
SWITCHACK

S W I T C H A C K

Under Supervision of:

Dr. Sameh Salem Dr. Shahira Habshy

Helwan Engineering 2018

Mohamed Osama Hassan

Mohamed ElSayed ElMorsy

Karim Hashim Abdel Razek

Hesham Gamal Mohamed

Radwa Nabil Ahmed

Jannat Saeed Mohamed

Nada Ahmed Abdel Kader

Alaa Hussien Mohamed

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of my project. All that we have done is only due to such supervision and assistance and I would not forget to thank them. In a try to express our gratitude to our esteemed doctors Dr. Sameh Salem and Dr. Shahira Habashy who both directed us in each step in this project, we're thankful for their guidance, patience, enthusiasm, motivation and their generosity that kept providing us with the guidance we needed.

ABSTRACT

Egyptians across the country are becoming increasingly frustrated with the gross increase of electricity bills in recent months, Where prices increased by an average of 33% for all segments. This problem face almost all the Egyptians and they can do nothing about this increase but paying bills to use the electric power. And from here

our idea came, We intend to minimize electricity usage by knowing electricity usage itself which will help individuals saving the wasted or the unwanted electric power in their everyday.

We propose to solve this critical problem using a user friendly mobile application and a smart Electrometer could be installed at your home besides your electrical panel to measure your usage and give you an estimation on how much your next bill is likely to be, recommend how to economize your usage and you can also determine the power of your device and the usage you do not want to exceed, so the application will recommend the best way to limit it.

TABLE OF CONTENTS

INTRODUCTION

1.1 History & Statistics of the Problem

1.1.1 Electricity consumption in Egypt

Energy statistics is a crucial tool to analyse the energy situation in order to provide the necessary vital information for policy makers, planners and researchers.

The most important measure in the energy balance of Egypt is the total consumption of 150.40 billion kWh of electric energy per year. Per capita this is an average of 1,572 kWh.

Egypt could provide itself completely with self-produced energy. The total production of all electric energy producing facilities is 172 bn kWh, which is 114% of the countries own usage. Despite this, Egypt is trading energy with foreign countries. Along with pure consumptions the production, imports and exports play an important role. Other energy sources such as natural gas or crude oil are also used.

Energy Balance

Eletricity	Total	Egypt per capita	Compared to Europe per capita
Own consumption	150.40 bn kWh	1,571.76 kWh	5,412.25 kWh
Production	171.90 bn kWh	1,796.45 kWh	5,831.12 kWh
Import	43.00 m kWh	0.45 kWh	731.24 kWh
Export	1.16 bn kWh	12.10 kWh	719.23 kWh

