School of Electronic Engineering and Computer Science

Final Report

Programme of study:
BSc (Hons) Computer Science with Management (ITMB)

Project Title:

Re-Energise – Energy Management System

https://re-energise.xyz/

Supervisor:
Bing Han

Student Name:
Abid Miah

Final Year
Undergraduate Project 2022/23

Queen Mary University of London

Date: 2nd May 2023

Contents

Chapter 1: Introduction 5

1.1 Problem Statement 5

1.2 Background Context 5

1.3 Aim 6

1.4 Objectives 7

1.5 Research Questions 8

1.6 Project Milestones 8

Chapter 2: Literature Review 9

2.1 Context to the Energy Industry 9

2.1.1 Energy Market and Trends 10

2.1.2 Demand and Supply 11

2.1.3 Green Energy 11

2.2 Living Cost Crisis in the UK 12

2.2.1 Comparison 13

2.2.2 Impacts of Living Cost on UK households 14

2.2.3 Tracking Energy Consumption 16

2.2.4 Energy Providers and their Rates 16

2.3 Existing Systems 19

2.4 Proposed Solution 22

Chapter 3: Design and Analysis 23

3.1 Requirements Analysis 23

3.1.1 Functional Requirements 23

3.1.2 Non-functional Requirements 24

3.2 Use Cases 25

3.2.1 Use Case Diagram 25

3.2.2 Use Case Descriptions ... 25

3.3 SDLC - Development Environment ... 27

3.3.1 System Architecture - Django ... 27

3.3.2 Required Tools & Knowledge ... 28

3.4 Initial UI Screen Designs ... 28

Chapter 4: Implementation ... 30

4.1 Platform and Technologies Rationale ... 30

4.1.1 Django & Python ... 30

4.1.2 Bootstrap ... 30

4.1.3 PostgreSQL Database ... 30

4.1.4 Railway Platform ... 31

4.1.5 Google Cloud Platform ... 31

4.1.6 Conda Virtual Environment ... 31

4.2 Implementation Details ... 32

4.2.1 Database Models' Implementation ... 32

4.2.2 Setting up PostgreSQL database ... 35

4.2.3 Google Cloud Configuration ... 36

4.2.4 Django Admin Portal ... 37

4.2.5 Link to Repository and Website Link ... 37

4.2.6 UI and Backend Implementation ... 38

Chapter 5: Testing ... 46

5.1 Unit & Functional Testing ... 46

5.2 User Participant Testing ... 49

5.3 Non-Functional Testing ... 50

5.4 Penetration Testing ... 51

5.5 Test Findings & Summary ... 52

Chapter 6: Evaluation ... 53

6.1 Evaluation against Project Objectives ... 53

6.2 Evaluation against Existing Solutions ... 54

6.3 Project Strengths ... 54

6.4 Project Limitations & Challenges ... 54

6.5 Social, Legal & Ethical Issues ... 55

Chapter 7: Conclusion ... 56

7.1 Achievements ... 56

7.2 Future Implementation ... 56

7.3 Conclusion ... 57

References 58

Appendix 61

7.4 Gant Chart ... 61

7.5 Risk Assessment ... 62

7.6 User Participant Testing ... 63

Chapter 1: **Introduction**

1.1 Problem Statement

Post-Covid and Inflation in the UK has had significant implications on the ways in which and how money is spent, particularly for lower social-economic groups. The rise in inflation has impacted energy bills and this has further contributed to the cost-of-living crisis currently being faced in the UK.

However, many individuals and households are unaware of the sources of the costs shown on their bill, particularly which appliances are using the most energy in relation to the cost on the bill. The absence of a tool and a need for a system that enables users to effectively track their energy consumption transcends into the knowingness of their energy usage, which in turn reducing consumer energy bills.

1.2 Background Context

In recent years, the cost of domestic energy (gas and electricity) has skyrocketed. Energy consumption surged dramatically, particularly after 2020, when the coronavirus pandemic hit the UK, as well as Russia's recent invasion of Ukraine, which disrupted natural gas delivery from Russia to the UK (Hui, 2022). As a result, these factors have triggered the high cost of energy, due to the rising demand and declining supply from Russia. It has fuelled inflation and limited people's ability to spend, contributing to the cost-of-living crisis currently faced in the UK. The cost of natural gas increased 96% in the UK that concluded in July of this year. In addition to, the cost of electricity increasing by 54%. It is projected that household energy usage will increase to above £4000 annually in January 2023 from the current capped price of £1971 (Cooban, 2022). This is a staggering increase that will raise the expense of living even further.

The main factor in this is that the rising cost of energy has had various impacts on households. 13% of households in England have fallen into the fuel poverty band. That is 3.16 million households that are unable to afford to heat their home to an adequate temperature. According to the to the Opinions and Lifestyle Survey (OPN) that was conducted in January 2022, 79% of people who have experienced a rising cost of living in the UK relate high energy bills as the cause (Office for National Statistics, 2022). Energy is essential to the economy, thus when energy prices rise, so do the prices of other goods like housing and food. As a result, low-income households feel constrained because they are no longer able to live balanced lives due to not being able to afford everything they require. This inability to fulfil one's wants, and needs has resulted in a decline in wellbeing and the development of health problems, such as stress, anxiety and other mental health problems. Individuals' happiness has declined from 34% to 31%, while the percentage of people who feel like their lives are worthwhile, satisfied with their lives, and satisfied with the things they do in life has decreased from 31% to 26%. (Scott, 2022). These patterns reflect the effects of the declining living standards due to rising cost of living in the UK.

1.3 Aim

The aim of this project is to solve the problem of consumers being unable to understand and have the knowledge of the sources of energy usage that reflect on their energy utility bill, given by providers like EDF. Household energy bills have increased by 54% in April 2022 and further increased by 90% in October.

My system aims to integrate functionalities and design components to address these problems in a single web application. By offering a system that enables consumers to properly monitor, manage, and track their gas and electricity consumption, along with a breakdown of the main appliances that are contributing to the high costs displayed on their energy bill.

There are not many systems or tools available which provide a breakdown/analysis of energy usage, the capability to monitor and examine home appliances, and the ability to determine which devices are being used the most in terms of the cost of the total utility bill. For instance, energy providers like EDF and British Gas only send a bill statement outlining the amount that is owed and are not always transparent and do not provide a thorough explanation or view of how they have arrived at the bill amount.

The main aim is to develop a web application that would enable the user to select their numerous household appliances, including how many hours per day or week they use that device. An automatic computation would determine the number of kWh used and provide an estimate of how much each device would cost based on the energy provider they are with and the corresponding tariff of that provider. This would be displayed in a breakdown with the system's overall estimate bill being compared to the actual bill provided by the energy supplier.

In order to avoid information overload, the application also aims to provide a straightforward, user-friendly interface for tracking and displaying trends of prior energy consumptions and forecasts via visual graphs. As a result, households which employ the various features within the application will enable the consumer to better track and control their appliance usage, giving them the information and understanding they need to lower their energy usage in the future.

The project will involve research to gain a better sense of the energy industry, and how the key providers in the electricity/gas supplying industry calculate utility bills with the use of standard tariffs and how many units of energy is consumed by households. The effect of inflation on energy prices and fluctuations over time will also be considered in this research.

The overall goal is to create a user-friendly web management system that offers consumers an effective energy-tracking web application that is both engaging and simple to use. As well as giving users important news updates from energy suppliers and government about any changes that might have an impact on them.

1.4 Objectives

Development/Programming Objectives:

- Design and develop a web-based application that provides users with the best tools for managing and tracking energy consumptions with added tips to reduce costs. Some of these will include:
    - Home Dashboard Screen which displays personalised overall energy consumption in a specific time interval, showing graphs of trends and latest news updates.
    - Users being able to add their different appliances which would calculate the cost based on the watts and tariffs supplied by the user.
    - Breakdown page of all appliances with the given quantities and consumption pet unit. Showing daily, weekly, monthly and yearly.
    - Providing tips and advice. The system will provide tips and advice to the user on how they could save on their energy costs.
    - The estimate cost calculating by the system would compare against their actual bill.

- To compare and contrast projected utility bill generated by the app to energy provider's bill.
- To implement an interface that is user friendly and navigate through the application easily.
- To implement the use of Django web framework to help with developing the application, both frontend and backend.

Evaluation/Testing Objectives:

- To perform user testing to conclude the effectiveness of the application.
- To validate if the application is complete and works as per the expectation of the users.
- For the system to provide users with the knowledge of how much energy usage and the costs of each household appliances to better aid with reducing utility bills.
- Several tests of the application conducted and whether fits all needs and requirements. Testing surveying purpose and proofing design.

Research Objectives:

- Conduct a literature review on existing systems and the potential benefits my system will bring upon from its use, through:
    - Understanding the current systems and the ways in which it helps their users. Evaluating their pros and cons.
    - Understanding the user needs and desired functionalities.
    - Understanding of relevant concepts such as measures of energy and how it is calculated by related providers.
- To examine and assess how different energy providers calculate on basis of how many units of energy individuals consume and how tariffs charge customers in different ways.
- To understand the reasons behind the surge in energy prices in the UK.

- To understand how individuals are being affected by energy prices and establish efforts being made by government and officials to help challenges faced by individuals.

1.5 Research Questions

1. How can you improve the users understanding of energy usage in their household?
2. How to different energy providers calculate the energy consumption by households?
3. What are the reasons for the significant increase in the cost of energy for households?
4. How is the cost-of-living crisis affecting households and individuals?
5. Can an energy tracking system be beneficial to users to reduce utility bills?

1.6 Project Milestones

1. In-depth literature review, with clear aims and objectives.
2. Following deadlines of task breakdown indicated by the Gant chart created.
3. List of functional and non-functional requirements.
4. Final report completed with all necessary elements included.
5. Fully functional web application with desired features.

Chapter 2: **Literature Review**

This chapter will help in laying the groundwork for the features, functional and non-functional requirements. The overarching goal of the literature research is to gain a thorough understanding and perspective of the energy industry as a whole, in order to determine what ideal computations and functionality would occur in the application's backend. Furthermore, conducting extensive research would help me to understand consumer concerns and how the current living cost crisis is affecting individuals, allowing me to tailor the application to their needs. Similar solutions will be sought and researched to fully understand user needs from an unbiased perspective.

2.1 Context to the Energy Industry

At the moment, reducing global energy consumption is a top priority for households. The building sector is recognised as a key contributor to climate change due to its massive energy requirements. Rising energy consumption is intrinsically tied to both population growth and climate change, particularly given the current dominance of non-renewable energy sources in the energy industry (Bürer et al., 2019). Despite accounting for 36% of worldwide energy consumption and 39% of CO2 emissions, the construction sector is predicted to contribute 13% of global GDP in the future years. According to Bürer et al., the United Kingdom faces severe competition for its energy resources, perhaps more so than other countries, by creating a market for leading global power stations as a conglomerate (2019). Since May 1999, individuals within the United Kingdom have had the freedom to select their energy power provider due to the diversity in several market providers, with the ease to choose various types of energy and power based on needs and preferences.

On the other hand, the price of electricity and energy has been affected by deregulation, which was noted by increased market supply by more providers, reducing the price of power. For example, the graphical representation below illustrates that demand for energy use increased in early January 2009 but decreased by July of the same year. This is due to various factors, such as increased energy consumer’s awareness of conservation, energy efficiency measures, improved home insulation, and many more.

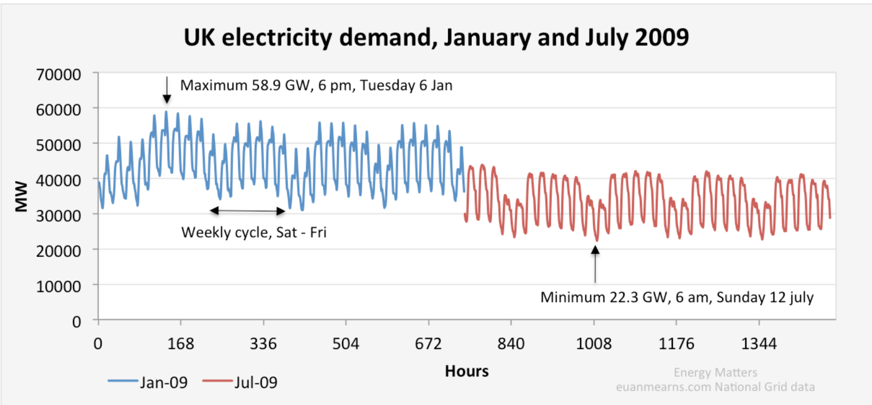


Figure 1 - UK Electricity Demand, Source: http://www.euanmearns.com/wp-content/uploads/2013/10/UK_electricity_demand.png

Companies from the UK, Germany and France dominate the British supply sector. However, market participants from other nations are present to examine the local workplace rivalry. Providers include: Southern and Scottish Energy, Scottish Power, RWE Npower, E. ON, British Gas, and EDF (Smith, 2022). These suppliers buy the electricity from the generating companies and supply it to households through transmission lines.

2.1.1 Energy Market and Trends

The major component of energy supply in the UK is electricity generation from fossil fuels, such as coal and gas, with some utilisation of renewable energy and imports, such as wind, solar, or hydro power plants. The electricity is supplied through the grid system to businesses, homes and users. Lastly, consumers obtain energy from suppliers through wholesale.

Energy Trends

UK, April to June 2022

Percentage change from Quarter 2 2021, primary energy basis

(mtoe basis)	Production	Imports	Exports	Demand
Total energy	+21%	+11%	+52%	0.0%
Coal	-48%	+37%	-25%	-19%
Primary oil	+10%	+3.3%	-2.4%	+16%
Petroleum products	+13%	+26%	+25%	+15%
Gas	+55%	+17%	+576%	-10%
Electricity	+19%	-68%	+568%	+19%

Figure 2 - Energy Trends, Source: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1107456/Energy_Trends_September_2022.pdf

Compared to the record-low energy output in the previous year, when maintenance had a detrimental effect on oil and gas production, the UK's energy output increased. Natural gas production increased by more than 50% while oil production increased by 10%. Low carbon energy also increased (Department for Business, 2022).

Warmer temperatures reduced demand and offset rising economic activity, resulting in a 0.2% reduction in total final energy consumption compared to the second quarter of 2021. The usage of transportation increased by 23%, with the consumption of petrol and diesel almost reverting to pre-pandemic levels. The warmer weather and a reduction in time spent working from home caused a 28% decline in domestic consumption (Department for Business, 2022).

2.1.2 Demand and Supply

The electrical market involves power generation, transmission, distribution and delivery and the interaction between energy distributions. Despite the fact that distribution and supply are still restricted, licences are nevertheless available to any qualified potential supplier. Out of the 60 energy suppliers operating in the UK, OVO, Centrica, Npower, Scottish Power, E.ON, and EDF are among the top six (Malinauskaite et al., 2019).

The supplier can determine the cost of electricity by breaking down the customer's ties with the incumbent provider after calculating the cost of wholesale shipping and signing the power contract. Although, the market provides ancillary services like meter readings, the supplier is ultimately responsible for invoicing and compiling a single bill for all vertical points. This will be discussed in more detail later.

The marginal changes in supply and demand by the restricted supply on non-renewable energy led to price imbalances as the demand for such sources rose (Malinauskaite et al.,2019). Due to the pandemic and the current war between Russia and Ukraine, low-cost global energy has restrictive availability due to the increase in energy prices to tackle climate change. Discussion of these issues frequently results in suggestions for employing alternative energy sources. Moreover, the supply-demand mismatch is widened by alternative energy arrangements, though, which require greater energy expenditure. Despite what might have felt like an excessive amount of change, technological systems of energy production and consumption measures were enough to combat these issues.

2.1.3 Green Energy

Even though the Big Six produce more than three-quarters of all energy in the UK, it would be a mistake to ignore energy from smaller sources. Their decreased operating expenses allow them to offer superior customer service, "green" energy, and competitive pricing.

bulb	Bulb Energy is a fast-growing energy supplier with the aim of offering affordable, renewable energy to homes and businesses.
ecotricity	Ecotricity is a British energy supplier specialising in providing green energy to customers through its wind power portfolio.
good energy	Good Energy has been in operation for more than 20 years aiming to power a cleaner, greener world.
F&S	F&S Energy was founded in 2011 with the aim of increasing competition in the green energy market.
geuk	Green Energy UK (also known as GEUK) is an independent energy supplier selling 100% Ofgem certified renewable electricity and gas.
octopusenergy	Octopus offers all green energy tariffs to domestic and business customers with the aim of providing

Figure 3 - Green Energy Suppliers, Source: https://britishbusinessenergy.co.uk/suppliers/

There has been an uptick in the availability of renewable commercial energy sources in recent years. Using a renewable energy tariff is a simply way for businesses to demonstrate their concern about the environment (Bürer et al., 2019). Conversion of green energy tariff, gas and electricity service will not be interrupted. The energy company will obtain renewable electricity or finance environmentally conscious initiatives to meet consumer needs. Due to time constraints, the main aspect that my project will focus on would be energy providers that use fossil fuels as their source of energy, as it is the most common companies adopted by UK households.

2.2 Living Cost Crisis in the UK

The rising cost of living in the UK has had different impacts on individuals and thus various writers, websites, journals, magazines and online media have diverse views on inflation, particularly with regard to energy costs. Since 2021, the UK has seen a major decline in disposable income; as a result, the high inflation rate depletes a significant portion of people's income through costly living and high tax rates. Any salary increase has little effect on the current state of the economy since it cannot keep up with the rise in the cost of living. There is a significant increase in energy prices, especially for domestic gas, petrol, and other sources of energy.

According to an economic report from the New Economics Foundation journal, the economic situation has been affecting the United Kingdom before the Covid-19 pandemic, making individuals unable to access resources to fund their needs, but the outbreak worsened the situation (Blundell et al., 2022). Economic downturn affects more than 32% of the UK's total population even without changes in government policy, which has seen more than 21 million people living below the social standards, since their income remained the same while prices of common household commodities increased. The pandemic stagnated the UK economy through slow growth in earnings and low working-age benefits that could not cover the rising housing expenses, foodstuffs, and other essentials. In addition to this, economic inflation saw the lower-economic households suffering more in the UK, and social classes and inequality prevailed as a result of lockdowns that prevented the working class from earning their living which in turn also caused others to lose their jobs. The UK social security's efforts to avert the economic crisis were less successful because there was a lack of steady income, thus, some households were forced to rely on relief and welfare services to cater to their needs. (Arnold, 2021). According to this publication, the Covid-19 pandemic worsened the economy's condition and existing inflation rates, making social security and other measures ineffective well before the pandemic outbreak.

