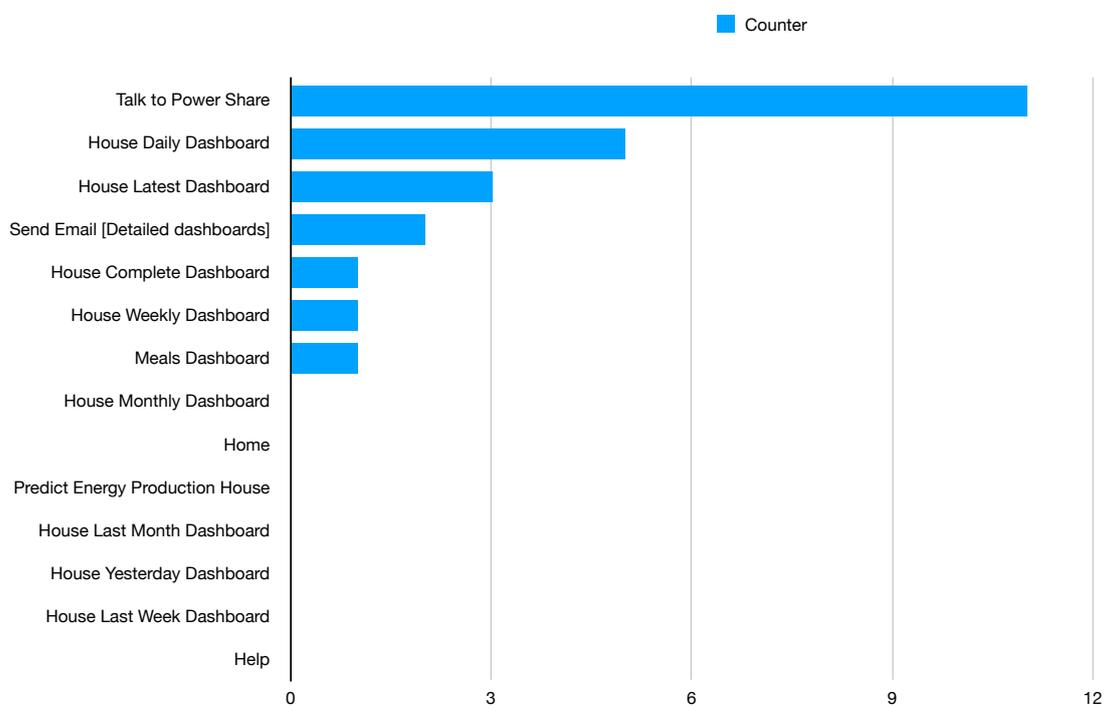


Figure 4.4: Interface ranking in Houses in Madeira

The data shows that the interactions with the system lasted an average of almost 1 minute and were mainly performed around 21:00. As for the well-interpreted commands, it is stated a success rate of 76%, measured by the calls made to the *Match Any* scene.

The results of this family test lead to the conclusion that the user was more concerned about the real-time energy information. This assertion is, in fact, applicable across all of the system tests. Users tend to be more interested in current values, whether they are daily, weekly, or monthly data.

In addition, the comparison of this family assessment with that of the family in the pre-test showed

some similarities in the way they used the system. Both families requested to visualize the *house daily dashboard* more than any other interface and also saw their energy consumption and production during mealtimes on the respective dashboard, as mentioned above. And finally, the email with detailed information about the dashboards was another commonly desired functionality. A notable difference between the families was regarding energy production. One family does not have energy production in its facilities but was interested in installing solar panels when faced with the values predicted by the system. On the other hand, the other family has energy production and used the system to manage it better, and does not use energy from the grid.

Finally, it was then performed final interview, to consolidate the gathered data during the test.

- **The use of the system:** The user has never interacted with any application of this kind, nor virtual assistants, which led to difficulties, such as Google Assistant interpreting the commands incorrectly. However, they were able to see the dashboards mentioned above and gather useful information.
- **Feedback:** Given that the user was not familiar with such technologies, they suggested extending the system to a mobile application or online platform to not be limited to voice commands. The user expressed preferring the conventional applications.