Figure 1.0 - Energy Balance

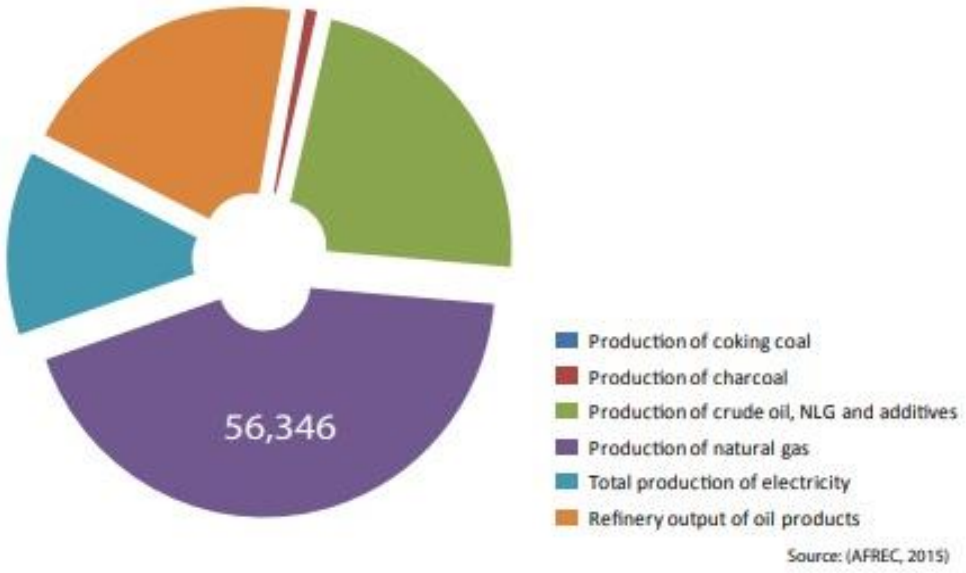


Figure 1.1 Total energy production

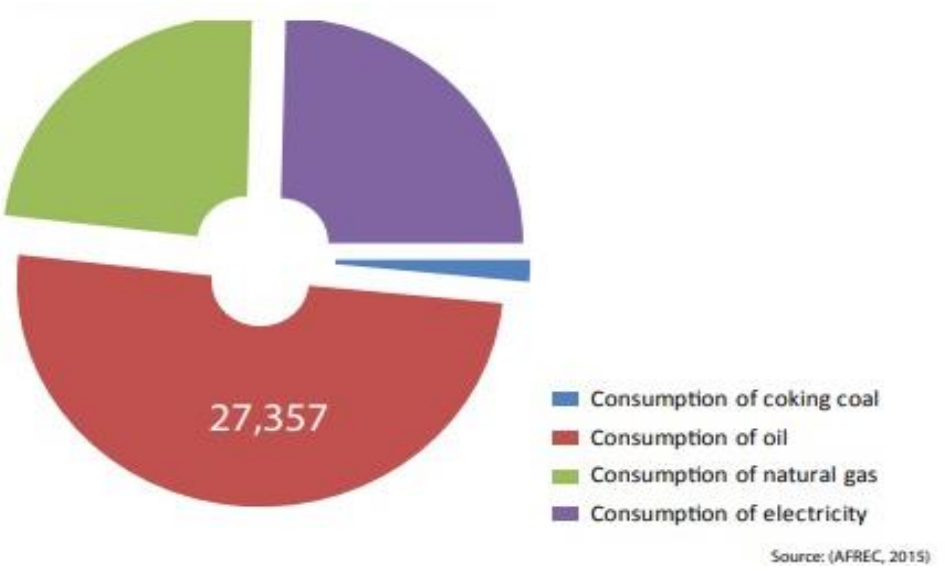


Figure 1.2 Total energy consumption

- Electricity generation increased from 78.1 TWh in 2000/2001 to 156.6 TWh in 2011/2012 (6% yearly).
- Electricity generated by thermal power stations amplified from about 64.3 TWh in 2000/2001 to 141.7 TWh in 2010/2011.
- Electricity generated from hydropower is slightly decreased from 13.7 TWh to 12.9 TWh.
- In addition to 0.137 TWh generated by renewables in 2000/2001 boosted to 1.958 TWh in 2011/2012.

UNDERSTANDING ENERGY USE AT HOME

It's not always easy to understand your monthly electric bill. The first step in energy saving measures is to determine which of your appliances consume the most.

HEATING & COOLING

Heating and cooling devices are often energy-intensive. This is very apparent when looking at the way energy consumption is distributed in the household. As shown in the following diagram, heating and cooling alone (either water or space) represents approximately 45% of the energy used at home.