Furthermore, based on economic remarks from an article written by 'Institute of Government Analysis' on the living cost crisis in the UK, the current economic fluctuations like tax rate increment and other benefits have increased the cost of resources. The UK government had previously stated their intended measures to support households that were unable to afford the cost of energy and those struggling to make their economic ends meet. The household support policies would protect the people from the inflation crisis that consequently increase energy bills, thus reduction of these costs would likely lower the living cost. The energy cost regulation policies would however apply to the lower-income and poorer households, while others would adhere to the inflated prices and tax requirements (OFBR, 2022). According to the Bank of England, energy cost inflations would continue to increase steadily not only in 2022 but for the next two years with the following expected inflation target illustrated in the graph below.

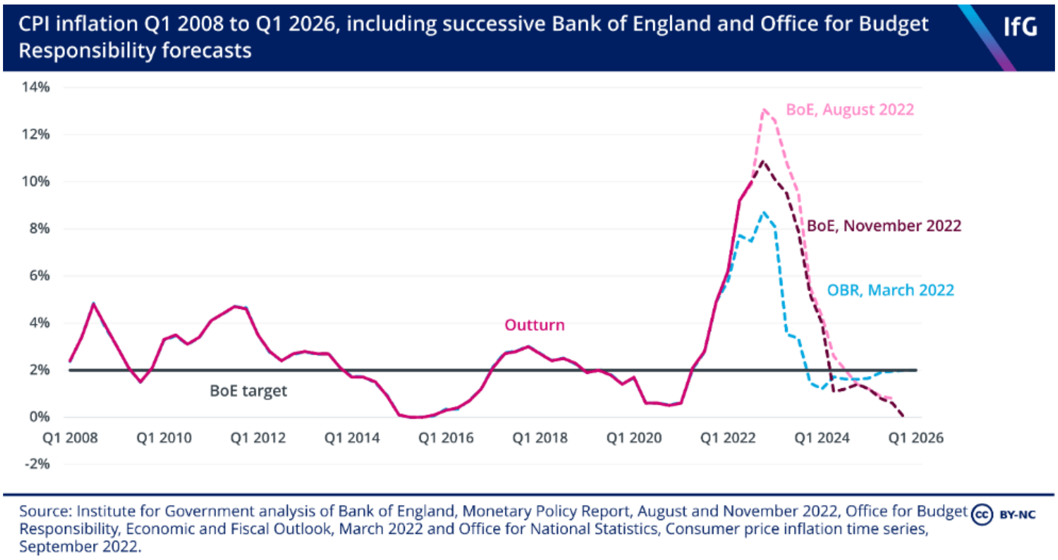


Figure 4 - CPI Inflation

According to this journal, gas inflation in the United Kingdom has been caused by various economic, social, and political issues both within and outside the country, particularly from gas suppliers. Since mid-2021, there have been an increased in demand for gas with unstable supply, especially from Russia, a major worldwide supplier. Due to the conflict in Ukraine, the majority of storage is depleted, and the difficulty to import Liquified Natural Gas into the UK has resulted in frequent disruptions (OFBR, 2022).

There are significant supply chain challenges globally which increase the transportation cost to the United Kingdom from outside suppliers, hence the result of the spiked prices. Increased energy cost has affected the overall economy because gas is the main source that the UK utilises to create other forms of energy for both household use and economic purposes. Although only 13% of the UK's fuels imports come from Russia, any disruptions in the supply chains will influence inflation in the EU as the two regions lie within similar energy markets. The government's efforts to shield consumers from economic inflation and provide alternatives to maintain living standards face the challenge of constant increase in energy costs which consumes more wages or pension benefits from individuals (Monetary Policy Report, 2022).

2.2.1 Comparison

Both articles acknowledge the impacts of the Covid-19 pandemic on the UK's economy, but they also agree to the fact that living cost had been rising even before the pandemic outbreak (Blundell et al., 2022). With instable economic opportunities and benefits for households, the governments intentions to help the lower-economic households may not be practical because higher inflation will still affect the unemployed and low-income groups (Graby et al., 2019). The high cost of living is affecting most countries because the pandemic slowed the productivity and supply of energy resources, hence the economy deteriorated. In 2021, tax announcements in the UK were the highest due to the rates increasing by 1.25% from the previous year, indicating the possible inflation in the subsequent years as the country seeks to recover from the economic impacts of the pandemic (Monetary Policy Report, 2022).

The influence of high living expenses on different social groups, who depend on social security programs to survive the economy, is acknowledged by the two sources. Therefore, establishing laws to protect the poor households from the effects of inflation will lessen social difficulties from lifestyle classes. Comparable to rich homes in the UK having alternative ways to absorb the increasing energy expenses. The Bank of England emphasizes that everyone should receive the previously mentioned 4% salary increase in order to improve their living standards. Despite salary increases, household income cannot keep pace with the economy's fluctuations since food and energy prices are rising faster than household income.

In addition, according to the UK chancellor, tax rates in 2023 are expected to increase by more than 5% due to more government expenditures on the nation's activities. Generally, UK's recovery strategies from the post-pandemic economy and inflation of living cost indicate that the country requires several years to regain the initial stability and improve living standards of households. The inflation rates could potentially rise further depending on both internal and external economic factors, and the estimates for inflation simply reflect the UK government's expectations for the economy. Social security for everyone in this unstable economy would be ensured by more practical economic policies that aim to raise living conditions for UK citizens (HM Treasury, 2022).

2.2.2 Impacts of Living Cost on UK households

The power of UK citizens for wage increases has not been improved by the rising inflation rate because the extra leverage and income individuals receive from wage increases are being offset by rising inflation-related expenses. Governmental policies are ineffective to support UK citizens with the challenges of increasing living costs. The UK's rate of inflation is higher than the average pay increase rate. With the struggle against the rising rate of inflation, the cost-of-living problem in the UK is continuously stoking consumers. Consumers are unable to reduce the impact of the cost-of-living crisis that is having a negative influence on them. The combined shocks that are being delivered from the UK's price spiral are continuously impacting and causing more global concerns. There has been a large decline in household incomes, which has resulted in a greater cost of living, particularly for energy, food, and taxes, with a worsening situation for the public. This is despite the anticipated increase in interest rates to one person, indicating the highest level in 13 years (The Newsroom, 2022).

There are numerous UK households whose primary source of income comes from government programs, such as obtaining aged benefits or state pension, that may see a significant increase annually. The rate of state benefits and pensions increased by 3.1% in 2022 as a result of consumer needs as the household disposable cash rates are used to calculate the index inflation price rate. According to the findings of economic and psychological specialists, the cost-of-living problem is affecting people's psychological and physical health since it is depriving them of their independence by putting financial strain on them (HM Treasury, 2022).

According to a case study regarding Amazon workers protesting over inadequate pay in the United Kingdom. There were speculations of a £1-per-hour raise at the Tilbury distribution centre in southeast England, representing a 9% boost for many employees. However, just 35 pence, which equates to 3% was shown for workers. Amazon employees are not the first in the UK's IT sector to protest about salary increases that they believe do not keep pace with growing inflation and energy prices. After a pay increase of 3 to 8% was announced in late July, thousands of BT telecom employees went on strike (Meaker, 2022). Numerous UK Amazon facilities, according to Steve Garelick, a regional organiser for GMB, have never had a walkout. "This is the first time ever that the employees have taken coordinated action," he remarked. Garelick claimed that Amazon's failure to adequately address employees' concerns about growing living costs brought on by inflation and an increase in interest rates from the Bank of England was the cause of the rebellion (Meaker, 2022).

In relation to this, the government has used a variety of economic techniques to assist UK people in managing their energy bills. For example, the government has announced a £15 billion package of assistance for rising energy bills for over 28 million homes, equating to up to £550 per household. This includes: The Energy Bill Support Scheme, which is a £400 payment for all home energy consumers that will appear as a credit from energy providers which commenced in October 2022. The Council Tax Rebate is a non-repayable £150 rebate offered to households in council tax categories A through D. The Discretionary Fund, worth £144 million, is available to billing authorities to assist any home in need, regardless of council tax category. The government is aware that billing agencies may also want to assist other individuals who are not qualified under the conditions of the core program or to make carefully targeted "top-up" payments to the neediest households in bands A to D. Consequently, it will provide a portion of the £144 million Discretionary Fund to each billing authority (citizensadvice, 2022).

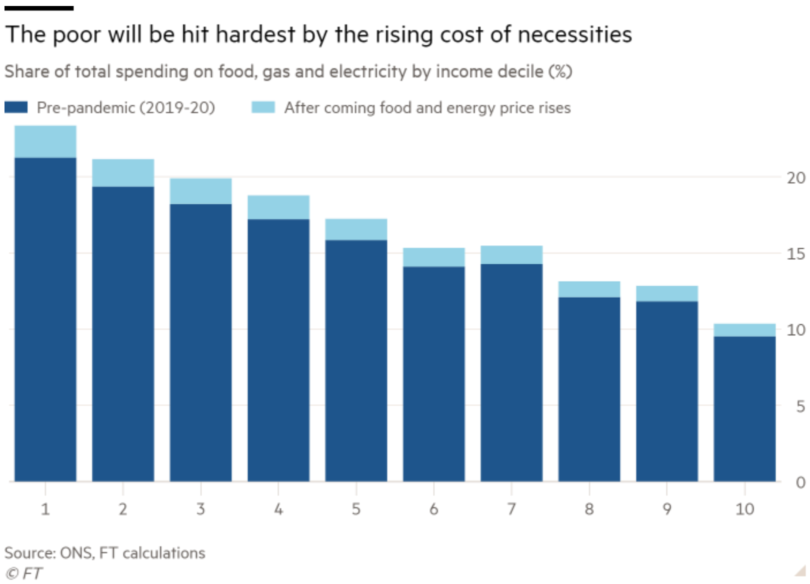


Figure 5 - Share of total spending on essentials

2.2.3 Tracking Energy Consumption

According to the UK government, homes with central gas heating accounted for the majority (48%) of the variation in gas use. The most significant characteristics that contributed to an increase in gas consumption were ‘rising floor area, declining energy efficiency, increasing heating hours, and increasing household size’. The number of wet and cold appliances and the size of the home were the two most significant characteristics that contributed to an increase in electricity usage (EFUS pg27, 2017). Other influences included the kind of residence and the use of other appliances.

Software for energy monitoring offers crucial information useful in managing and conserving energy in the future. Without energy tracking, it is difficult to identify the areas of high consumption in your household and the necessary steps in order to fix them. The UK government developed schemes as mentioned previously for households as a result of the current energy crisis. In the short term, this assists households, but in the long run, individuals need to start thinking about their own consumption to minimise their utility bills (Hark, 2022).

Energy management is crucial because it ensures that energy measurements are done right and the utility bill owed to the provider, gives you a total amount of usage and cost, only split down by electricity or gas. However, it is unable to explicitly explain the source of the usage or the rationale behind the fluctuating monthly expenditures. Thus, failing to give individuals the necessary data they requirement to make data-driven decisions, in regard to their consumption (Ofgem, 2022).

The ability to track, determines energy problems such as the usage of inefficient devices, or the ability to detect older devices that may have low energy efficiency compared to newer models. Additionally, pricing per unit would help individuals to better understand the cost per unit of energy by keeping track of the different devices within the household, which in turn aids cutting power factor costs. When done in real-time, energy monitoring is even more efficient, but due to time constraints, real-time rolling data would not be implemented in the development of my application.

2.2.4 Energy Providers and their Rates

Mostly energy prices depend on the purchase plan for households such as wholesale, shared bills, and other weekly, monthly or annual bills. The cost of gas and electricity on the wholesale market has the greatest impact on average energy bills; as a result of the gradual increase in wholesale prices since last August, it currently represents just under 50% of bills, or £1,077 of the average capped bill. Energy costs are subject to VAT, which is set at 5% and amounts to £98 annually for the average household (up from £61 before April) or more than £2.1 billion overall. Before the price cap announcement, proposals for the government to temporarily reduce VAT in order to provide more assistance to struggling families were made in response to the Treasury windfall, but this has so far been rejected. Policy expenses total £153, or 8%, of the typical home cap payment. In addition to the renewable’s obligation, which requires suppliers to source renewable electricity, and the energy company obligation scheme, which pays to upgrade home insulation for low-income households, this also covers the cost of the warm homes discount, which will pay vulnerable customers £150 annually next winter and be expanded to include another 800,000 homes. The feed-in-tariff payments that were provided to those who paid to have solar panels installed on their roofs and that were contractually required to be made are also covered by policy costs (Brignall, 2022).

There are various energy producers and suppliers in the UK for both renewable and non-renewable energy, as mentioned in the beginning of this section. One of the newer energy providers in Britain is The Octopus Energy, which was established in 2016. It invests energy generation and uses only renewable energy for its electricity. Additionally, it provides carbon-offset gas on a few of its tariffs. Moreover, it provides carbon-offset gas on a few of its tariffs, making it an obvious winner in the eyes of its clients (Bunney, 2021). The company utilises the agile price algorithm:

min(2.20 x W + P, 95)

In this equation,

- **2.20** is a coefficient that includes our **distribution costs,** which varies based on where you are in Britain;
- **W** is the **wholesale cost of electricity** for that period in ***pence per kilowatt-hour*** (p/kWh);
- **P** is the **peak-time premium**, which ranges from 11 - 14 based on where you are, and is only applied between 4pm and 7pm.
- **95** is chosen to ensure the price is capped at 100p/kWh once VAT is added.

Figure 6 - Agile Price Algorithm for Octopus Energy, Source: https://octopus.energy/blog/agile-pricing-explained/

This agile calculation is one example of how Octopus Energy calculates utility bill for its consumers.

The usage cost of household appliances will be determined using the following formula in order for my application to estimate how the device consumption will affect the overall cost of their electricity bill:

1. Multiply the appliance's wattage by the number of hours the device is used.
2. Convert W to kW = Dividing by 1000
3. Multiply by providers' kWh (kilowatt-hours) rate

(Novi Electrics, 2017)

Due to different energy providers within the UK having different kWh rates and tariffs, the calculated estimate costs would be different for users even with the same device and same amount of usage.

The unit rate is how much you pay for every kWh of electricity and gas you use, in comparison to a standing charge which is a small daily charge you pay for your electricity and gas supply, even if you don't use any energy (EDF, 2022). The table below shows the energy providers that would be used within my application, displaying their energy per kwh costs with their standard charge per day.

Energy Provider	Electricity (per kWh)	Gas (per kWh)	Standard Charge (p/day)
EDF	34.81p	10.789p	Electricity: 38.28p Gas: 37.51p
British Gas	34.037p	10.330p	Electricity: 46.356p Gas: 28.485p
EON	22.31p	4.40p	Electricity: 29.48p Gas: 30.78p

On the other hand, calculating the gas usage is slightly different as you will not be able to estimate for each individual device. The value from the gas metre, which is typically in cubic metres, is taken first (m^3). To determine the actual amount of energy used, the measurement must then be converted into kWhs (kilowatt-hours). To achieve this, multiply the amount by the provider's price per kWh after dividing it by 3.6 to convert it to kWh. The application would display this to the user, provide data on their usage for that month to forecast for subsequent months, as well as offer guidance on how to lower their energy consumption (Lepage, 2020).

2.3 Existing Systems

Hark Energy Tool

Hark is an energy management system, which over the years has seen its organisation establish awards in Industry IOT and Energy Analytics (Hark, 2022). The organisation behind the system is based in Leeds. Their vision is that efficiency and cost reduction should be connected to, monitored, and automatically identified for all user types. 3 pillars of Harks mantra include: Improve Efficiency, Maximise Yield, Reduce Waste.

In addition to the organisation is also known for other innovative technology in its subscription cloud platform. The Hark Platform among others helps in alerting asset operators and energy managers on abnormalities in their estate. It also assists in informing actions such as regulatory compliance, energy management, asset control, and preventative maintenance (Hark, 2022).

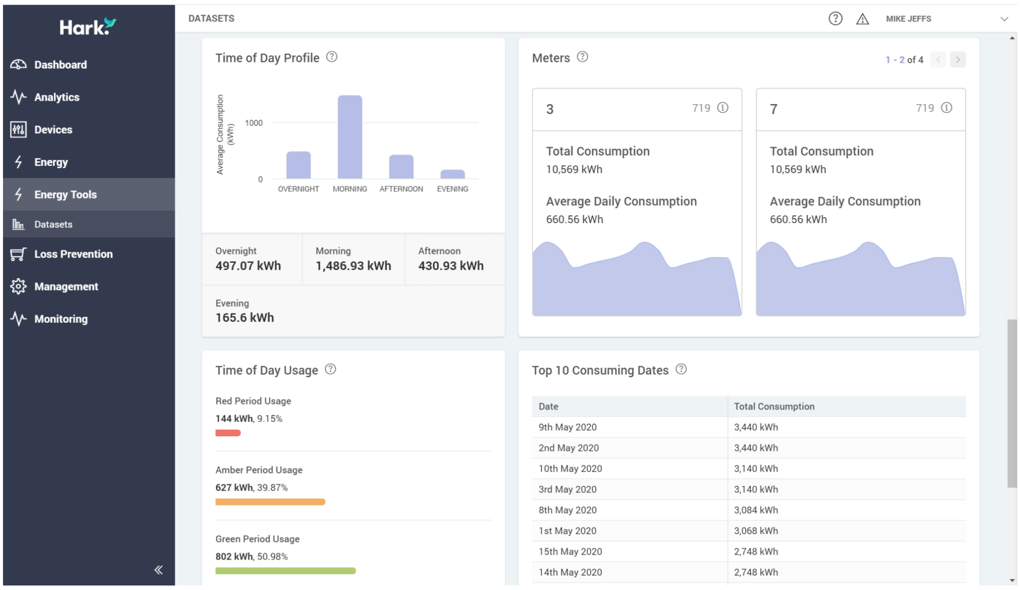


Figure 7 - Hark Dashboard, Source: https://harksys.com/products/energy-tools

<table>
<tr><th colspan="2"><u>Key functionalities</u></th></tr>
<tr><td colspan="2">• Real-time alerts
• Anomaly detection – Use of machine learning algorithms to understand energy profiles
• Instant notifications – Issued to maintenance team of a asset malfunction
• Peak load analysis
• Predictive maintenance</td></tr>
<tr><th>Pros</th><th>Cons</th></tr>
<tr><td>• Maximize yield of device usage in relation to energy consumed
• Reduce waste with the features, aiming for carbon neutrality
• Improve efficiency, reduce costs & carbon emissions</td><td>• Expensive
• Complex features – hard to understand for standard users
• Support ticket issues</td></tr>
</table>

Smappee

Smappee was created in 2012, it currently has 10,000+ users and is available in Belgium and the UK. It is a smart application system for residential energy efficiency that is available on smart phones through the play and app store. It offers energy data and smart control for homes and EVs with the use of hardware as well as a mobile app.