With the range of available functionalities, they found helpful information to be aware of real-time consumption and production and also be able to understand the data displayed. They highlighted the email feature for the details presented in each email and the convenience of having system data saved and available to easily consult at any time.

From the information provided by the system through the interfaces and the email, the user was able to track their energy habits and even take measures to be more efficient. By analyzing the energy production flow throughout the day, the user configured the heat pump to turn on when the production was reaching the highest values. And when the heat pump was not working, they turned on appliances, such as the washing machine, to use the energy produced instead of the grid energy.

Moreover, they emphasized the system's value in increasing people's awareness of energy use and characterized the application as sporadic consultation. That is, this application, contrary to others, does not require the users' attention so often.

Finally, the integration of notifications was suggested, in future versions of the system, for when the energy produced is being wasted. That way, users would be able to improve their routines and use energy instead of wasting it.

- **Future:** The interviewee found interest in using the system afterward and saw potential in being part of future houses, highlighting their previous suggestion.

5

Conclusion

Contents

5.1	Conclusions	55
5.2	Limitations	56
5.3	Future Work	57

This section presents the conclusions taken from this work, the limitations found during the development and testing of the system, as well as future work.

5.1 Conclusions

Technology is evolving more and more, and virtual assistants are becoming more helpful, which reflects an increase in its presence in people's lives in home assistant devices. With the evolution of IoT, many solutions emerged from different industries and expanded the power and functionalities of virtual assistants. The interaction between these two technologies made the transfer of data across multiple devices easy, enabling the automation of daily tasks and leading to personalized user experiences. Moreover, people now have access to information they were not able to have before these technologies.

Power Share is an integrated system that comprehends a smart meter solution, a conversational Google Assistant application, a web application, and a server. The solution was envisioned to raise people's awareness about energy, informing them about its use by using a virtual assistant and seamlessly integrating it into their daily lives. Power Share provides attractive interfaces regarding facilities' energy consumption and production while offering an engaging virtual assistant experience. These interfaces include a variety of dashboards in which users can see historical and real-time data from different perspectives and even compare them to understand their energy use habits.

The system was successfully tested in three distinguished environments with different types of energy use, namely, houses, houses with a factory, and a restaurant, demonstrating its versatility in different scenarios. Power Share receives this characteristic by not only being adaptable to various facilities but also by being available across multiple platforms, such as Google Home Hub and Android smartphones.

The tests were split into a pre-test and a final test, in which were conducted quantitative and qualitative assessments to understand the user experience. The house with the factory was included in the pre-test that lasted a month. The houses in Madeira and the restaurant in Lisbon were part of the final test and lasted two weeks each.

From the data collected by Dashbot.io [46], the interactions were short-lived as the system has a fast response rate, and the dashboard displays information efficiently, allowing it to be integrated into work environments such as restaurants and factories, where time is scarce. In this type of environment the summarized dashboard was preferred, since the data is more consolidated and embraces more than one scope.

Furthermore, the data shows that the real-time comparison between days, weeks, and months was one of the most praised functionalities, as it allows users to implement new measures to reduce their energy consumption and improve production. Users even expressed a tendency to prefer real-time information over data about past days, weeks, and months. Finally, the tests characterized the system

as a system of long-term use, as people use it over time and not exhaustively over a short period of time.

Testing the system resulted in users monitoring their energy use habits and creating effective ways to mitigate unnecessary energy consumption. It even boosted users' curiosity to start producing energy and finding the best solar panel solutions. To conclude, the adherence to testing the system, as far as the number of interactions, aligns with the previously made studies and with the literature.

This new iteration of the Power Share project takes it to the next level, incorporating new technologies in which the virtual assistant excels, contributing to the future of IoT and home automation. Not only that, but Power Share differs from the solutions available on the market by being adaptable to any environment, available on multiple devices, and requiring little equipment.