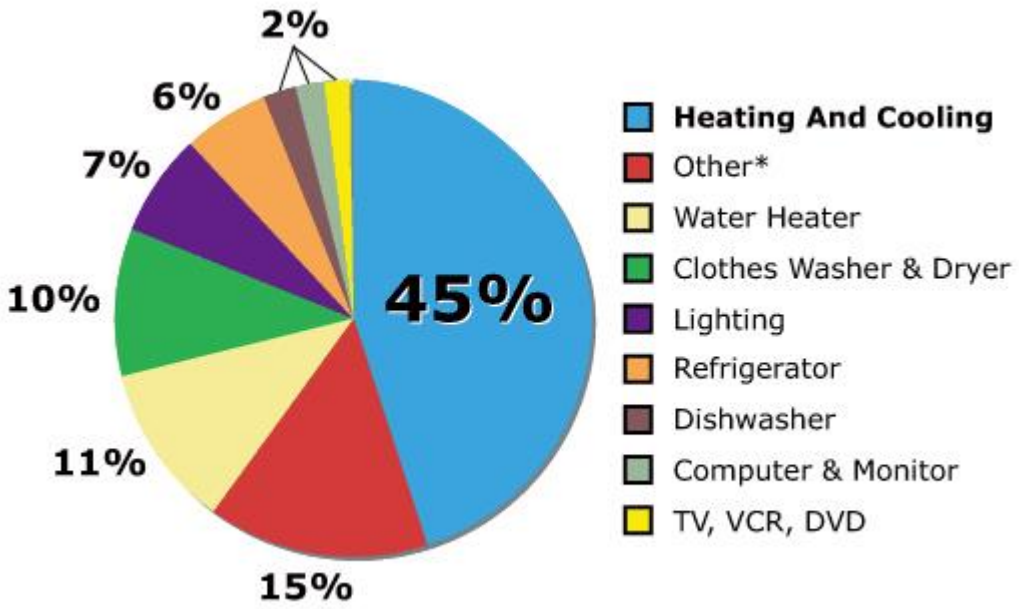


Figure 1.3 - Electricity usage at homes

APPLIANCES, ELECTRONIC DEVICES AND LIGHTING ALSO INCREASE THE BILL

The rest of your energy use goes to various appliances and electronic devices, as well as lighting. The quality and the age of your devices will significantly affect their energy consumption, but you can consume less electricity without replacing them. For example, switch them off instead of leaving them on standby.

Lighting represents more energy than what we may think, often around 7% of our total consumption. This may not seem like much, but changes are easy to apply, simply by investing in greener bulbs, such as LED bulbs.

So, our first problem to solve is the waste of energy where conserving energy is not just about saving on your electricity costs, but also involves being eco-friendly, thereby, protecting the environment. Producing electricity is highly depends on burning oil. This releases an abundant amount of harmful carbon emissions. Using a limited amount of electricity to fulfill one's needs is acceptable, however, it becomes a major issue when we start consuming excess electricity and start wasting it.

1.1.2 Increasing the electricity prices in Egypt

The Electricity Ministry raised the price of electricity by around 42% for household consumers in July, 2017. The move, which follows an increase to the prices of fuel, removes subsidies entirely for the highest-tier consumers. The new prices for household consumers per month are broken down as follows:

- The 0-50 KW consumption bracket will pay EGP 0.13 per KW, up from EGP 0.11

- The 51-100 KW consumption bracket will pay EGP 0.22 per KW, up from EGP 0.19
- The 100-200 KW consumption bracket will pay EGP 0.22 per KW, up from EGP 0.19
- The 200-350 KW consumption bracket will pay EGP 0.55 per KW, up from EGP 0.45
- The 350-650 KW consumption bracket will pay EGP 0.75 per KW, up from EGP 0.55
- The 650-1,000 KW consumption bracket will pay EGP 1.25 per KW, up from EGP 0.95
- The 1,000+ KW consumption bracket will pay EGP 1.35 per KW, up from EGP 0.95

The prices of electricity for industrial and commercial use were also increased across the different brackets, Al Mal reports. Heavy industries will be paying a high of EGP 0.654 per KW during peak hours and EGP 0.982 per KW in the off-peak. The monthly tariff for commercial usage changed across five brackets as follows:

- The 0-100 KW consumption bracket will pay EGP 0.45 per KW, up from EGP 0.35
- The 100-250 KW consumption bracket will pay EGP 0.84 per KW, up from EGP 0.69
- The 250-600 KW consumption bracket will pay EGP 0.96 per KW, up from EGP 0.69
- The 600-1,000 KW consumption bracket will pay EGP 1.35 per KW, up from EGP 0.96
- The 1,000+ KW consumption bracket will pay EGP 1.40 per KW, up from EGP 0.96.