Smappee tracks everyday habits and provides users with options to manage their environmental footprint. It offers real-time energy insights and control options that offer businesses and consumers a range of smart energy management tools. Smappee provides a solution that works regardless of the electrical installation, difficulty, or circumstances. From smart EV charging and self-consumption of renewable energy to major energy and cost savings for enterprises with one or more locations (Romatech, 2019).

Figure 8 – Smappee, Source: https://www.smappee.com/infinity/

<table>
<tr><th colspan="2"><u>Key functionalities</u></th></tr>
<tr><td colspan="2">• Installing a Smappee Infinity system allows households and companies to monitor the energy they consume (and produce) in real-time, enabling them to take educated steps to minimize energy expenditures and optimize solar energy usage.
• A competent electrician can rapidly and effortlessly install the Smappee Infinity system, which delivers plug-and-play power monitoring and smart control technology.
• The Infinity solution may be set for both residential and commercial applications, with current control choices ranging from 50A to +4,000A.</td></tr>
<tr><th>Pros</th><th>Cons</th></tr>
<tr><td>• Provides a 15% decrease in individual energy usage and energy efficiency.
• Provides a 15% decrease in individual energy usage and energy efficiency.
• Smappee is a one-device solution for tracking whole household energy usage.
• Offers a smartphone app for on-the-go processing and assessment, as well as 'Comfort Plugs,' which often include a substitute for regular electrical plugs to monitor devices constantly and plan their utilization.</td><td>• Unavailable in other markets such as the United States
• Slow responses
• Hardware not reliable detecting appliances.
• Software is not well designed.</td></tr>
</table>

Wattics

Wattics is a cloud-based energy data analytics tool that allows users to manage energy use online efficiently, via the analyse of data. It connects business data, with the use of sensors in providing necessary analytics. The software was made for the purpose of energy efficiency. Key features include trend analytics, energy monitoring, measurement and verification (M&V), cost allocation analysis, wastage analysis and projected savings (SoftwareAdvice, 2022). Wattics assists users in managing activities such as user behaviours, saving projects, performance contracts and energy preservation. The readings and meters help in the identification of high usage and stress periods. In addition to this it is also insightful in identifying when energy is being misused, and thus helps in savings and in efforts towards sustainability in the long-run (SoftwareAdvice, 2022).

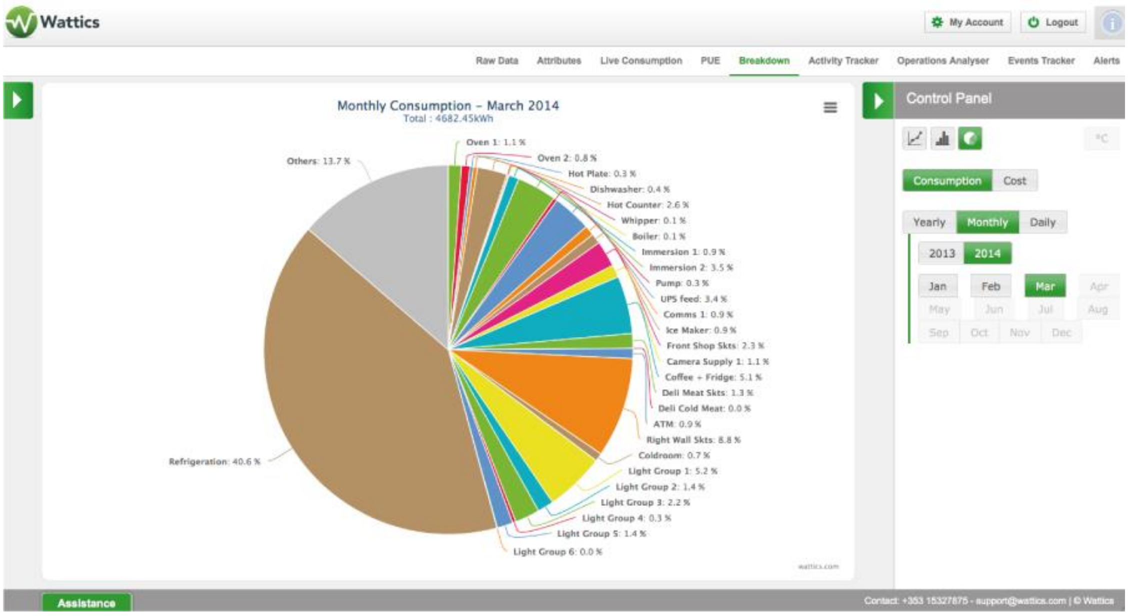


Figure 9 – Wattics, Source: https://www.wattics.com/

<table>
<tr><th colspan="2"><u>Key functionalities</u></th></tr>
<tr><td colspan="2">• Performance metrics
• Power quality and operation wastage analysis
• Verification and measurement
• Formula composer
• Real-time analytics
• Compares and analyses consumption trends/data</td></tr>
<tr><th>Pros</th><th>Cons</th></tr>
<tr><td>• Universal data integration
• Machine learning tool to find anomalies and receive notifications
• Visualisation of energy data & functions with various brands of meters.
• Moderately priced to use and provides a custom pricing model for the customers to meet their needs</td><td>• Limited to cloud hosting services
• Additional costs for custom integrations</td></tr>
</table>

2.4 Proposed Solution

This aim of this project is to develop an application while taking into account the areas to improve from existing applications. The current systems analysed primarily offer energy management and tracking systems for businesses and organisations. This is undesirable because the functionality offered by these existing systems are expensive and quite complex for the typical person. Due to my application being focused on solely individuals, households and not businesses, it would be tailored with less complexity and ease of use for the individual themselves, in addition to being a free web application.

From the research gathered, there are not many systems available providing individual with an energy tracking. One system discussed above was Smappee. However, with an overall rating of 3 stars on the play store and the required use of their hardware for the app to function, it does not meet the satisfaction and needs of the user.

Especially during the living cost crisis, many individuals are feeling anxious. In particular, low-income households may struggle to pay off their bills. The purposed solution for this system to help and aid users to see which household appliances are using the most energy – given their energy provider, they'll be able to better understand and gather the knowledge from the application, which in turn assists on reducing energy bills in the future. For instance, by stopping the use of the appliance that is causing a spike in their utility bill, or even by purchasing in devices with better energy efficiency.

The project will be developed by using features that would calculate an overall bill and each device energy costs in relation to their usage with the computation's formula mentioned in chapter 2. In addition, my application would provide frequent tips and forecasts and graph visualisation of previous months consumption – so individuals would be able to notice a significant difference in managing their energy usage in consequent months.

Moreover, this does not resolve the problem of the energy crisis experienced in the UK, which is beyond the control of consumers, but with a system to help aid and track energy usage will be mindful the users.

Chapter 3: **Design and Analysis**

3.1 Requirements Analysis

The literature review conducted has provided valuable insights into the desired operation and specifications of the system, taking into account existing systems. These insights have informed the identification of key features for my application, which are documented below as both functional and non-functional requirements. These requirements serve as a roadmap for the implementation and latter stages of the project, ensuring that the application aligns with the stated aims and objectives.

3.1.1 Functional Requirements

Functional requirements are a set of specific, tangible, and measurable functions and capabilities that a system must possess to fulfil its intended purpose. They outline what the system must do, and serve as the basis for determining whether a system is fit for its intended use. The table below shows the key functionalities for this project.

ID	Functional Requirements
F1	Users can create an account via signing up with a unique username.
F2	Users can login to the system via their username and password.
F3	Users can logout of the system.
F4	Users can edit and change their details and add a profile image.
F5	Users can add appliances and track energy usage and costs over time.
F6	Users can set usage goals and track progress towards reaching those goals.
F7	Users must set what energy provider they are with.
F8	Users will be able to view energy usage, which includes both electricity and gas and their usage history over time.
F9	Users can chat to other users on the application using forums feature.
F10	Users can view tips and advice on how to reduce energy usage and costs.
F11	Users can view and compare the estimated bill generated by the system to their actual energy provider’s bill.
F12	Users can view forecasts of future estimated bill in regard to their past energy consumption.

F13	Users can view which household appliance is using the most energy in terms of electricity or gas.
F14	The system will calculate the estimated daily, weekly, and monthly energy bills based on the inputted energy usage data.
F15	The system will generate energy usage data in the form of graphs and charts.

3.1.2 Non-functional Requirements

The below non-functional requirements are those that specify constraints or characteristics of a system that are not related to a specific function or behaviour, but instead describe the general system qualities and characteristics.

ID	Non-Functional Requirements	Type
NF1	The application will be compatible to run on both mobile and web browsers.	Platform Compatibility
NF2	The application will protect the privacy and security of user's data, such as password and energy usage.	Confidentiality
NF3	The application must be reliable and available to use at all times, with minimal downtime for maintenance and updates.	Availability/Reliability
NF4	The application should compute calculations and achieve desired response given an input into the system.	Performance/Response Time
NF5	The system must have a user-friendly interface that is easy to navigate and use.	User interface
NF6	New data and updates for existing features should be reviewed frequently.	Extensibility
NF7	The system must be able to accommodate increasing numbers of users and data without a significant decrease in performance.	Scalability

3.2 Use Cases

This section provides the use case diagram and some use case descriptions of the application. It illustrates the relationship and interaction between the system and the actors, as well as how the user will make use of the different functions.

3.2.1 Use Case Diagram

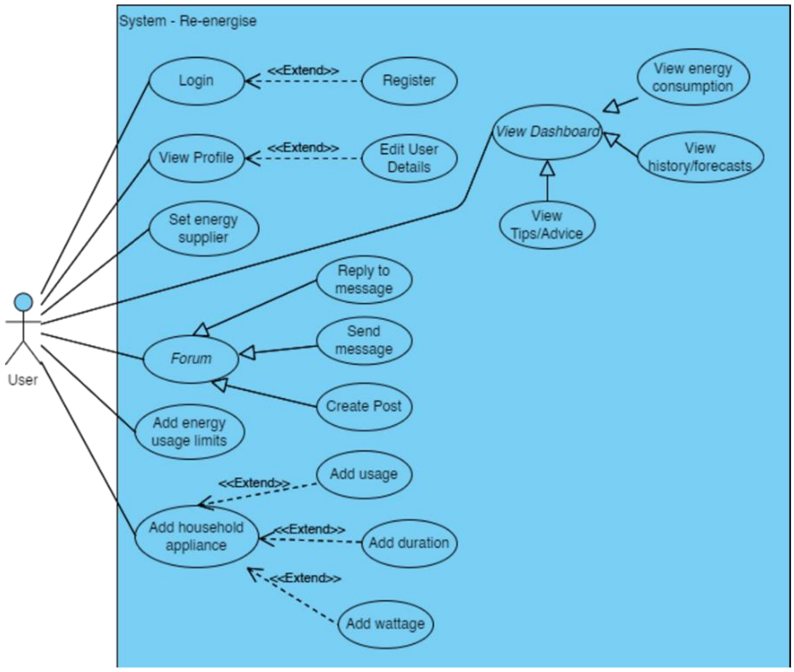


Figure 10 - Use Case Diagram

3.2.2 Use Case Descriptions

Use Case	Add household appliance
Actor	User
Pre-condition	The user needs to be registered in the system and logged into their account
Basic Path	1. User navigates to the 'appliance' page. 2. They click the button to add a new household appliance. 3. The user selects the electricity appliance option. 4. The user clicks the dropdown and selects their household appliance. 5. Inputs their usage, duration, and the wattage of the item. 6. User clicks submit and close. 7. New appliance added to database and updates the page to show the representation of the new appliance being added.
Post-condition	New household appliance added to the database and representation of the changes reflected on the page

Use Case	Create forum post
Actor	User
Pre-condition	The user needs to be registered in the system and logged into their account
Basic Path	1. User navigates to the ‘forum page. 2. They click the button to create a new post. 3. The user inputs the title and body of the post. 4. The user clicks the ‘submit’ button. 5. New post added to database and updates the page to show the representation of the new post being added.
Post-condition	The new user’s post will be displayed on the page, where other users can reply and add to the message post

Use Case	Add energy limit
Actor	User
Pre-condition	The user needs to be registered in the system and logged into their account
Basic Path	1. User navigates to the ‘limit’ page. 2. They click the button to create a new limit. 3. The user selects their specific electricity/gas appliance to be the goal. 4. The user selects the daily duration limit for that specific appliance. 5. User clicks submit and close. 6. New limit is added to database and updates the page to show the representation of the user’s limit.
Post-condition	The user's limit progress will be monitored by adding a new limit to the previous limit list, to determine if the appliance usage is exceeding the set limit.

3.3 SDLC - Development Environment

Developing the project as a web application is a practical choice due to several advantages. Firstly, web applications are highly accessible, allowing users to access the application from anywhere with an internet connection. Moreover, developing and maintaining websites is cost-effective compared to other types of applications, as ha websites are highly scalable and can accommodate growing numbers of users and data. Additionally, websites provide a secure platform for storing and managing sensitive information with proper security measures in place. Therefore, developing a website for the energy management application provides users with a convenient, accessible, and secure solution.

I will be using Visual Studio Code as my integrated development environment (IDE) for this project. Visual Studio Code is a highly customisable and feature-rich code editor that offers a smooth and streamlined coding experience. Additionally, I will be utilising GitHub for version control. The GitHub Desktop app will be used to manage my repository, allowing me to easily keep track of changes, as well as being able to access and work on my application from multiple devices. (https://github.com/Abid-M/Energy-Management-System)

3.3.1 System Architecture - Django

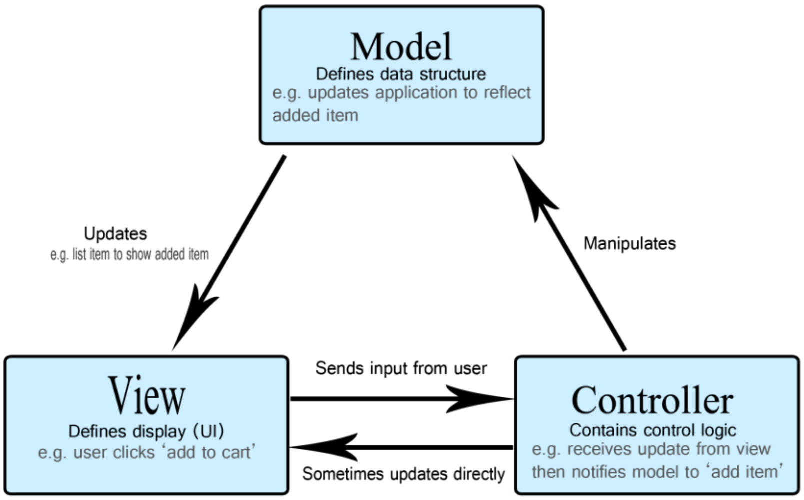


Figure 11 - MVC, Source: https://developer.mozilla.org/en-US/docs/Glossary/MVC

The Model-View-Controller (MVC) design pattern is a widely used software architectural pattern that separates an application into three main components: the model, the view, and the controller. The model component is responsible for managing the data of the application, such as retrieving and storing data in a database. The view component is responsible for presenting the data to the user, typically in the form of a graphical user interface. The controller component is responsible for receiving user input and updating the model and view components accordingly.

In Django, the MVC design pattern is implemented as the Model-View-Template (MVT) pattern. The model component is represented by Django's models, which manage the data of the application. The template component, which takes the place of the view component, is represented by Django's templates, used to present the data to the user. Finally, the controller component is represented by Django's views, which receive user input, update the model, and render the template. The MVT pattern in Django provides a clear separation of responsibilities within an application, making it easier to develop, maintain, and test.

3.3.2 Required Tools & Knowledge

1. Proficiency in HTML, CSS (including Bootstrap), and JavaScript.
2. Knowing and understanding python Django and how it operates, to develop the web application.
3. Identifying different testing techniques, including unit and integration testing.
4. Backend database server, PostgreSQL needs to host the web application.

3.4 Initial UI Screen Designs

As a means of laying the foundation for further development and refinement throughout the software implementation process, this section offers an outline of fundamental user interface design concepts. The designs were fashioned in such a way as to satisfy specific application requirements while maintaining their usefulness during future iterations of development. Of particular importance are two key screens highlighted within initial sketches: firstly, there is the homepage that will be visible to users upon loading the website; secondly, once they log-in it would redirect them with access to a dashboard screen where the user will have an overview of the application.

Figure 12 – UI Homepage Design

The Re-Energise application's homepage design aims to establish a unique identity through the prominent display of its name and logo, while also providing users with outlining the purpose as an energy management system. The strategically positioned login and registration buttons aim to simplify user access for an enhanced experience when using our platform. Overall, the design aims to facilitate user engagement by utilising an intuitive interface that enables smooth and easy navigation throughout the application.

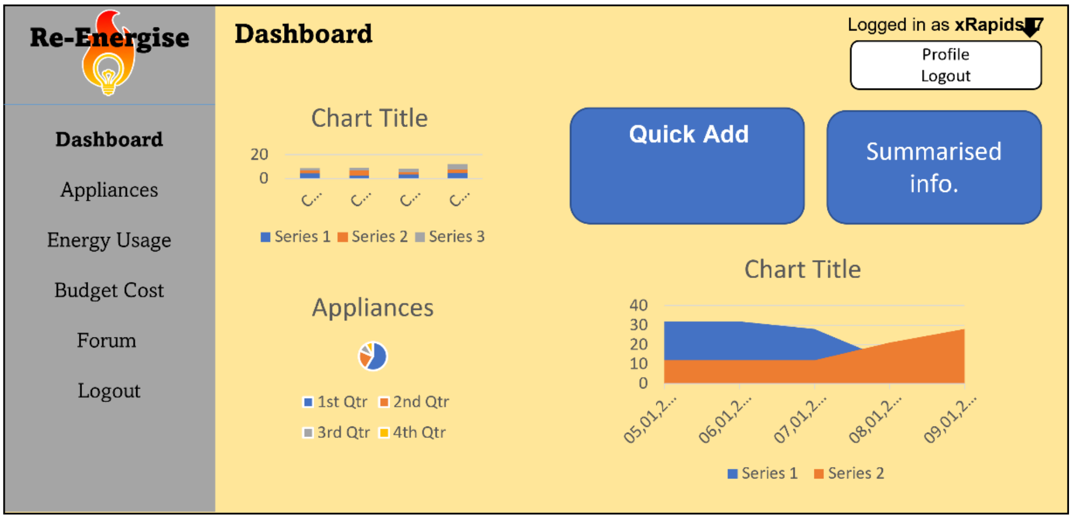


Figure 11 - UI Dashboard Design

The dashboard design for the Re-Energise application features a navigation bar that enables users to access different pages easily. It also displays the name of the currently logged-in user, providing a personalised touch to the interface. The dashboard itself provides an overview of important information from other pages in the application, such as charts, graphs, and quick add options. Overall, the design aims to present data in a clear and organised manner, providing users with an easy-to-understand view of the system's features.