This work contributed to bringing users closer to virtual assistants and taking advantage of them while keeping them informed about their energy use in their facilities. Users found functionalities that helped them to manage their energy and even explore ways to produce it by installing solar panels. And are even willing to use the system in the future for the same purposes. Ultimately, it contributed to leveraging the research on home assistants and their potential in the energy field, as we believe that it can be further explored in future investigations.

5.2 Limitations

The current system is a conversational application, built on Google Actions platform. Google offers this platform an API to develop Google Assistant applications, but it has some limitations in its functionalities.

Most of the methods available in this API are under *Developer Preview*, meaning, it is only possible to use such functionalities in the simulator. As soon as the app is deployed, these methods no longer work, and the features developed using them will not work properly. An example of that, is the method, used in the web app, that allows developers to use the Google Assistant voice to say custom phrases.

Interactive Canvas, the functionality of which the previous API is integrated, has some limitations. This functionality allows to show custom interfaces on Google Assistant apps, but it is only supported on a few devices, such as smart devices from Google Nest and Android phones. Meaning, that IOS smartphones do not support such functionality, which reflects in an incapability to use Power Share on these devices.

Furthermore, the notification feature implemented on Power Share to inform the user to see the *meals dashboard* has limitations. Here, they are divided into two aspects. First, the Google Home Hub, the device used in this system, allows to configure the notification correctly, but it is not capable of

receiving them. And secondly, IOS smartphones are capable of receiving such notifications, but cannot set them, as it needs to open the app which is impossible on these devices.

A constraint arises when invoking the system. To do so, the users must say *"Hey Google, talk to Powershare"*, which restricts how they interact with the system. Instead of clicking the app icon like conventional apps, users have no choice but to say the phrase. This constraint can lead to users interacting with the system less, as they do not have a visual representation of the app, for instance, an icon, to remind them to use it.

Google recently made an announcement [47, 48] saying it will shut down the platform for developing conversational apps in June 2023. This is happening due to the fact that Google wants to make Google Assistant available through Android apps, which results in an extension of the typical Android application. Taking into consideration the developers opinion on Google Assistant, Google says that "users want to engage with their favorite apps using voice, and developers want to build upon their existing investments in Android" [47]. This makes Power Share to change, by turning into an Android app and extended it to use Google Assistant features, that promises to help "developers to create deeper, more meaningful voice-forward experiences" [47].

5.3 Future Work

Power Share was built with new technologies and procedures, namely Google Actions, which consequently gives opportunities for improvements in its various parts.

The Google Assistant application was developed with the latest technology and Google's latest platform, the Google Action platform. However, due to Google's decision to make its assistant only available through Android apps, the conversational application must be transformed into an Android application and consequently integrate the Google Assistant.

With the analysis of the test results, the interfaces can be improved to fit best the type of information that users want to see the most. Moreover, it is possible to understand which functionalities could be improved or added to provide a better experience.

In addition to the functionalities, the smart meter solution could be enriched by exploring the NILM integration and delivering specific information regarding each active appliance. With live data about the appliance, it is possible to create warnings to turn them off and notifications to inform them about their activity throughout the day, week, and month.