The state will continue to subsidize power for another five years until 2022 rather than eliminate them by 2019 as had been originally planned "owing to the conditions related to the big increase in the exchange rate," the minister also confirmed. This latest increase is expected to save the state around EGP 30 bn in energy subsidies, bringing the bill down to EGP 52.8 bn in the FY 2017-18. Customers in the highest consumption tiers will be paying a higher price for power than its cost of production of EGP 0.97 per KW and the profit generated will be used partially to cover the subsidy for the lowest consumption bracket, Shaker had said previously.

1.2 Monitoring Devices Survey

The idea of monitoring devices has been there the whole time but few took the action to bring the concept on the ground, the following survey go through the monitoring devices projects that have been developed to this day.

1.2.1 Energy Consumption Analyzer

This is an application on Google play (fig 1.4), keeps track of your energy consumption. You can add your meters for gas, electricity, or water to the database and record the current meter readings from time to time. Readings can be color-coded and comments may be added to remember special situations which may explain unusual energy usage.

Users have to add their measurements manually which waste time and effort.

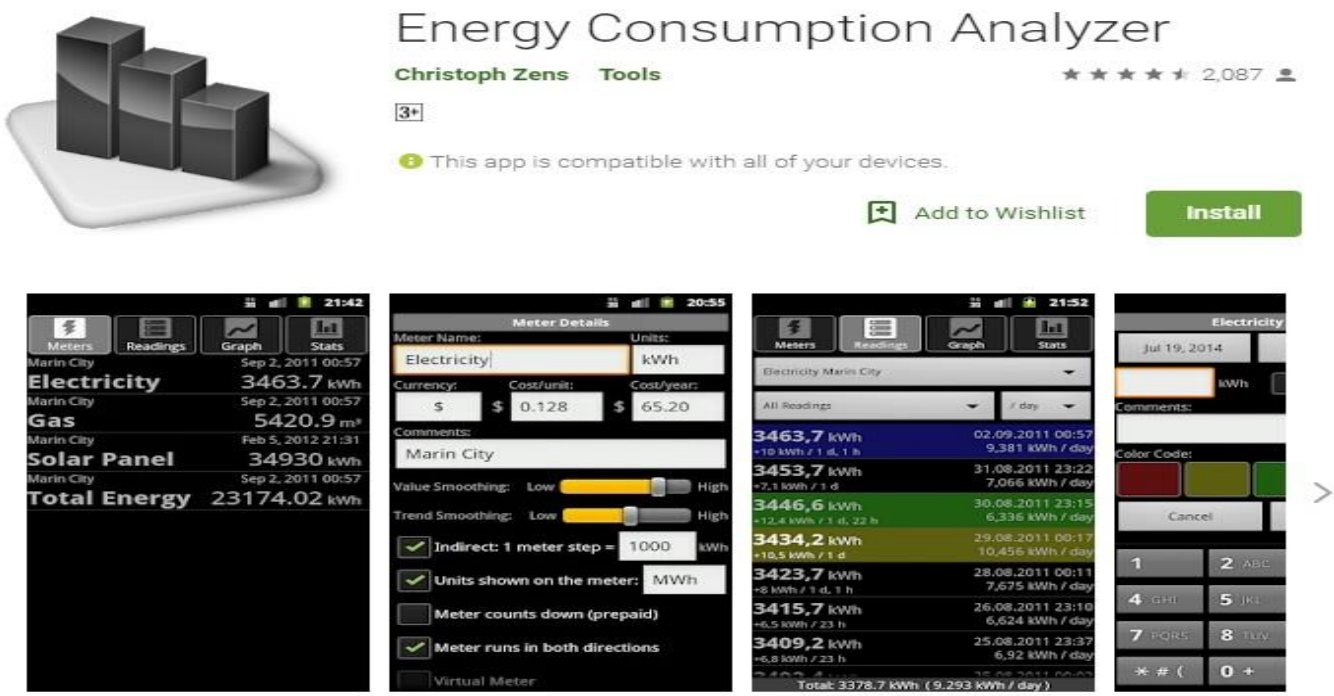


Figure 1.4 - Energy consumption analyzer application

1.2.2 Smappee:

With the Smappee energy monitor (fig. 1.5) you always know how much power your appliances consume, wherever you are. For more comfort, a better insight and sustainable energy savings up to 30%.