While the two screen designs shown in this section provide an initial idea of the user interface, the final results will display the complete view of the application. In addition to further details on the design of other screens will be provided later in the report.

Chapter 4: **Implementation**

4.1 Platform and Technologies Rationale

4.1.1 Django & Python

The application will be developed using the Django web development framework in Python. As a framework, Django offers a range of tools and libraries that make it ideal for efficient development, providing strong security features and scalability. This makes it a suitable choice for building dynamic websites and web applications that require a robust and flexible foundation.

Some of the benefits of using Django include:

- *Rapid development:* Django provides a range of pre-built components and libraries, which allows for a fast and efficient development of web applications.

- *Security features:* Django includes built-in security features such as protection against cross-site scripting (XSS) and cross-site request forgery (CSRF) attacks.

- *Scalability:* Django's modular architecture and support for multiple databases make it easy to scale applications as they grow.

- *Built-in administration panel:* Django includes an insightful administrative interface, which can be used to manage the content and functionality of web applications.

4.1.2 Bootstrap

Bootstrap.css is a widely used front-end framework that provides a range of pre-built components, styles, and tools that allow for the creation of dynamic and versatile websites. It is particularly useful in constructing various visual aspects of websites such as buttons, layouts, and forms when integrated with Django templates. This integration not only saves time but also enhances the efficiency of the website development process.

4.1.3 PostgreSQL Database

In consideration of the application's needs, PostgreSQL will be used as the database management system for the project. PostgreSQL is a robust and scalable relational database that can handle the expected amount of data and users for this energy management application. Additionally, PostgreSQL is integrated with Django, which handles the low-level SQL language, used to interact with the database, abstracting away the complexities of the database design and allowing for a more simplified, efficient, and secure development process. With its reliable performance and compatibility with Django, PostgreSQL is a suitable option for the efficient management of data and user information on the application's website.

Likewise, incorporating Bootstrap.css into my energy management program offers numerous benefits. Firstly, it helps to maintain consistency and professionalism throughout the entire website, as many pre-built components and stylesheets can be used rather than building everything from scratch. Moreover, it enables the website to be responsive, allowing it to adapt to different screen sizes and devices, making it more user-friendly and accessible. Overall, using Bootstrap.css is a practical choice for creating a visually appealing, user-friendly, and responsive website.

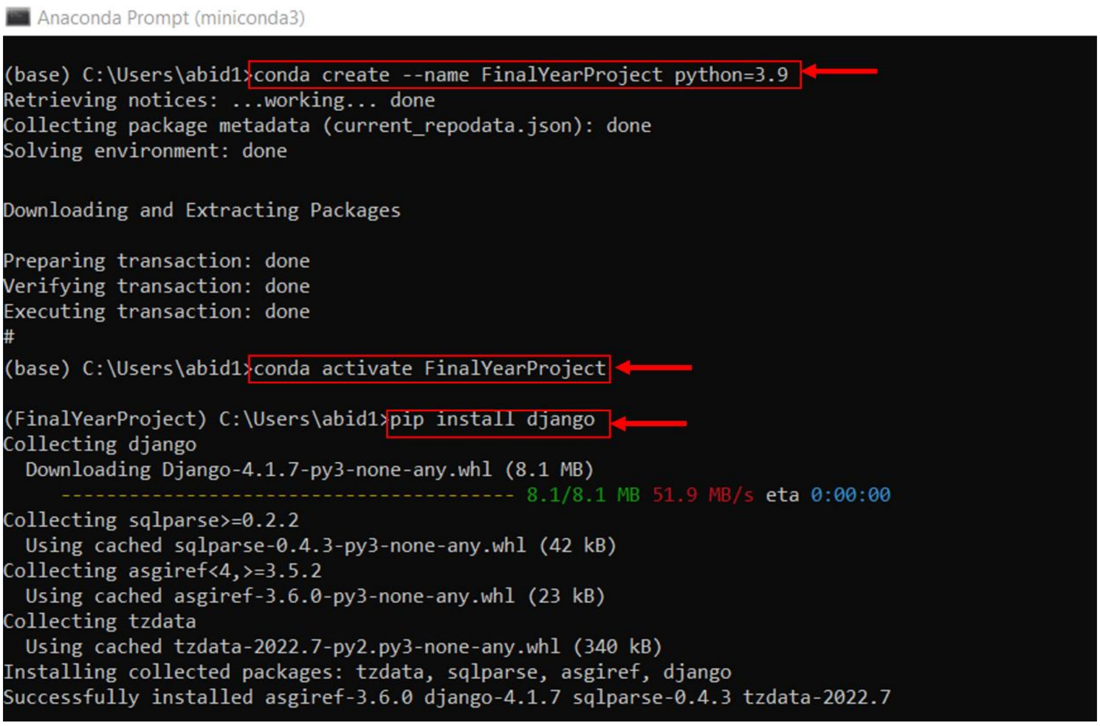
4.1.4 Railway Platform

For the project, I decided to use the Postgres database service offered by railway.app. Railway.app is a platform that provides a variety of services to developers, including managed databases. This means that the database is fully managed and maintained by railway.app, so I would not have to worry about things like configuration, security, or maintenance. This allowed me to focus on developing the web application and utilise the features of the Postgres database to efficiently manage and retrieve data.

4.1.5 Google Cloud Platform

For my Django web application project, I am using Google Cloud primarily for storing media files. This is because Google Cloud offers a highly scalable and reliable storage solution that can accommodate large amounts of media files without affecting the performance of my application. Additionally, Google Cloud provides an easy-to-use storage interface that seamlessly integrates with my Django project, allowing me to quickly upload and retrieve media files.

4.1.6 Conda Virtual Environment



```
Anaconda Prompt (miniconda3)

(base) C:\Users\abid1>conda create --name FinalYearProject python=3.9
Retrieving notices: ...working... done
Collecting package metadata (current_repodata.json): done
Solving environment: done

Downloading and Extracting Packages

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
(base) C:\Users\abid1>conda activate FinalYearProject

(FinalYearProject) C:\Users\abid1>pip install django
Collecting django
  Downloading Django-4.1.7-py3-none-any.whl (8.1 MB)
     ---------------------------------------- 8.1/8.1 MB 51.9 MB/s eta 0:00:00
Collecting sqlparse>=0.2.2
  Using cached sqlparse-0.4.3-py3-none-any.whl (42 kB)
Collecting asgiref<4,>=3.5.2
  Using cached asgiref-3.6.0-py3-none-any.whl (23 kB)
Collecting tzdata
  Using cached tzdata-2022.7-py2.py3-none-any.whl (340 kB)
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.6.0 django-4.1.7 sqlparse-0.4.3 tzdata-2022.7
```

Figure 12 - Django Environment Creation

Setting up a Conda environment is an essential part of developing software using Python, especially when using Django as the web framework. Conda is a popular package and environment management system that is widely used in data science, scientific computing, and software development. It allows users to create separate environments with specific package versions and dependencies, making it easy to manage different projects with conflicting requirements.

The screenshot provided shows the command 'conda create --name FinalYearProject python=3.9' which creates a new environment called 'FinalYearProject' using Python version 3.9. Once the environment is created, the command 'conda activate FinalYearProject' is used to activate it. This ensures that any packages installed, or commands executed are performed within the context of the environment.

Finally, the screenshot displays the command 'pip install django,' which is used to install Django within the environment. This means that Django and its dependencies are separated from other Python installations on the device. By creating a Conda environment and installing Django, I can use its MVC design to create an effective and maintainable web application.

4.2 Implementation Details

4.2.1 Database Models' Implementation

This part shows the various models that will be utilised in the application. The models are used for the database's structure and describe the relationships between entities. By defining what each model represents, it allows you to determine how the application will function and how the data will be organised within it.

4.2.1.1 Energy Provider Model

```
class EnergyProvider(models.Model):
    name = models.CharField(max_length=100, unique=True, default="Other")
    elecDailyCharge = models.DecimalField(
        decimal_places=2, max_digits=10, default="0")
    gasDailyCharge = models.DecimalField(
        decimal_places=2, max_digits=10, default="0")
    elecPerKwh = models.DecimalField(
        decimal_places=2, max_digits=10, default="0")
    gasPerKwh = models.DecimalField(
        decimal_places=2, max_digits=10, default="0")

    def __str__(self):
        return str(self.name)
```

Figure 13 - EnergyProvider Model

The code contains details of the energy providers, namely British Gas, EON, and EDF. The details include the name of each provider, their respective electricity and gas per kwh rates, and their daily charge for both electricity and gas. These details will remain fixed throughout the usage of the program, with a fixed number of objects in the model. However, the values for each provider will be utilised for calculations.

4.2.1.2 User Profile Model

```
class UserProfile(models.Model):
    PROVIDER_CHOICES = [
        ("EDF", "EDF"),
        ("British Gas", "British Gas"),
        ("EON", "EON"),
        ("OTHER", "Other"),
    ]

    user = models.OneToOneField(User, on_delete=models.CASCADE)
    dob = models.DateField()
    provider = models.ForeignKey(EnergyProvider, on_delete=models.CASCADE,
                                 max_length=20, choices=PROVIDER_CHOICES, default="OTHER")

    def __str__(self):
        return str(self.user)
```

Figure 14 - UserProfile Model

The User Profile model is integral for storing user information in the program. It includes a user field, which is a OneToOneField from the default Django user interface. In addition to this, it includes dob, and provider a foreign key. The provider field is linked to the energy provider model and is used to store the user's selected provider. Overall, the User Profile allows for storage and management of user information. This information can then be utilised to provide personalised calculations based on user's energy usage and preferences.

4.2.1.3 Appliance Models

```
class Appliance(models.Model):
    applianceType = models.CharField(max_length=11)
    name = models.CharField(max_length=100, default="Other")
    wattage = models.IntegerField()
    user = models.ForeignKey(UserProfile, on_delete=models.CASCADE)

    def __str__(self):
        return f"{self.user} - {self.name}"


class ApplianceUsage(models.Model):
    appliance = models.ForeignKey(Appliance, on_delete=models.CASCADE)
    duration = models.DecimalField(decimal_places=2, max_digits=10)
    date = models.DateField(auto_now_add=False)

    def __str__(self):
        return f"{self.appliance.user} - {self.appliance.name}"
```

Figure 15 - Appliance/Usage Models

These models shown in the code above, serve different purposes in storing data related to household appliances in the database. The Appliance model holds information about the appliances that a user has added, with the user's foreign key linking the relationship between the user and their appliances. On the other hand, the 'Appliance Usage' model has a foreign key of appliance and additional fields for duration and date. This allows the user to track how long an appliance was used and on what date. The reason for splitting up these models is to ensure that the data is organised and structured in a clear and efficient way for ease of use and maintenance.

4.2.1.4 Budget

```
class Budget(models.Model):
    user = models.ForeignKey(UserProfile, on_delete=models.CASCADE)
    month = models.CharField(max_length=50)

    dailyElecCostBudget = models.DecimalField(max_digits=10, decimal_places=2, null=True, blank=True)
    weeklyElecCostBudget = models.DecimalField(max_digits=10, decimal_places=2, null=True, blank=True)
    monthlyElecCostBudget = models.DecimalField(max_digits=10, decimal_places=2, null=True)

    dailyGasCostBudget = models.DecimalField(max_digits=10, decimal_places=2, null=True, blank=True)
    weeklyGasCostBudget = models.DecimalField(max_digits=10, decimal_places=2, null=True, blank=True)
    monthlyGasCostBudget = models.DecimalField(max_digits=10, decimal_places=2, null=True, blank=True)

    def __str__(self):
        return f"{self.user} - {self.month}"
```

Figure 16 - Budget Model

The Budget model is where users can set daily, weekly, and monthly budget for either Electricity or Gas. The data is then used to generate progress bars for users to track their spending and see how close they are to their budget, or if they have exceeded it.

4.2.1.5 Forum Models

```
class ForumPost(models.Model):
    title = models.CharField(max_length=200)
    description = models.TextField()
    createdBy = models.ForeignKey(UserProfile, on_delete=models.CASCADE)
    createdAt = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.title


class ForumComment(models.Model):
    post = models.ForeignKey(ForumPost, on_delete=models.CASCADE, related_name='comments')
    body = models.TextField()
    createdBy = models.ForeignKey(UserProfile, on_delete=models.CASCADE)
    createdAt = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.body
```

Figure 17 - Forum Models

The Forum models in Django's models.py is used to facilitate discussions among users. This involves creating posts, which are represented by the 'ForumPost' model. This model contains fields such as title, description, createdBy, and createdAt. Whenever a new discussion is created, an object of this model is instantiated and populated with relevant information. Users can also comment on these posts, and this is where the 'ForumComment' model comes in. It includes a 'post' field, which is a foreign key that references the actual post. In addition, it contains the reply body, the user who created the reply, and the time at which it was created. Overall, this setup allows for a seamless discussion experience where users can create posts and engage in conversations through comments and replies.

4.2.2 Setting up PostgreSQL database

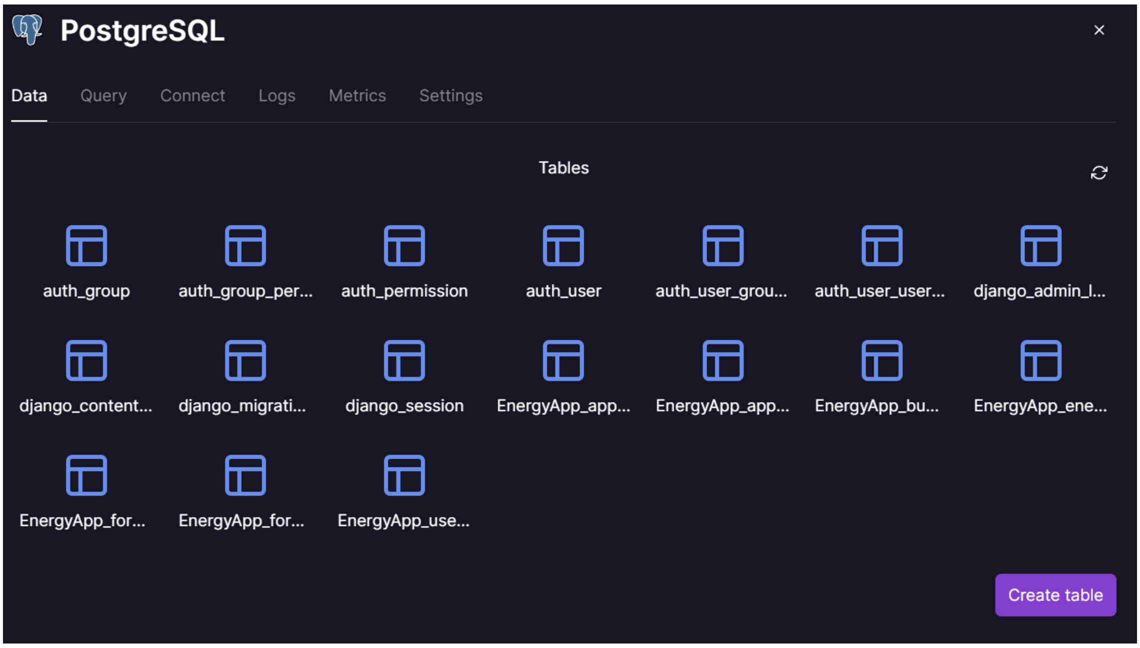


Figure 20 - Railway Platform

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'railway',
        'USER': 'postgres',
        'PASSWORD': 'vCD6JzSrJ2IB7nTOjQb3',
        'HOST': 'containers-us-west-195.railway.app',
        'PORT': '7844',
    }
}
```

Figure 21 - PostgreSQL configuration

This code snippet displays the PostgreSQL database configuration in the Django web application's settings.py file. It's important since it allows my app to connect to the database and perform data operations. My application would be unable to store or get data from the database without it, which is a requirement for many web applications.

Additionally, the database configuration is required to enable users to use the data in my application. Users can interact with the application, and the application can process user input and present relevant information to the user.

Furthermore, the database configuration serves as the foundation for the development and expansion of my application. It offers a framework for data management and scaling the application as the user base expands. This means that as more users use my application, the database can handle the increasing traffic while still providing a smooth user experience.

4.2.3 Google Cloud Configuration

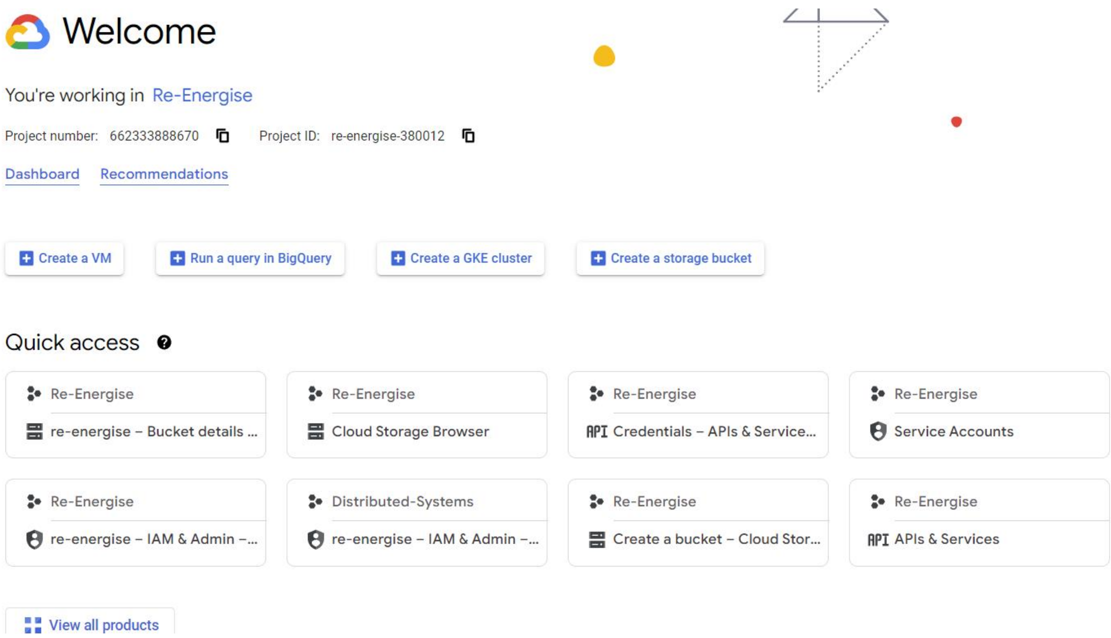


Figure 22 - Google Cloud Platform

Before configuring Django to connect with Google Cloud services, I first had to set up a service account under the API & Services section of the Google Cloud console. This service account allows the web application to securely access the Google Cloud storage bucket that would hold user media files, such as profile pictures. Once the service account was set up, I created a bucket in Google Cloud storage and applied the appropriate permissions to allow all users to access the media files stored in the bucket.

E.g., https://storage.googleapis.com/re-energise/profiles/lukeshaw.jpg