Bibliography

- [1] T. S. R. Correia, "Power share: Eco feedback and energy trading system," Master's thesis, Instituto Superior Técnico, 2018. [Online]. Available: https://fenix.tecnico.ulisboa.pt/downloadFile/1970719973967573/thesist_78572.pdf
- [2] M. Dukan, "Climate policy info hub," 2015. [Online]. Available: <https://climatepolicyinfohub.eu/energy-efficiency-policy-instruments-european-union>
- [3] M. Roccotelli and A. M. Mangini, "Advances on smart cities and smart buildings," *Applied Sciences*, vol. 12, no. 2, 2022. [Online]. Available: <https://www.mdpi.com/2076-3417/12/2/631>
- [4] S. R. Shaw, S. B. Leeb, L. K. Norford, and R. W. Cox, "Nonintrusive load monitoring and diagnostics in power systems," 2005. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/4463663>
- [5] S. Makonin and F. Popowich, "Nonintrusive load monitoring (nilm) performance evaluation," 2015. [Online]. Available: http://makonin.com/doc/ENEF_2014.pdf
- [6] Sense Labs. Home. [Online]. Available: <https://sense.com>
- [7] Emporia. Emporia Smart Home Energy Management System. [Online]. Available: <https://www.emporiaenergy.com>
- [8] Smappee. Smappee - Fueling energy efficiency for people and businesses. [Online]. Available: <https://www.smappee.com>
- [9] J. Froehlich, L. Findlater, and J. Landay, "The design of eco-feedback technology," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 1999–2008. [Online]. Available: <https://doi.org/10.1145/1753326.1753629>
- [10] H. Isyanto, A. S. Arifin, and M. Suryanegara, "Design and implementation of iot-based smart home voice commands for disabled people using google assistant," in *2020 International*

- Conference on Smart Technology and Applications (ICoSTA)*, 2020, pp. 1–6. [Online]. Available: <https://doi.org/10.1109/ICoSTA48221.2020.1570613925>
- [11] M. J. A. Baig, M. T. Iqbal, M. Jamil, and J. Khan, “Iot and blockchain based peer to peer energy trading pilot platform,” in *2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 2020, pp. 0402–0406. [Online]. Available: <https://doi.org/10.1109/IEMCON51383.2020.9284869>
- [12] H. Jayakumar, A. Raha, Y. Kim, S. Sutar, W. S. Lee, and V. Raghunathan, “Energy-efficient system design for iot devices,” in *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2016, pp. 298–301. [Online]. Available: <https://doi.org/10.1109/ASPDAC.2016.7428027>
- [13] Insider Intelligence. Voice Assistants in 2022: Usage, growth, and future of the AI voice assistant market. [Online]. Available: <https://www.insiderintelligence.com/insights/voice-assistants/>
- [14] S. Scuri, G. Tasheva, L. Barros, and N. J. Nunes, “An hci perspective on distributed ledger technologies for peer-to-peer energy trading,” in *Human-Computer Interaction – INTERACT 2019*, D. Lamas, F. Loizides, L. Nacke, H. Petrie, M. Winckler, and P. Zaphiris, Eds. Cham: Springer International Publishing, 2019, pp. 91–111. [Online]. Available: https://doi.org/10.1007/978-3-030-29387-1_6
- [15] S. Scuri and N. J. Nunes, *PowerShare 2.0: A Gamified P2P Energy Trading Platform*. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: <https://doi.org/10.1145/3399715.3399948>
- [16] IOTA. IOTA. [Online]. Available: <https://www.iota.org/get-started/what-is-iota>
- [17] R. J. P. Martins, “Power share - blockchain management system and p2p energy trading,” Master’s thesis, Instituto Superior Técnico, 2019. [Online]. Available: https://fenix.tecnico.ulisboa.pt/downloadFile/563345090418820/79532-Ruben-Martins_dissertacao.pdf
- [18] IBM. What is Blockchain Technology? - IBM Blockchain — IBM. IBM - What Is Blockchain. [Online]. Available: <https://www.ibm.com/se-en/topics/what-is-blockchain>
- [19] BurstIQ. BurstIQ — Research Foundry — blockchain based healthcare data solutions. [Online]. Available: <https://burstiq.com>
- [20] E. S. Kang, S. J. Pee, J. G. Song, and J. W. Jang, “A blockchain-based energy trading platform for smart homes in a microgrid,” in *2018 3rd International Conference on Computer and Communication Systems (ICCCS)*, 2018, pp. 472–476. [Online]. Available: <https://doi.org/10.1109/CCOMS.2018.8463317>