Users can not control their usage remotely. It’s just an energy monitor.

Figure 1.5 Smappee device

1.3 Problem Statement & Motivation

Looking at the history and statistics of electricity waste and electricity prices, one thing is obvious, This problem face almost all the Egyptians and they can do nothing about this increase but paying bills to use the electric power. And from here our idea came, We intend to minimize electricity usage by knowing electricity usage itself which will help individuals saving the wasted or the unwanted electric power in their everyday.

Also by looking to the survey, the following point concluded:

1. Some competitors are just software applications, mobile or web, Users have to add their measurements manually which waste time and effort.
2. Some competitors are just monitoring devices without any controlling features.
3. Most of competitors are only available for android users.

So what we are willing to implement differently that the system is consists of two parts, a small device installed beside your electrometer and measure the current, send it to a mobile application which is available for both android and IOS users, this application helps you to monitor your usage, control some fuses and devices, estimate your next bill and recommend how to reduce your usage.

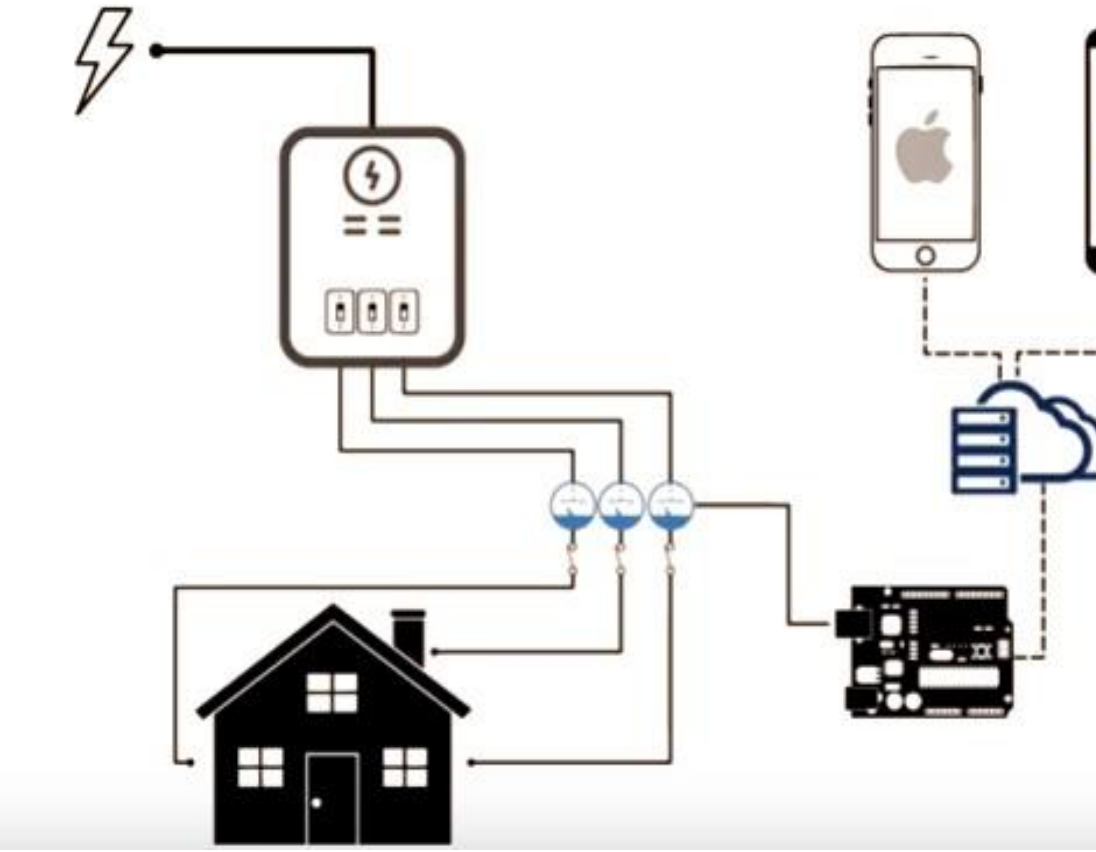


Figure 1.6 System context diagram (SCD)

1.4 Objectives & Benefits:

The objective of creating this system is to measure the consumed current in your home, monitor your usage, control some fuses and devices in your home, estimating your electrical bills and recommend how to economize your usage trying to solve the problem of raising the electricity prices and wasting the energy.