```
from google.oauth2 import service_account
GS_CREDENTIALS = service_account.Credentials.from_service_account_file(
    os.path.join(BASE_DIR, 'credential.json')
)
DEFAULT_FILE_STORAGE='EnergyProject.gcloud.GoogleCloudMediaFileStorage'
GS_PROJECT_ID = 're-energise-380012'
GS_BUCKET_NAME = 're-energise'
MEDIA_ROOT = "media/"
UPLOAD_ROOT = 'media/uploads/'
MEDIA_URL = 'https://storage.googleapis.com/{}/'.format(GS_BUCKET_NAME)
```

Figure 23 - Google Cloud Config

The code snippet above represents the configuration for Django to connect with Google Cloud services. This connection is necessary to store media files, such as profile pictures of users, for the web application. Specifically, the configuration specifies the file storage method, project ID, bucket name, and root directory location for the media files. By using Google Cloud services, the web application is able to efficiently and securely store and retrieve media files, providing an improved user experience.

4.2.4 Django Admin Portal

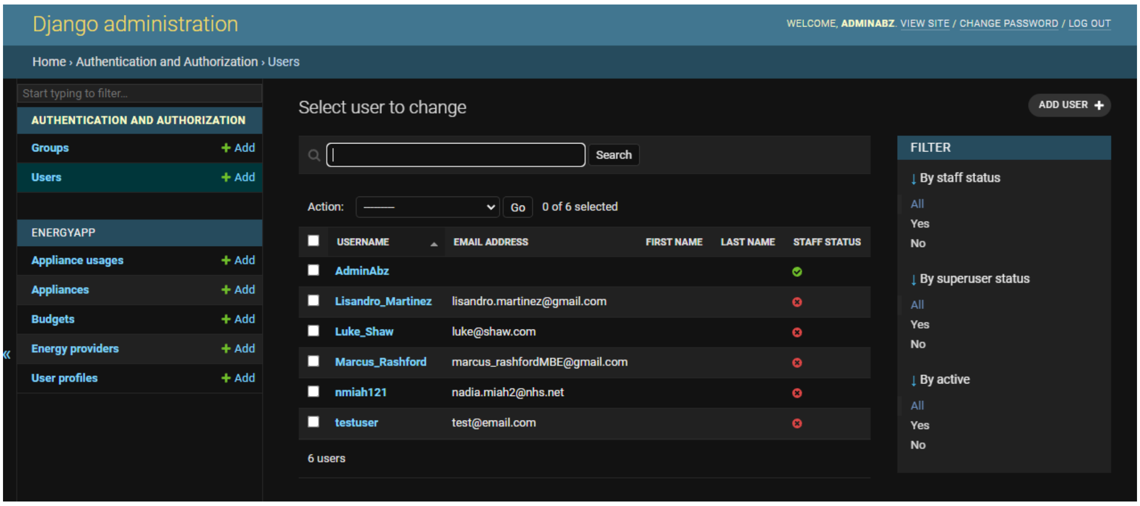


Figure 24 - Django Admin Portal

During the development of my Django web application, I utilised the Django Admin Portal to manage the database models in context. This portal provided a user-friendly interface for adding, modifying, and deleting data from the database, making it easier to manage without the need for complex SQL queries. With the help of the portal, I was able to create, modify and delete instances of the models, and access all the data present in those models.

While the Django Admin Portal did fulfil my application's needs, given more time, I would have preferred to design a personalised admin portal that could better address my application's specific needs. Nevertheless, the Django Admin Portal sufficiently provided the features I required, allowing me up to concentrate on other key elements of my project.

4.2.5 Link to Repository and Website Link

You can access the source code of the project application through the repository link, which is https://github.com/Abid-M/Energy-Management-System.

The live version of the application can be accessed through the website link, which is https://re-energise.xyz/

4.2.6 UI and Backend Implementation

In this section it discusses the user interface with the backend implementation of the application. The UI is the visual elements that the user sees and the controls that users interact with. Whilst the backend is the underlying technology that handles data processing that enables the application to function properly.

4.2.6.1 Landing Page

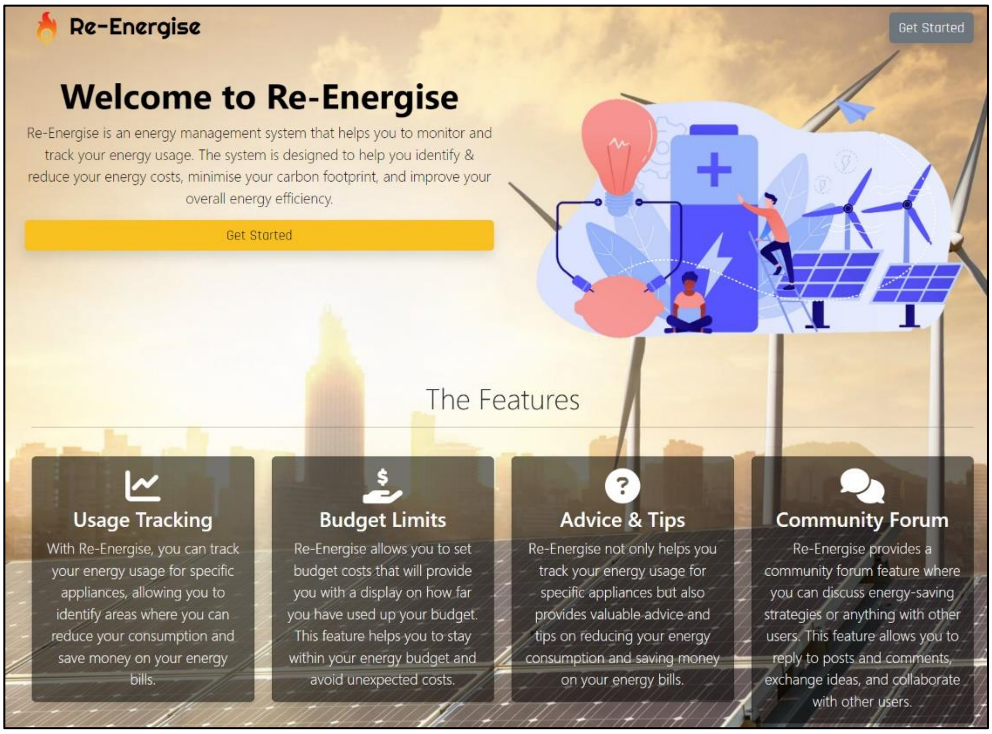


Figure 25 - Landing Page UI

The landing page serves as the user's initial interaction upon loading the website. It provides a clear introduction to the 'Re-Energise' application, highlighting its purpose and features. The aim of this page is to ensure that the user has a good understanding of what the application is all about.

4.2.6.2 Login/Register Screen

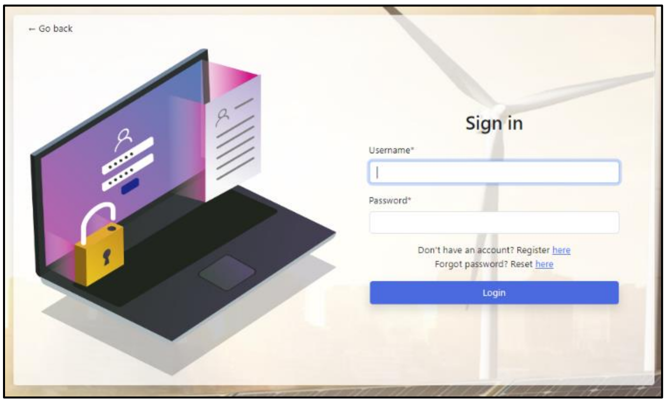


Figure 26 - Login Screen UI

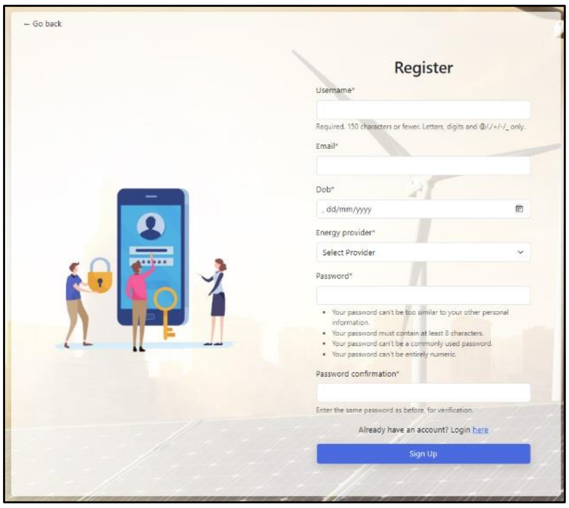


Figure 27 - Register Screen UI

4.2.6.3 Profile Page

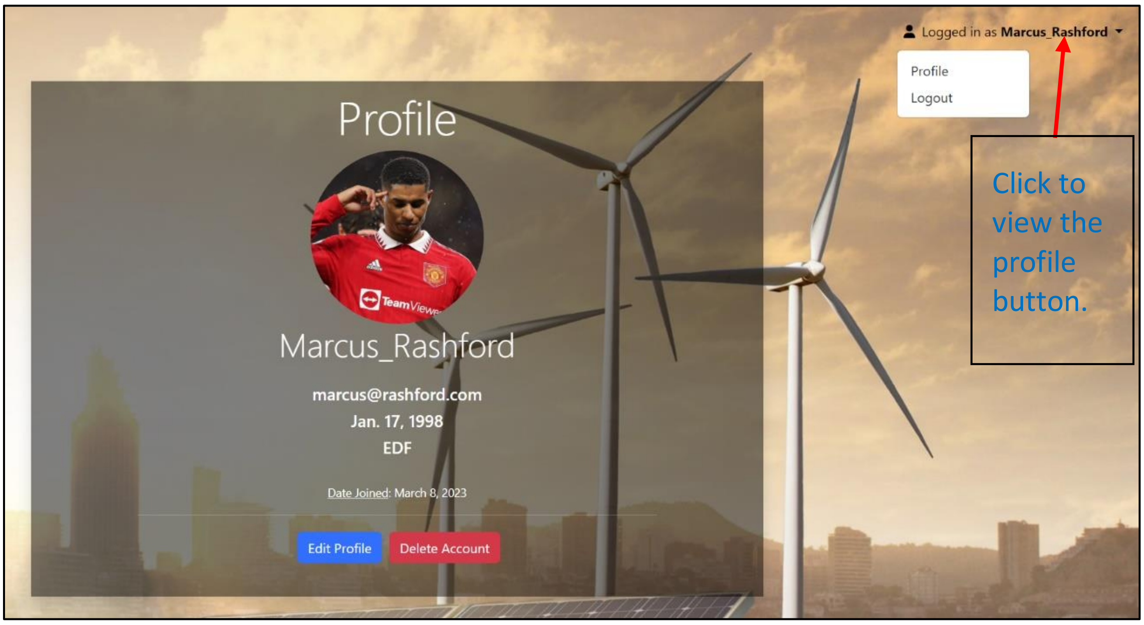


Figure 28 - Profile Page UI

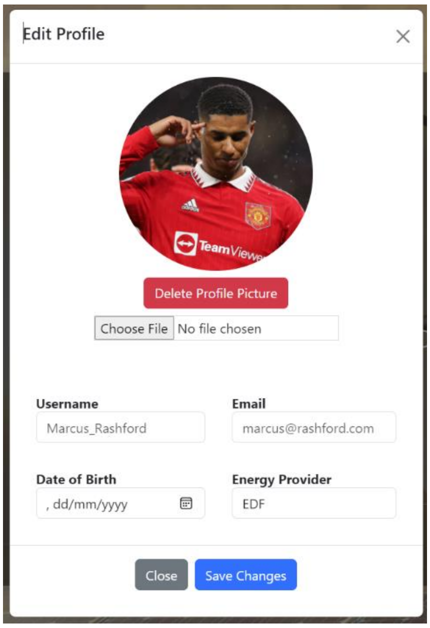


Figure 30 - Edit Profile UI

```python
@login_required(login_url="/login")
def profile(request):
    userProfile = UserProfile.objects.get(user=request.user)

    if request.method == 'POST':
        print('Printing POST:', request.POST)

        if request.POST.get('delete'):
            user = request.user
            user.delete()
            return redirect('homepage')

        if request.POST.get('delete_pic'):
                storage_client = storage.Client(credentials=settings.GS_CREDENTIALS, project=settings.GS_PROJECT_ID)
                bucket = storage_client.bucket(settings.GS_BUCKET_NAME)
                default_pic_path = 'static/images/defaultImage.svg'

                userProfile.picture = None
                userProfile.save()

        else:
            username = request.POST.get('username', '')
            email = request.POST.get('email', '')
            picture = request.FILES.get('profile_pic', '')
            dob = request.POST.get('dob', '')
            provider = request.POST.get('provider', '')

            user = User.objects.get(id=request.user.id)
            userProfile = user.userprofile

            if username:
                user.username = username
            if email:
                user.email = email
            if picture:
                # if userProfile.picture:
                    # os.remove(userProfile.picture.path)
                userProfile.picture = picture
            if dob:
                userProfile.dob = dob
            if provider:
                userProfile.provider = EnergyProvider.objects.get(name=provider)

            user.save()
            userProfile.save()

            return redirect('profile')

    return render(request, 'main/profile.html')
```

Figure 31 - Profile backend

The profile page provides users with the ability to manage and edit their information as needed, such as changing their energy provider, if they switch to a new supplier. This page is designed to be user-friendly, allowing users to easily update their details as necessary.

Upon click on their profile picture it would show in full view, e.g., https://storage.googleapis.com/re-energise/profiles/0_rashford.jpg

4.2.6.4 Dashboard Page

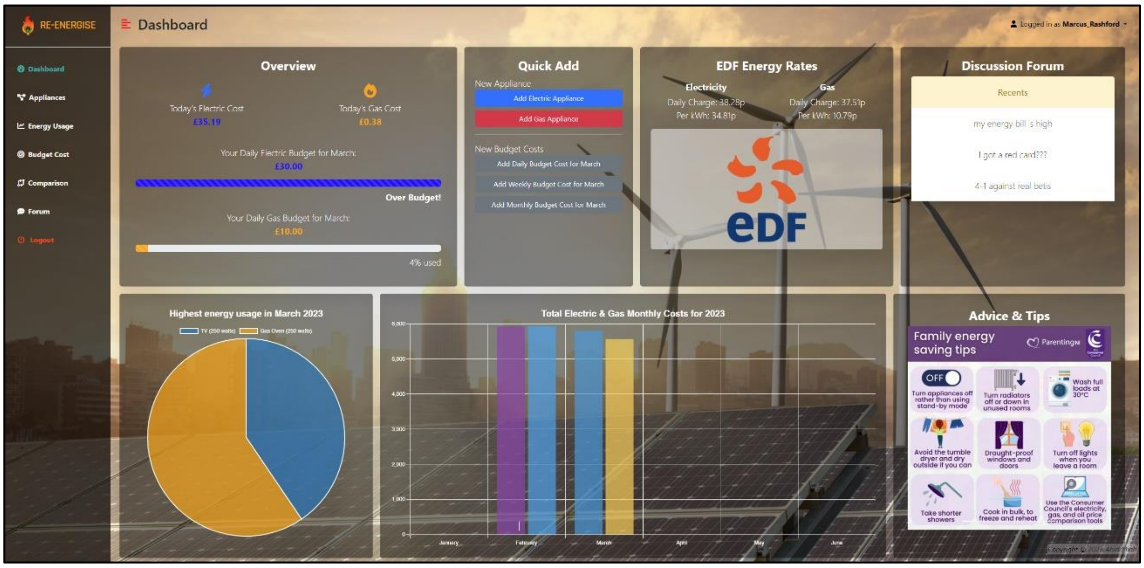


Figure 32 - Dashboard UI

After logging in, users are redirected to the dashboard page. Here, they can view an overview of today's electric and gas costs, as well as progress bars that show their budget status if one is set. Quick-add buttons provide users with easy access to frequently used features, and the page also displays information on the energy rates of the user's provider and recent posts from the discussion forum. Two graphs are also included on this page - a bar chart that shows the total monthly electric and gas costs for the current year, as well as a pie chart that displays the appliance with the highest energy usage for the current month. Finally, a slideshow of advice and tips is included to assist users in managing their energy usage.