- [21] B. K. Barman, S. N. Yadav, S. Kumar, and S. Gope, "Iot based smart energy meter for efficient energy utilization in smart grid," in *2018 2nd International Conference on Power, Energy and Environment: Towards Smart Technology (ICEPE)*, 2018, pp. 1–5. [Online]. Available: <https://doi.org/10.1109/EPETSG.2018.8658501>
- [22] S. K. Vishwakarma, P. Upadhyaya, B. Kumari, and A. K. Mishra, "Smart energy efficient home automation system using iot," in *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, 2019, pp. 1–4. [Online]. Available: <https://doi.org/10.1109/IoT-SIU.2019.8777607>
- [23] IFTTT. [Online]. Available: https://ifttt.com/explore/new_to_ifttt
- [24] M. Poongothai, K. Sundar, and B. Vinayak, "Implementation of iot based intelligent voice controlled laboratory using google assistant," *International Journal of Computer Applications*, vol. 182, pp. 6–10, 09 2018. [Online]. Available: <https://doi.org/10.5120/ijca2018917808>
- [25] M. S. Hadi, M. A. A. Shidiqi, I. A. E. Zaeni, M. A. Mizar, and M. Irvan, "Voice-based monitoring and control system of electronic appliance using dialog flow api via google assistant," in *2019 International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, vol. 6, 2019, pp. 106–110. [Online]. Available: <https://doi.org/10.1109/ICEEIE47180.2019.8981415>
- [26] Google. [Online]. Available: <https://cloud.google.com/dialogflow/>
- [27] C. Gavazzi. [Online]. Available: https://gavazziautomation.com/nsc/HQ/EN/energy_power_analyzers
- [28] H. SMILE. [Online]. Available: <https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5bb185870&appId=PPGMS>
- [29] S. Pi. [Online]. Available: <https://www.sferalabs.cc/product/strato-pi-ups/>
- [30] G. Developers, "Design guidelines." [Online]. Available: <https://developers.google.com/assistant/interactivecanvas/design?hl=en>
- [31] —, "Interactive canvas." [Online]. Available: https://developers.google.com/assistant/interactivecanvas#supported_devices
- [32] GoogleDevelopersBlog. (2019, July) Illustration. [Online]. Available: <https://developers.googleblog.com/2019/07/developer-preview-of-local-home-sdk>
- [33] G. Actions. Illustration. [Online]. Available: <https://developers.google.com/assistant/conversational/actions?hl=ar>

- [34] G. Developers, “Scenes — conversational actions — google developers.” [Online]. Available: <https://developers.google.com/assistant/conversational/scenes>
- [35] G. Actions. Illustration. [Online]. Available: <https://developers.google.com/assistant/conversational/scenes>
- [36] G. Developers, “Intents—conversational actions—google developers.” [Online]. Available: <https://developers.google.com/assistant/conversational/intents>
- [37] —, “Types — conversational actions — google developers.” [Online]. Available: <https://developers.google.com/assistant/conversational/types>
- [38] R. Pi. [Online]. Available: <https://www.raspberrypi.com/software/>
- [39] G. Firebase. [Online]. Available: <https://firebase.google.com/>
- [40] G. C. Platform. [Online]. Available: <https://cloud.google.com/>
- [41] Mozilla. [Online]. Available: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>
- [42] Reactjs. [Online]. Available: <https://reactjs.org/>
- [43] O. Foundation. [Online]. Available: <https://nodejs.org/en/>
- [44] Solcast, 2019, “Global solar irradiance data and pv system power output data.” [Online]. Available: <https://solcast.com/>
- [45] SendGrid. [Online]. Available: <https://sendgrid.com/>
- [46] Dashbot.io. [Online]. Available: <https://www.dashbot.io/>
- [47] Google, “Helping developers create meaningful voice interactions with android,” June 2022. [Online]. Available: <https://developers.googleblog.com/2022/06/Helping-Developers-Create-Meaningful-Voice-Interactions-with-Android.html>
- [48] —, “Conversational actions sunset overview.” [Online]. Available: <https://developers.google.com/assistant/ca-sunset>



Power Share Structure

The structure of the Google Actions application for Power Share is represented in this appendix. The structure is composed of intents with user command examples, scenes, assistant responses, and the main invocation of the application. It is also represented the possible routes from scene to scene.