1.5 Project Phases:

Phase One: Starting Point (20 days starting from september-2017):

Team formation and we have searched for ideas then we got 15 ideas, we filtered them into 5 ideas then picked the most commercial one “Monitoring devices” with our supervisors and with agreement from each team member to help us in our vision of entering competitions and starting our startup after our graduation year.

Phase Two: Idea survey (10 days till october 2017):

We searched about projects concerned with Monitoring devices and its features and specifications and we reached a conclusion upon the main features we want to add in SWITCHACK.

Phase Three: Gaining knowledge (4 weeks from October till November):

We started to gather further information about electricity and how we can read current efficiently and each member in the team started courses that can serve our project. By the end of the 4 weeks we can successfully read the current.

Note: Time for midterm exams is considered.

Phase Four: Implementation (November 2017-February 2017):

First prototype and searching for the components we need in our prototype and getting gathered research and data. After having an overview about the first prototype and how it will be done and its function We divided our team, each team was responsible for a specific task (Hardware implementation and design - Android Development - IOS development) and worked in parallel sub-teams according to the timeline in our plan. The Implementation of the hardware, connection with internet through firebase and the software design are done. After finishing each milestone in the project we got back to our supervisors to get feedback.

Note: Time for Final exams is considered.

Phase Five: Complete first Prototype & prepare for competitions(4 weeks till 20 February):

Finished the remaining work on the first prototype and achieved a satisfied percentage of requirements.The team also participate in MIE competition and Maker Hackathon cairo competition, and we achieved the **3rd place** in **Maker Hackathon**.

Phase Six: Complete final prototype & Documentation (March 2017- June 2017):

Depending on feedbacks from mentors, we worked on developing our prototype, adding new features, data analysis and improving accuracy. We also finished the documentation.

Note: Time for Final exams is considered.