```python
# Budget Display
todayEDailyCost = 0
todayGDailyCost = 0
for usage in todayAppliances:
    if usage.appliance.applianceType == 'Electricity':
        wattage = usage.appliance.wattage
        duration = usage.duration

        print(todayEDailyCost, wattage, duration, providerInfo.elecPerKwh)
        todayEDailyCost = todayEDailyCost + (((wattage * duration)/1000) * providerInfo.elecPerKwh)

    elif usage.appliance.applianceType == 'Gas':
        wattage = usage.appliance.wattage
        duration = usage.duration

        todayGDailyCost = todayGDailyCost + (((wattage * duration)/1000) * providerInfo.gasPerKwh)


todayEDailyCost = todayEDailyCost + (providerInfo.elecDailyCharge/100)
todayGDailyCost = todayGDailyCost + (providerInfo.gasDailyCharge/100)

if budget is not None:
    if budget.dailyElecCostBudget and budget.dailyElecCostBudget != 0.00 :
        eDailyPercent = round((todayEDailyCost / budget.dailyElecCostBudget) * 100)
    else:
        eDailyPercent = 0.00

    if budget.dailyGasCostBudget and budget.dailyGasCostBudget !=0.00 :
        gDailyPercent = round((todayGDailyCost / budget.dailyGasCostBudget) * 100)
    else:
        gDailyPercent = 0.00
else:
    eDailyPercent = 0.00
    gDailyPercent = 0.00

todayEDailyCost = round(todayEDailyCost, 2)
todayGDailyCost = round(todayGDailyCost, 2)
# budget display end
```

Figure 33 - Dashboard code snippet

This code snippet is from the backend view of the dashboard and pertains to the budget functionality, which includes progress bars and calculations to determine whether the user is within their budget or not. The code loops through all the appliances and calculates today's gas and electric costs using the formula (wattage*duration/100), multiplied by the per kWh rate of the user's energy provider.

4.2.6.5 Appliance Page

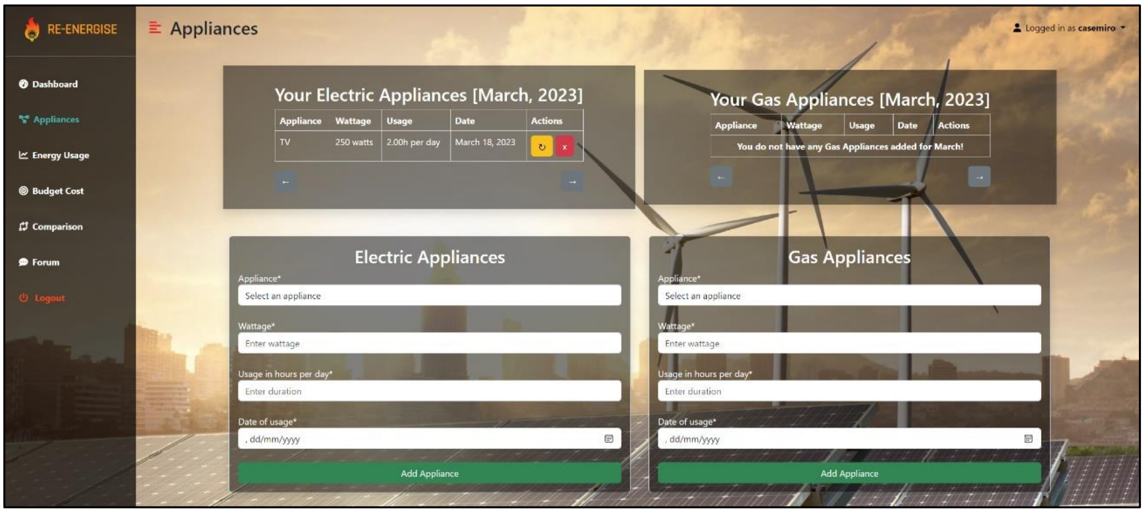


Figure 34 - Appliances UI

This page provides an overview of all the electric and gas appliances that the user has added. The page includes buttons that allow users to view the appliance history from previous months and gives them the ability to create, update, or delete appliances as needed.

```python
@login_required(login_url="/login")
def appliances(request):
    userProfile = UserProfile.objects.get(user=request.user)
    userAppliances = Appliance.objects.filter(
        user=userProfile)

    # where the field appliance, __in is in the appliances query set
    applianceUsage = ApplianceUsage.objects.filter(
        appliance__in=userAppliances).order_by('-date')

    currentMonth = datetime.now().strftime("%B")

    context = {
        "applianceUsage": applianceUsage,
        "userAppliances": userAppliances,
        "currentMonth": currentMonth,
    }

    if request.method == 'POST':
        print(request.POST)
        userProfile = UserProfile.objects.get(user=request.user)

        if (request.POST.get('appType') == 'electricity'):
            print("this is an electric appliance")

            elecAppliance = request.POST.get('electricity_appliance')
            elecApplianceName = request.POST.get('electricity_appliance_name')
            elecWattage = request.POST.get('electricity_wattage')
            elecDuration = request.POST.get('electricity_duration')
            elecDate = request.POST.get('electricity_date')

            if elecAppliance == "Other" or elecAppliance == "":
                if elecApplianceName == "":
                    createdAppliance = Appliance.objects.create(
                        applianceType="Electricity",
                        user=userProfile,
                        name="Other (Not Specified)",
                        wattage=elecWattage,
                    )
                else:
                    createdAppliance = Appliance.objects.create(
                        applianceType="Electricity",
                        user=userProfile,
                        name=elecApplianceName,
                        wattage=elecWattage,
                    )

                createdAppliance.save()
```

Figure 35 - Appliance code snippet

This code snippet is taken from the backend view of the appliance page. It contains a loop that filters all the appliance objects in reverse order, and only includes those that are within the 'applianceUsage' object, effectively gathering all the user's appliances.

Additionally, the code shows the request method that handles a post request whenever the user adds an electric or gas appliance. The code retrieves values from the fields and creates a new object. In instances where the appliance is not specified, it is stored as 'other.

4.2.6.6 Energy Usage Page

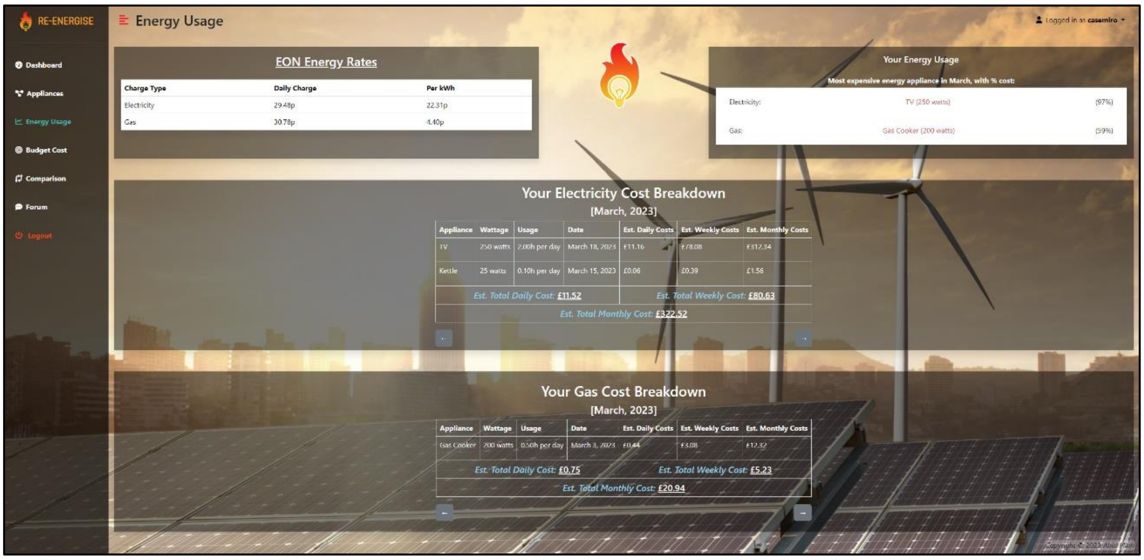


Figure 36 - Energy Usage UI

```
if (etotalDailyCosts > 0):
    etotalDailyCosts = round(
        etotalDailyCosts + (providerInfo.gasDailyCharge/100), 2)
    etotalWeeklyCosts = round(
        etotalWeeklyCosts + (((providerInfo.gasDailyCharge*7)/100)), 2)
    etotalMonthlyCosts = round(
        etotalMonthlyCosts + (((providerInfo.gasDailyCharge*28)/100)), 2)

if (gtotalDailyCosts > 0):
    gtotalDailyCosts = round(
        gtotalDailyCosts + (providerInfo.gasDailyCharge/100), 2)
    gtotalWeeklyCosts = round(
        gtotalWeeklyCosts + (((providerInfo.gasDailyCharge*7)/100)), 2)
    gtotalMonthlyCosts = round(
        gtotalMonthlyCosts + (((providerInfo.gasDailyCharge*28)/100)), 2)

if (etotalDailyCosts > 0):
    averageeHighest = round(((((highesteDailyCost/etotalDailyCosts)*100) + (
        (highesteWeeklyCost/etotalWeeklyCosts)*100) + ((highesteMonthlyCost/etotalMonthlyCosts)*100)) / 3), 0)

if (gtotalDailyCosts > 0):
    averagegHighest = round(((((highestgDailyCost/gtotalDailyCosts)*100) + (
        (highestgWeeklyCost/gtotalWeeklyCosts)*100) + ((highestgMonthlyCost/gtotalMonthlyCosts)*100)) / 3), 0)
```

Figure 37 – Usage code snippet A

```
for usage in applianceUsage:
    if usage.appliance.applianceType == 'Electricity':
        if usage.date.strftime("%B") == currentMonth:
            wattage = usage.appliance.wattage
            duration = usage.duration

            dailyCost = (((wattage * duration)/1000)
                         * providerInfo.elecPerKwh)
            weeklyCost = dailyCost * 7
            monthlyCost = weeklyCost * 4

            etotalDailyCosts = etotalDailyCosts + dailyCost
            etotalWeeklyCosts = etotalWeeklyCosts + weeklyCost
            etotalMonthlyCosts = etotalMonthlyCosts + monthlyCost

            dailyCost = round(dailyCost, 2)
            weeklyCost = round(weeklyCost, 2)
            monthlyCost = round(monthlyCost, 2)

            if dailyCost > highesteDailyCost:
                highesteDailyCost = dailyCost
                highesteWeeklyCost = weeklyCost
                highesteMonthlyCost = monthlyCost

                elecCostlyAppliance = f"{usage.appliance.name} ({usage.appliance.wattage} watts)"

            applianceDailyCosts[usage.appliance.id] = dailyCost
            applianceWeeklyCosts[usage.appliance.id] = weeklyCost
            applianceMonthlyCosts[usage.appliance.id] = monthlyCost
```

Figure 38 - Usage code snippet B

The energy usage page provides a breakdown of the user's costs per month, as well as estimated daily, weekly, and monthly costs. The accompanying code snippets illustrate how these values are calculated.

4.2.6.7 Budget Cost Page

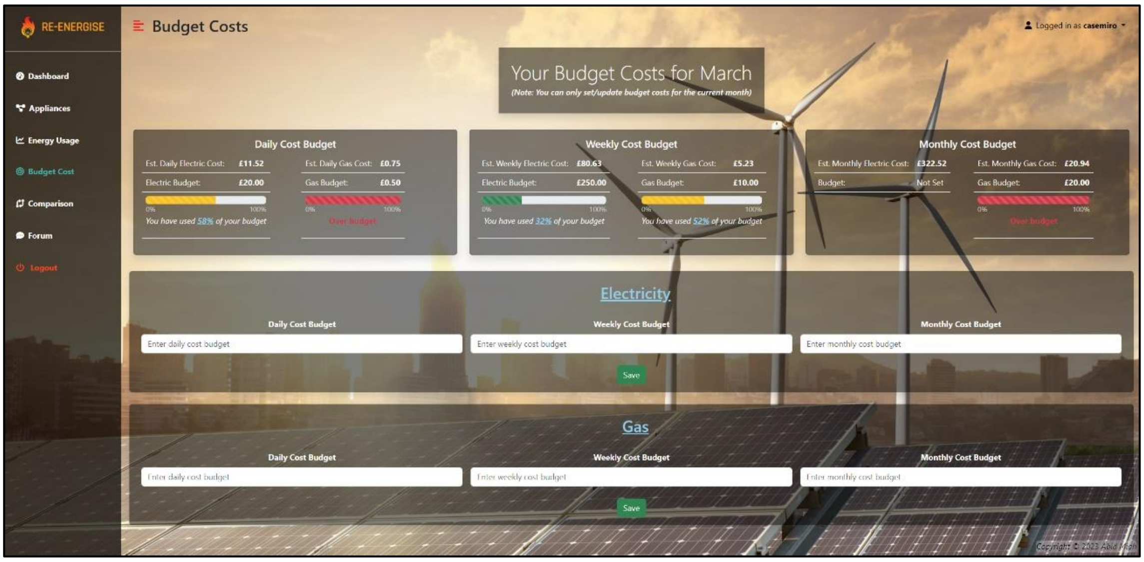


Figure 39 - Budget Costs UI

```javascript
function showAlert(type, message) {
    const alert = document.createElement('div');
    alert.className = `alert alert-${type} mt-3 text-center justify-content-center align-items-center`;
    alert.textContent = message;
    eform.prepend(alert);

    // Remove the alert after 3 seconds
    setTimeout(() => {
        location.reload();
        alert.remove();
    }, 1500);
}
```

Figure 40 - JavaScript Code Snippet A

```javascript
const eform = document.querySelector('#elec-budget');
const gform2 = document.querySelector('#gas-budget');

eform.addEventListener('submit', (event) => {
  event.preventDefault();
  let form = event.currentTarget
  // Submit the form using AJAX
  const formData = new FormData(form);
  fetch(form.action, {
      method: 'POST',
      body: formData,
  })
  .then(response => {
      if (response.ok) {
          showAlert('success', 'Added/Updated Budget Successfully');
      } else {
          showAlert('danger', 'An error occurred. Please try again.');
      }
  })
  .catch(error => {
      showAlert('danger', 'An error occurred. Please try again.');
  });
});
```

Figure 41 - JavaScript Code Snippet B

The budget cost page displays the user's set budget costs in relation to their estimated daily, weekly, and monthly costs. Users can add or update their budget for the current month using the form provided.

The code snippets illustrate the use of JavaScript, the fetch API, and AJAX requests to asynchronously update or add the user's budget.

4.2.6.8 Comparison Page

Figure 42 - Comparison UI

An iFrame was used to link to a website and display real-time and historical GB data that is of interest to the user.

4.2.6.9 Forum Page

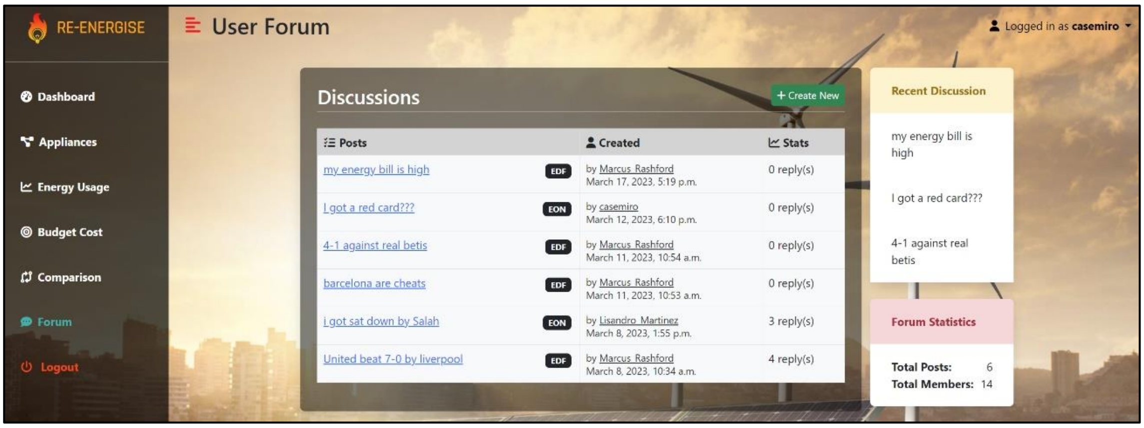


Figure 43 - Forum Page UI

```python
@login_required(login_url="/login")
def forum(request):
    userProfile = UserProfile.objects.get(user=request.user)
    userPosts = ForumPost.objects.order_by('-createdAt')
    totalUserPosts = ForumPost.objects.all().count()
    totalMembers = UserProfile.objects.all().count()

    recentPosts = []
    sortedPosts = []
    for post in ForumPost.objects.all():
        try:
            latest_comment = post.comments.all().latest('createdAt')
            last_active = latest_comment.createdAt
        except ForumComment.DoesNotExist:
            last_active = post.createdAt
        recentPosts.append((post, last_active))

    sortedPosts = sorted(recentPosts, key=lambda x: x[1], reverse=True)

    if request.method == 'POST':
        print('Printing POST:', request.POST)

        postTitle = request.POST.get('post_title')
        postDesc = request.POST.get('post_description')

        ForumPost.objects.create(
            title = postTitle,
            description = postDesc,
            createdBy = userProfile,
        ).save()

        return redirect('/forum')

    context = {
        "userPosts": userPosts,
        "totalUserPosts": totalUserPosts,
        "totalMembers": totalMembers,
        "sortedPosts": sortedPosts,
    }
```

Figure 44 - Forum Page Code Snippet

The discussion forum page allows users to engage in conversations about energy-saving strategies or any other relevant topics.

The accompanying code snippet illustrates the backend functionality, where all forum posts are fetched from the forum model, sorted in reverse order, and new posts are handled accordingly.

4.2.6.10 Forum Post Page

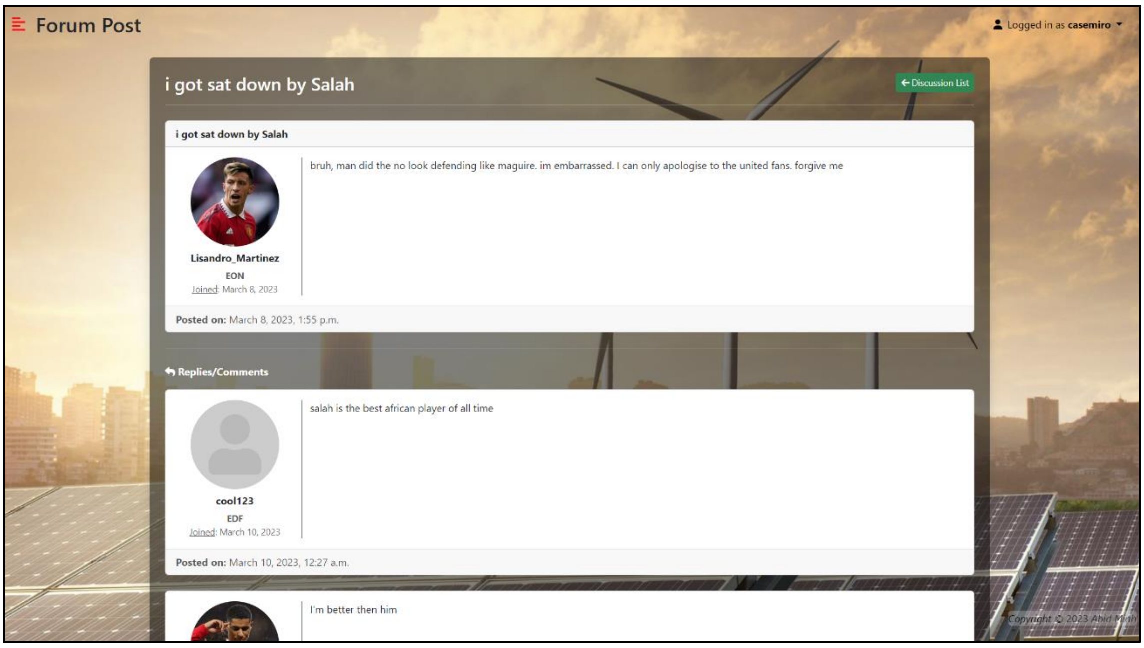


Figure 45 - Forum Post UI

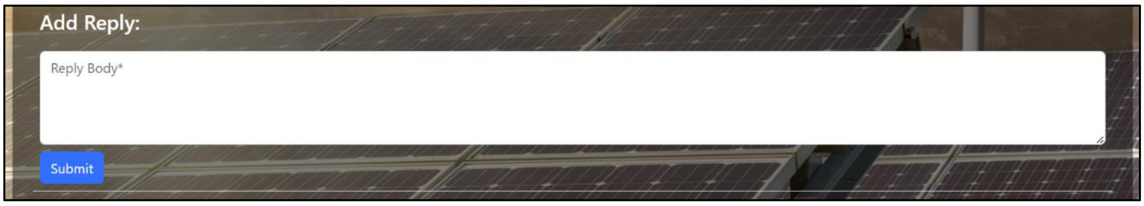


Figure 46 - Forum Post UI reply