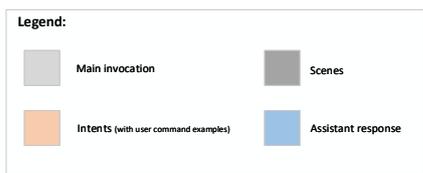
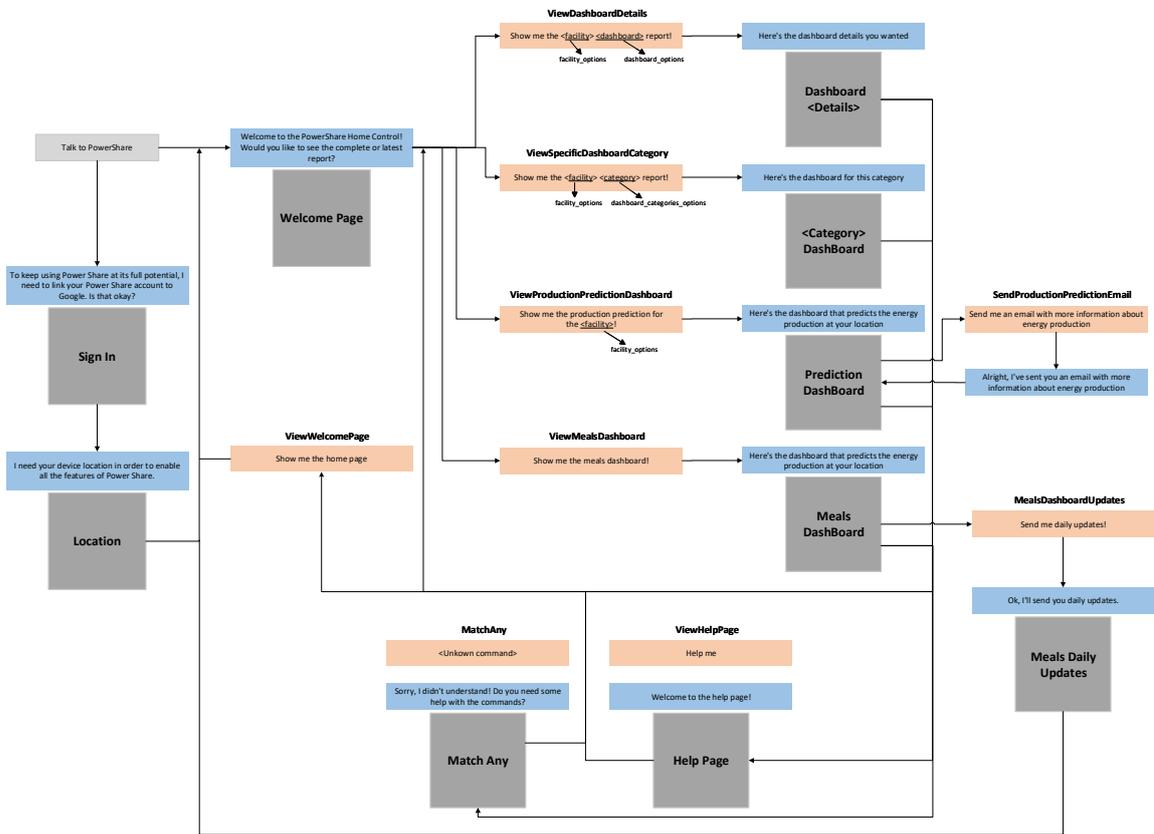


Figure A.1: Power Share Structure