```python
@login_required(login_url='/login')
def post(request, id):
    specificPost = get_object_or_404(ForumPost, id=id)
    postComments =  ForumComment.objects.filter(post=specificPost)
    userProfile = UserProfile.objects.get(user=request.user)

    if (request.method == 'POST'):

        if request.POST.get('delete'):
            specificPost.delete()
            return redirect('forum')


        print('Printing POST:', request.POST)
        replyDesc = request.POST.get('reply_body')

        ForumComment.objects.create(
            post = specificPost,
            body = replyDesc,
            createdBy = userProfile,
        ).save()

        return redirect(f'/post/{id}')


    context = {
        'specificPost' : specificPost,
        'postComments': postComments
    }

    return render(request, 'main/post.html', context)
```

Figure 47 - Forum Post Code Snippet

Upon clicking a forum post, the user will be redirected to the post page where they can view the title, description and add their own comments/replies.

The following code snippet retrieves the specific post using get_object_or_404 with the ID. It then filters all comments/replies related to that post and handles POST requests for users to comment.

Chapter 5: **Testing**

In this section, I am going to discuss the crucial aspect of software development called software testing, for my energy management application. Software testing is a vital step in the development process, as it guarantees the reliability, functionality, and performance of the application.

5.1 Unit & Functional Testing

Unit testing is a fundamental testing method that focuses on examining the smallest units of code, such as individual functions or methods, to ensure that they work correctly and as intended. The results are displayed below:

Dashboard Screen		
Feature	Expected Result	Outcome
View dashboard overview of the user’s energy costs today, quick add, graphs and charts, etc.	Intended information rendered on screen, that is customised for the logged in user.	Success (no user added appliance = no graphs shown, as there is no data)
Quick Add buttons working to add new appliances and budget costs.	New data (appliances or budget costs), created in the database and page refreshes to show update.	Success
Scroll through advice and tips and hover and click to view the advice or tip in full.	Users can smoothly scroll through the list of advice and tips, and when they hover over a particular item, carrousel stops. Upon clicking the item, the full image of the selected advice or tip are displayed in a new tab.	Success
View of discussion forum showing top 3 recent forum posts with clickable links redirecting to the respective post pages.	Upon clicking on any of these posts, the user is redirected to the corresponding post page, where they can read the full content and participate in the discussion.	Success

Profile Screen		
Feature	Expected Result	Outcome
Viewing the logged-in user's profile with their personal details	The user's profile page displays their username, email, date of birth, energy provider, and date joined, presented in a clear and organised format.	Success

Edit profile button allowing users to modify their personal details	Clicking the Edit Profile button opens a form for editing user details and profile image. Saving changes updates the database and reflects the new information on the profile page.	Success
Delete account button with confirmation alert and account removal	Clicking Delete Account prompts a confirmation alert. After confirming, the account is deleted, and the user is redirected to the homepage.	Success

Appliance Screen		
Feature	Expected Result	Outcome
Viewing the user's electric/gas appliances for the current month and navigating through past months.	The appliances page displays a list of the user's electric/gas appliances for the current month. Arrow buttons allow the user to navigate through previous months' data seamlessly.	Success
Adding electric/gas appliances, with fields for appliance name, wattage, usage in hours per day, and date of usage.	The user can access a form to add new electric/gas appliances. The form ensures input validation to prevent incorrect or incomplete data entry. Once submitted, data saved in database.	Success
Updating appliance details if the user made a mistake.	Users can edit appliance details if needed. When updating an appliance's information, a pre-filled input form allows them to modify the required fields. The revised data is then saved to the database and displayed on the appliances page.	Success
Deleting an appliance from the list	Users can delete an appliance by clicking its associated delete button. A confirmation alert verifies their aim. Once confirmed, the appliance is removed from the database, and the appliances page updates to exclude the deleted item.	Success

Energy Usage Screen		
Feature	Expected Result	Outcome
Viewing energy provider's rates for electricity and gas.	The energy usage screen displays the user's energy provider's rates, clearly showing charge types (electricity/gas) and their respective daily charges and per kWh rates.	Success
Displaying the most expensive energy appliance	The screen shows the most expensive energy appliance of the current month, along with its overall percentage cost, providing users with insights into their highest energy-consuming appliances.	Success
Viewing an energy cost breakdown.	The energy usage screen presents a comprehensive breakdown of energy costs for both electricity and gas. It lists all appliances and their corresponding daily, weekly, and monthly costs, as well as the overall estimated total costs for each time period.	Success

Budget Cost Screen		
Feature	Expected Result	Outcome
Adding daily, weekly, and monthly cost budgets for gas and electricity.	Users can set daily, weekly, and monthly cost budgets for the current month for both gas and electricity, which are saved in the database.	Success
Viewing daily, weekly, and monthly costs and budgets.	Users can see their daily, weekly, and monthly costs and budgets, along with a progress bar displaying the percentage of budget used relative to the estimated costs for each time period.	Success

Discussion Forum/Posts Screen		
Feature	Expected Result	Outcome
Displaying a list of forum posts.	The discussion forum page shows a list of posts, including the author, creation time, and the number of replies for each post.	Success

Viewing forum statistics.	Users can see forum statistics, like the total number of posts and members, providing an overview of the forum's activity.	Success
Creating a new discussion post.	Users can create their own discussion posts, entering a title and description. The new post is saved in the database.	Success
Viewing specific posts details.	When users click on a post, they can see its full details, including the description, providing a comprehensive view of the post's content.	Success
Adding replies/comments to a specific post.	Users can easily add replies or comments to a post, this is saved to the database.	Success
Deleting a post	Logged-in users have the option to delete their own created posts, allowing them to manage and control their contributions to the discussion forum.	Success (The delete button only presented for the posts created by the logged-in user, allowing them to delete only their own posts.)

Login and Register		
Feature	Expected Result	Outcome
User login with existing credentials.	Upon authentication, user is redirected to dashboard screen.	Success
Creating a new account.	New user is created and saved in database, given their details which include username, email, dob, energy provider and password.	Success

5.2 User Participant Testing

User Participant testing is an essential step in evaluating the energy management application. Since the application is user-oriented, it is crucial to observe and understand users' experiences and gather their feedback. This helps to ensure that the application is intuitive, user-friendly, and meets the target audience's needs effectively.

Participants were sent a link to the live website along with a questionnaire to fill out. This approach allowed me to collect valuable insights and feedback directly from the users. The results and findings of the User Participant testing can be found in the appendix c.

5.3 Non-Functional Testing

The application will be compatible to run on both mobile and web browsers.

- This requirement has been fulfilled, as the application has been designed to run on both mobile and web browsers, ensuring responsiveness and seamless scaling to various screen sizes. However, there is one page that is not scaled responsively.

The application will protect the privacy and security of user's data, such as password and energy usage.

- This requirement has been met, as the application implements security measures, including hashing user passwords and utilising Django's built-in security features, to protect user data and ensure privacy for energy usage information.

The application must be reliable and available to use at all times, with minimal downtime for maintenance and updates.

- The application is designed to be always reliable and available for use, utilising railway.app for deployment, which ensures minimal downtime during maintenance and updates, providing users with a consistent and seamless experience.

The application should compute calculations and achieve desired response given an input into the system.

- The application effectively computes calculations, including estimated energy costs, and generates accurate responses based on user inputs, ensuring reliable and consistent performance throughout the system.

The system must have a user-friendly interface that is easy to navigate and use.

- Although some pages may take longer to load, the system's overall user interface has been designed with a focus on user-centred design principles, resulting in an interface that is generally easy to use and navigate.

New data and updates for existing features should be reviewed frequently.

- As this is a final year project for university and may not intended for long-term use, the requirement for frequent review of new data and updates may not be applicable. However, I have made efforts to ensure that the system can accommodate future updates or modifications if needed.

The system must be able to accommodate increasing numbers of users and data without a significant decrease in performance.

- As the system has not been tested on a large-scale basis, I cannot confirm whether the requirement for accommodating increasing numbers of users and data without a significant decrease in performance has been met.

5.4 Penetration Testing

In this section, I'll be carrying out few penetration test cases on the application, focusing on detecting and resolving security vulnerabilities related to incorrect inputs and other possible issues. This process will allow me to enhance the application's security and make sure it operates effectively under different conditions.

Test Case	Input Value	Expected Output	Result
Adding valid data in the appliance form.	Appliance: ‘Refrigerator’, Wattage: ‘200’, Usage per day: ‘4’, Date: ‘2023-04-10’	New appliance entry successfully added to the database and displayed on the appliances page .	Success
Leaving a field empty in the appliance form.	Appliance: ‘Refrigerator’, Wattage: ‘200’, Usage per day: ‘’, Date: ‘2023-04-10’	Display an error message: "Please fill in this field".	Success
Entering usage in hours greater than 24.	Appliance: ‘Refrigerator’, Wattage: ‘200’, Usage per day: ‘30’, Date: ‘2023-04-10’	Display an error message: "Usage cannot be greater than 24 hours".	Success
Selecting a date in the future for appliance usage.	Appliance: ‘Refrigerator’, Wattage: ‘200’, Usage per day: ‘5’, Date: ‘*(future date)*’	Display an error message: "Date cannot be in the future".	Success
Typing letters in the usage in hours field.	Appliance: ‘Refrigerator’, Wattage: ‘200’, Usage per day: ‘*(letters entered)*’, Date: ‘2023-04-10’	The letters are not displayed in the usage field as it only accepts number inputs.	Success
Entering valid login credentials.	Username: ‘Marcus_Rashford’, Password: ‘testpass123’	User is successfully logged in and redirected to the dashboard.	Success
Entering incorrect login credentials.	Username: ‘Marcus_Rashford’, Password: ‘testpasss123’	Display an error message invalid credentials	Success (error message: "Please enter a correct username and password. Note that both fields may be case-sensitive.")

5.5 Test Findings & Summary

Throughout the testing process, various test cases were executed, covering both positive and negative scenarios. This thorough approach provided valuable insights into the application's performance, reliability, and security.

The positive test cases primarily focused on validating the application's functionality when provided with correct input values. These tests demonstrated the expected behaviour and confirmed the successful implementation of features such as appliance data submission, user authentication, and profile management. On the other hand, negative test cases were designed to evaluate the application's resilience against incorrect inputs and unexpected user actions. These tests revealed potential vulnerabilities and areas for improvement, such as handling empty fields, preventing excessive usage hours, and ensuring proper input validation for text and numeric fields.

The unit testing phase focused on individual components of the application, ensuring that each piece of functionality performed as intended. This testing revealed that the core functionalities, such as data manipulation and calculations, were well-implemented and robust. It also highlighted the importance of modular code and proper error handling at the component level.

During the functional testing phase, I assessed the application's overall performance and ensured that all features worked together cooperatively. Key findings included the successful implementation of features like appliance management, user authentication, and profile editing. Additionally, I identified areas where user experience could be improved, such as providing better feedback for empty fields and handling invalid input values.

Non-functional testing examined aspects like compatibility, privacy, security, and reliability. The findings demonstrated that the application was compatible with both mobile and web browsers, ensuring responsiveness and proper scaling to different screen sizes. Privacy and security were maintained through measures such as password hashing and the use of Django framework. The application's reliability and availability were supported using railway.app for deployment.

The penetration testing phase focused on assessing the application's resilience against incorrect inputs and potential security vulnerabilities within the scope of the application. Key findings from these tests included the importance of input validation and proper error handling. This testing process allowed me to uncover potential vulnerabilities and improve the overall security of the application.

In summary, the comprehensive testing approach has provided valuable insights into the application's strengths and areas for improvement. The key findings from unit testing, functional testing, non-functional testing, and penetration testing have guided me towards refining the application to ensure it is reliable, secure, and user-friendly.

Chapter 6: **Evaluation**

6.1 Evaluation against Project Objectives

This section provides an evaluation of the project against its initial objectives. This evaluation is essential to assess the extent to which the project has achieved its intended goals and identify any areas that may require further improvement or refinement.

	Objectives	Outcome
1	Design and develop a web-based application that provides users with the best tools for managing and tracking energy consumptions with added tips to reduce costs. Some of these will include: ▪ Home Dashboard Screen which displays personalised overall energy consumption in a specific time interval, showing graphs of trends and latest news updates. ▪ Users being able to add their different appliances which would calculate the cost based on the watts and tariffs supplied by the user. ▪ Breakdown page of all appliances with the given quantities and consumption pet unit. Showing daily, weekly, monthly and yearly. ▪ Providing tips and advice. The system will provide tips and advice to the user on how they could save on their energy costs. ▪ The estimate cost calculating by the system would compare against their actual bill.	**Chapter 4: Implementation** • 4.2 Implementation Details
2	• To implement the use of Django web framework to help with developing the application, both frontend and backend.	**Chapter 4: Implementation** • 4.1.1 Django & Python • 4.2.1 Database Models' • 4.2.4 Django Admin Portal
3	• Several tests of the application conducted and whether fits all needs and requirements. Testing surveying purpose and proofing design.	**Chapter 5: Testing**
4	• Conduct a literature review on existing systems and the potential benefits my system will bring upon from its use.	**Chapter 2: Literature Review** • 2.3 Existing Solutions • 2.4 Proposed Solution

6.2 Evaluation against Existing Solutions

A key distinction of the Re-energise application is its emphasis on regular users instead of businesses. This user-focused approach enables the provision of a more intuitive and straightforward experience, catering to individuals who may not possess extensive knowledge of energy management concepts. By prioritising simplicity and clarity, the application addresses a critical need that existing solutions have not fully catered to.

The Re-energise application is designed to empower users to make informed decisions about their energy usage without being overwhelmed by complex interfaces or technical jargon. Alongside its user-centric design, the application incorporates essential features like appliance management, budgeting, discussion forum, and usage tracking. These functionalities further enhance the application's value, making it a comprehensive tool for effectively managing energy consumption.

6.3 Project Strengths

The project has numerous strengths that contribute to its success and make it stand out from other energy management solutions. A notable strength is the incorporation of a discussion forum feature, which creates a sense of community among users. This feature allows for meaningful conversations, tips sharing, and idea exchanges on energy management, ultimately promoting better engagement within the platform.

Positive feedback received from user testing is another strong aspect of the project. Users found the application to be useful and user-friendly, validating its effectiveness in addressing the needs of the target audience. Furthermore, the project provides straightforward navigation across the system, allowing users to access different features and tools with ease. The attractive and well-structured user interface plays a vital role in drawing and retaining users, ensuring a comfortable and efficient interaction with the application.

These strengths, including the engaging discussion forum, positive user feedback, effortless navigation, and appealing user interface, all come together to make this project a successful and comprehensive energy management solution that meets the needs of its users.

6.4 Project Limitations & Challenges

Even though the project has been successful, it was important for me to recognise some limitations and challenges I faced along the process. One notable limitation is the slightly longer loading times for the website pages compared to other sites. This could be due to the volume of data being processed or the server's response time. Optimising the website's performance is an area to consider for future improvement to ensure a better user experience.

Another limitation is the lack of real-time energy tracking in the application. While it wasn't initially part of the project plan, adding this feature could have greatly enriched the application's capabilities, providing users with up-to-the-minute energy consumption data, and enabling more informed decision-making.

Regarding challenges, finding the right balance between working on the final year project and other modules was demanding. As a joint honour’s student, my technical expertise might not have been as extensive as students focusing on the core computer science programme. However, the knowledge and skills acquired from the web programming module played a crucial role in overcoming these challenges and completing the project successfully.

6.5 Social, Legal & Ethical Issues

In this section, I will discuss the ways in which re-energise considers social, legal, and ethical issues related to its development and use.

From a social perspective, the application encourages energy conservation by providing users with tools and information to help them effectively monitor and manage their energy consumption. Moreover, the discussion forum feature creates a sense of community, raising awareness about responsible energy use and allowing users to share knowledge and support each other in their energy management efforts.

In terms of legal considerations, the application complies with relevant laws and regulations, such as data protection and privacy laws. By hashing passwords and ensuring secure data storage, as well as adhering to the General Data Protection Regulation (GDPR) guidelines, user privacy is maintained, and potential legal issues are avoided.

Ethically, the application is transparent about its data collection and usage, gathering only the necessary information for its functioning and informing users about how their data is being used. The application is designed to be unbiased and focused on the needs of the users, aiming to help them make informed decisions about their energy consumption. A code of conduct guiding the development and operation of the application ensures that user interests are prioritised, and ethical behaviour is promoted throughout its use.

By addressing these social, legal, and ethical issues, my energy management application demonstrates a commitment to aligning with societal values, complying with legal requirements, and upholding ethical standards in its interactions with users.

Chapter 7: **Conclusion**

7.1 Achievements

I am genuinely pleased with the outcome of this project. Throughout the process, I learnt a great deal and was able to develop my knowledge and skills in programming and software engineering. This project allowed me to apply the skills I acquired from various modules during my university career, demonstrating the practical value of the education I have received.

I believe that the final result of the project exceeded my initial expectations, showcasing my ability to create a comprehensive and functional energy management application that caters to users' needs. This experience has not only been a valuable learning opportunity, but it has also been enjoyable, as I genuinely love coding and find it incredibly rewarding to see my work come to life.

Overall, the achievements of this project have been highly satisfying and have contributed to both my personal and professional growth. I am confident that the skills and experiences gained during this project will prove invaluable in my future endeavours within the technological field.

7.2 Future Implementation

Throughout the project process, there were future improvements and features that were identified for the software for a more large-scale environment. These possibilities will be considered in the future development of the system to enhance its capabilities and sustainability. These include:

- **Real-time energy monitoring:** Continuous monitoring of energy usage data in real-time, providing insights on energy usage patterns, peak demand periods, and potential energy savings opportunities.
- **Automated data collection:** Automatic collection of data from energy meters and sensors, reducing the need for manual data collection and improving data accuracy and consistency.
- **Automatic control of energy-consuming devices:** Automatic control of energy-consuming devices such as HVAC systems, lighting systems, and industrial equipment based on energy demand and occupancy patterns, optimising energy usage and reducing energy waste.
- **Integration with renewable energy sources and energy storage systems:** Integration with renewable energy sources such as solar panels and wind turbines, as well as energy storage systems such as batteries, to manage energy usage in a more sustainable and efficient way, maximising the use of renewable energy sources and reducing reliance on the grid.

7.3 Conclusion

Overall, this project provided me with an opportunity to apply my skills and knowledge acquired throughout the modules that I took at Queen Mary University. The project not only allowed for the development of valuable skills and for future projects and career prospects but also proved to be an enjoyable experience. Re-energise successfully satisfied the milestones and targets originally set out and addressed the problem at hand by implementing appropriate solutions and features within the application for users.

References

Annual Fuel Poverty Statistics in England, 2022 (2020 data). [online] Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1056777/annual-fuel-poverty-statistics-lilee-report-2022-2020-data.pdf. [Accessed: 20 Nov. 2022].

Blundell, R., Costa Dias, M., Cribb, J., Joyce, R., Waters, T., Wernham, T., & Xu, X. (2022). Inequality and the COVID-19 Crisis in the United Kingdom. Annual Review of Economics, 14, 607-636.

Brignall, M. (2022). How are UK gas and electricity bills calculated? [online] the Guardian. Available at: https://www.theguardian.com/money/2022/apr/19/how-are-uk-gas-and-electricity-bills-calculated [Accessed 20 Nov. 2022].

Bunney, M. (2021). Where does our 100% green electricity come from? [online] Octopus Energy. Available at: https://octopus.energy/blog/how-we-supply-green-electricity/ [Accessed 20 Nov. 2022].

By The Newsroom (2022). Interest rates expected to rise to highest level in 13 years. [online] Newsletter.co.uk. Available at: https://www.newsletter.co.uk/business/interest-rates-expected-to-rise-to-highest-level-in-13-years-3728633 [Accessed 20 Nov. 2022].

Case Study: Meaker, M. (2022). Amazon Workers in the UK Walk Out in a Cost-of-Living Rebellion. [online] WIRED. Available at: https://www.wired.com/story/amazon-uk-pay-protests/ [Accessed 20 Nov. 2022].

Citizensadvice (2022). Grants and benefits to help you pay your energy bills. [online] Available at: https://www.citizensadvice.org.uk/consumer/energy/energy-supply/get-help-paying-your-bills/grants-and-benefits-to-help-you-pay-your-energy-bills/ [Accessed 20 Nov. 2022].

Cooban, A. (2022) *Why UK energy prices are rising much faster than in Europe | CNN business*, *CNN*. Cable News Network. Available at: https://edition.cnn.com/2022/08/19/energy/energy-prices-uk-europe-explainer/index.html [Accessed 20 Nov. 2022].

Compare the UK's Top Business Energy Suppliers | British Business Energy. (2019). Compare the UK's Top Business Energy Suppliers | British Business Energy. [online] Available at: https://britishbusinessenergy.co.uk/suppliers/ [Accessed 20 Nov. 2022].

Department for Business, Energy & Industrial Strategy (2021). Energy Follow Up Survey: Household Energy Consumption & Affordability. [online] Consumption &

Affordability. BRE. Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1018725/efus-Household-Energy-Consumption-Affordability.pdf [Accessed 20 Nov. 2022].

Department for Business, Energy & Industrial Strategy (2022). Energy Trends. [online] Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1107502/Energy_Trends_September_2022.pdf. [Accessed 20 Nov. 2022].

Hark. (2022). Energy Analytics - Analyse Energy Meter Reading Data in Seconds - Hark. [online] Available at: https://harksys.com/products/energy-tools/ [Accessed 25 Nov. 2022].

Hark. (2022). Why Is Energy Monitoring So Important? - Hark. [online] Available at: https://harksys.com/solutions/energy-monitoring/why-is-energy-monitoring-so-important/ [Accessed 20 Nov. 2022].

HM Treasury (2022). Chancellor delivers plan for stability, growth and public services. [online] GOV.UK. Available at: https://www.gov.uk/government/news/chancellor-delivers-plan-for-stability-growth-and-public-services [Accessed 20 Nov. 2022].

HM Treasury (2022). Cost of living support Factsheet. [online] GOV.UK. Available at: https://www.gov.uk/government/publications/autumn-statement-2022-cost-of-living-support-factsheet/cost-of-living-support-factsheet [Accessed 20 Nov. 2022].

Hui, S. (2022) *Why UK energy bills are skyrocketing*, *PBS*. Public Broadcasting Service. Available at: https://www.pbs.org/newshour/politics/why-uk-energy-bills-are-skyrocketing [Accessed 20 Nov. 2022].

Just Energy (2021). *Kilowatts and Calculations: What You Need To Know?* [online] Just Energy. Available at: https://justenergy.com/blog/kilowatts-and-calculations/ [Accessed 20 Nov. 2022].

Lepage, M. (2020). *Four clever ways to estimate your gas consumption*. [online] Blog of Energyprice.be. Available at: https://www.energyprice.be/blog/calculate-gas-consumption/ [Accessed 20 Nov. 2022].

Monetary Policy Report. (n.d.). [online] Bank of England, Available at: https://www.bankofengland.co.uk/-/media/boe/files/monetary-policy-report/2022/august/monetary-policy-report-august-2022.pdf [Accessed 20 Nov. 2022]

New Economics Foundation - S Arnold, A Harper, A Stirling. (2021). The UK's living standards crisis. [online] Available at: https://neweconomics.org/2021/06/the-uks-living-standards-crisis [Accessed 20 Nov. 2022].

Novi Electrics. (2017). How to calculate electricity - Novi Electrics. [online] Available at: http://www.novielectrics.uk/how-to-calculate-electricity/ [Accessed 20 Nov. 2022].

Office for Budget Responsibility (2022). Economic and fiscal outlook. [online] APS Group. Available at: https://obr.uk/docs/dlm_uploads/CCS0822661240-002_CCS001_SECURE_OBR_EFO_November_2022_BOOKMARK.pdf [Accessed 20 Nov. 2022].

Office for National Statistics. (2022) *Energy prices and their effect on households - Office for National Statistics*. Office for National Statistics. Available at: https://www.ons.gov.uk/economy/inflationandpriceindices/articles/energypricesandtheireffectonhouseholds/2022-02-01 [Accessed 20 Nov. 2022].

Ofgem. (2022). Your energy bill explained. [online] Available at: https://www.ofgem.gov.uk/information-consumers/energy-advice-households/costs-your-energy-bill [Accessed 20 Nov. 2022].

Romatech (2019). *7 ways Smappee Infinity can help reduce energy bills for homes and businesses.* [online] Available at: https://www.romatech.co.uk/blog/page/view/7-ways-Smappee-Infinity-can-help-reduce-energy-bills-for-homes-and-businesses [Accessed 25 Nov. 2022].

Scott, E. (2022) *Cost of living: Impact on public wellbeing - House of Lords Library*, *Cost of living: Impact on public wellbeing*. UK Parliament. Available at: https://lordslibrary.parliament.uk/cost-of-living-impact-on-public-wellbeing/ (Accessed: November 10, 2022).

Smappee. (2022). *Fueling energy efficiency.* [online] Available at: https://www.smappee.com/ [Accessed 25 Nov. 2022].

SoftwareAdvice. (2022). *Wattics*. [online] Available at: https://www.softwareadvice.co.uk/software/30733/wattics [Accessed 25 Nov. 2022].

Steele, P. (2019). *Agile pricing explained*. [online] Octopus Energy. Available at: https://octopus.energy/blog/agile-pricing-explained/ [Accessed 20 Nov. 2022].

Wattics. (2018). Wattics. [online] Available at: https://www.wattics.com/ [Accessed 25 Nov. 2022].

Appendix

Appendix A - Gant Chart

A Gant chart illustrates the project schedule, it can also show dependencies amongst activities and showing the actual schedule of those particular activities. It is not the project plan but rather a planning tool. It is all about milestones and timing of the projects' activities, from start to finish.

TASK	STATUS	START	END
Project Definition & Plan	Completed	9th Oct 2022	17th Oct 2022
Aims, Objectives and Literature Review	Completed	17th Oct 2022	20th Nov 2022
Interim Report	In Progress	1st Nov 2022	28th Nov 2022
Project Report Slides	In Progress	28th Nov 2022	5th Dec 2022
Requirements & Design	Not Started	26th Dec 2022	16th Jan 2023
UI (HTML/CSS) Development	Not Started	23rd Jan 2023	20th Feb 2023
Draft Report	Not Started	20th Feb 2023	20th Mar 2023
DB Development	Not Started	27th Feb 2023	1st Apr 2023
Web-Based Functionality Development	Not Started	13th Mar 2023	24th Apr 2023
Final Testing	Not Started	17th Apr 2023	1st May 2023
Final Report	Not Started	20th Mar 2023	2nd May 2023
Project Video	Not Started	17th Apr 2023	4th May 2023

MONTH	Oct	Oct	Oct	Oct	Nov	Nov	Nov	Nov	Dec	Dec	Dec	Dec	Jan	Jan	Jan	Jan	Jan
DAY	10	17	24	31	7	14	21	28	5	12	19	26	2	9	16	23	30

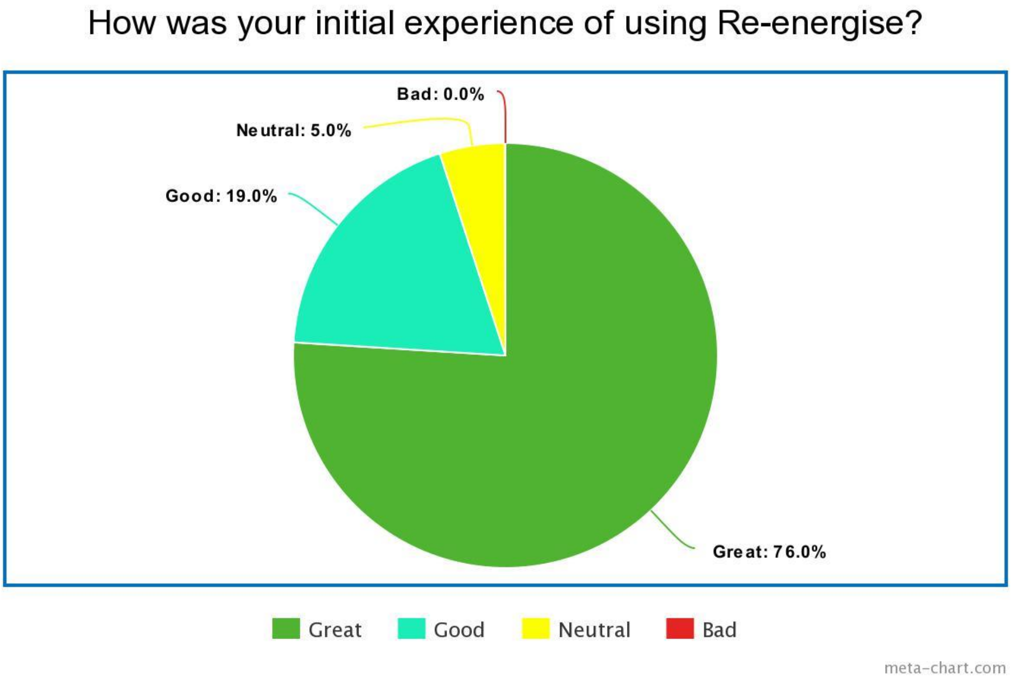
MONTH	Feb	Feb	Feb	Feb	Mar	Mar	Mar	Mar	Apr	Apr	Apr	Apr	May	May
DAY	6	13	20	27	6	13	20	27	3	10	17	24	1	8

Appendix B - Risk Assessment

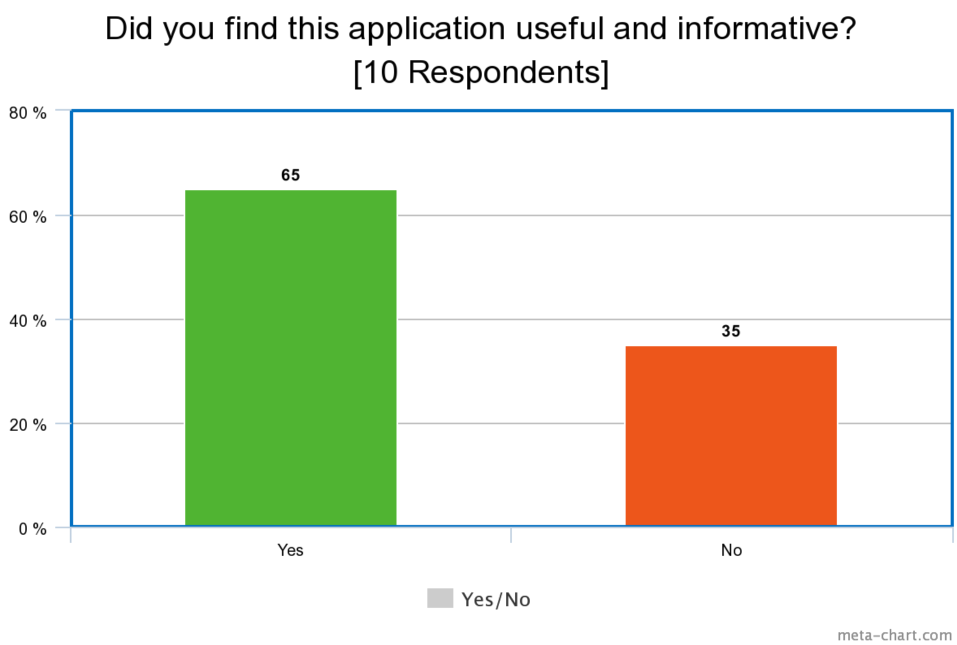
Description of Risk	Impact of Risk	Likelihood Rating	Impact Rating	Preventative Actions
Complexity of Project	Delay in development and coding/unable to complete the project.	Medium	Medium	Start development early, learning key skills needed to complete the project.
Personal illness/emergency	Delay in project timelines	Medium	Medium	Using the key requirements as planning for time management. Adhering to deadlines.
Loss of files	Restart of the whole project or a section of the project	Low	High	Make sure to always keep constant backups locally and externally on cloud
Time Constraints	Procrastinating the project due to other assignments which would lead to tight deadlines	Medium	High	Have a plan in order to manage time appropriately and organise. Balancing other work with the project
Overwhelm and stress	May lead to mental health issues, which in turn impacts on the completion of the project	Medium	Medium	Don’t overwork, take breaks where needed. Communicating with others to overcome difficulties
Unmet Requirements	Incomplete application	Medium	Medium	Ensure to have a structure to the requirements and not overcomplicating
The backend computations are incorrect	Incorrect information displayed to the user	Low	High	Ensure research is done for a properly understanding on backend calculations
Not enough testing	Certain features or minor functionality may crash the application	Low	Medium	Ensure testing is done throughout the development process as well as final testing at the end
Scope Creep	Too many scope changes, leading to added features or change in process	Low	Low	Clearly define requirements and a plan on what the application will do
Not meeting all the criteria’s	Not getting the best possible grade for the project	Medium	Medium	Ensure checking and applying the marking criteria for each development phase

Appendix C - User Participant Testing

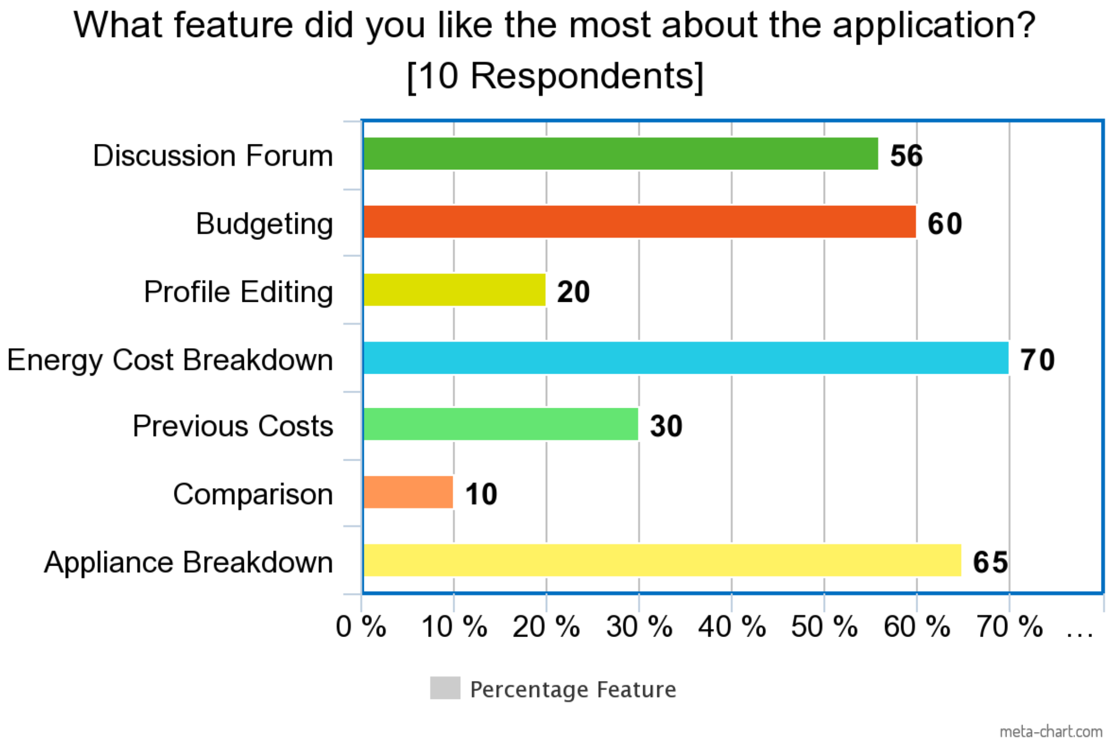
Q1 – How was your initial experience of using Re-energise? [10 Respondents]



Q2 – Did you find this application useful and informative?



Q3 – What feature did you like the most about the application?



Q4 – One a sale from 1 to 5 (1 = very unlikely, 5 = very likely), how likely are you to recommend this application?