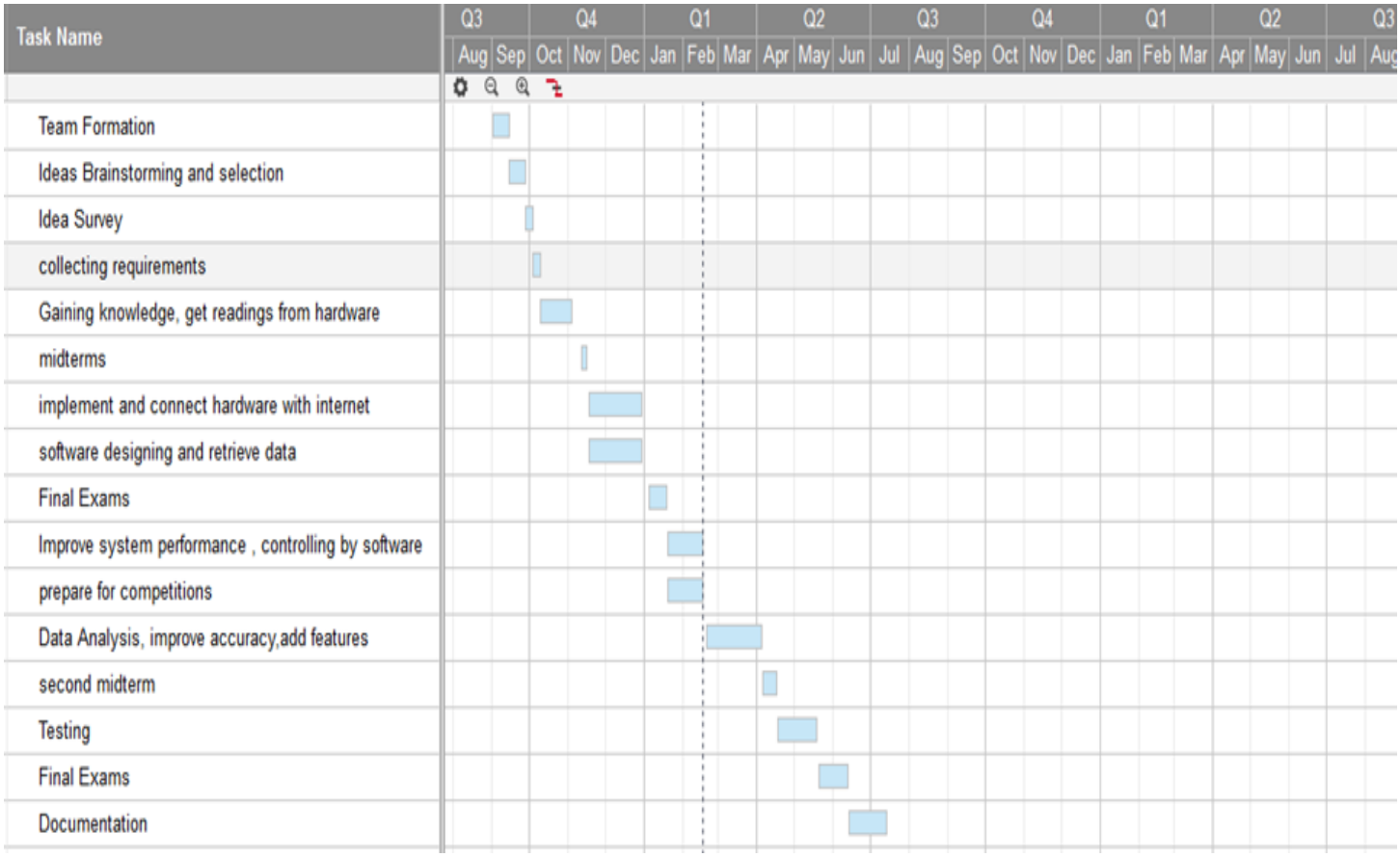


Figure 1.7 - Project Gantt chart

1.6 Document Contents:

This document is organized as follows:

Chapter 2 includes a background that will help the reader to be familiar with all the concepts they will face throughout this book.

Chapter 3 includes the architecture of the system, the hardware design and the software design and modules of the system.

Chapter 4 includes the results reached in each module in the system.

Chapter 5 concludes what’ve been achieved in the system and the improvement that can be applied in the future on the system to make it more effective.

BACKGROUND

2.1 Hardware

2.1.1 IOT:

The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

In the Internet of Things, can be a person with a heart monitor implant, a farm animal with a biochip transponder, an automobile that has built-in sensors to alert the driver when tire pressure is low -- or any other natural or man-made object that can be assigned an IP address and provided with the ability to transfer data over a network.

The problem is, people have limited time, attention and accuracy all of which means they are not very good at capturing data about things in the real world. If we had computers that knew everything there was to know about things using data they gathered without any help from us we would be able to track and count everything and greatly reduce waste, loss and cost. We would know when things needed replacing, repairing or recalling and whether they were fresh or past their best. Practical applications of IoT technology can be found in many industries today, including precision agriculture, building management, healthcare, energy and transportation.

2.1.2 How does it work?

Devices and objects with built in sensors are connected to an Internet of Things platform, which integrates data from the different devices and applies analytics to share the most valuable information with applications built to address specific needs. These powerful IoT platforms can pinpoint exactly what information is useful and what can safely be ignored. This information can be used to detect patterns, make recommendations, and detect possible problems before they occur.

2.2. Developing for Android and IOS:

The two most popular mobile operating systems are Android and IOS. There is debate on which is truly the most popular but mostly all sources agree these are the two market leaders. The two top companies for gathering data on the popularity of mobile operating systems are StatCounter and Net Market Share. Net Market Share gathers its statistics by measures total traffic and StatCounter measures daily unique users (Bott, 2014). Statistics from; (Net Market Share, 2015), (StatCounter, 2015a), are shown in Figure

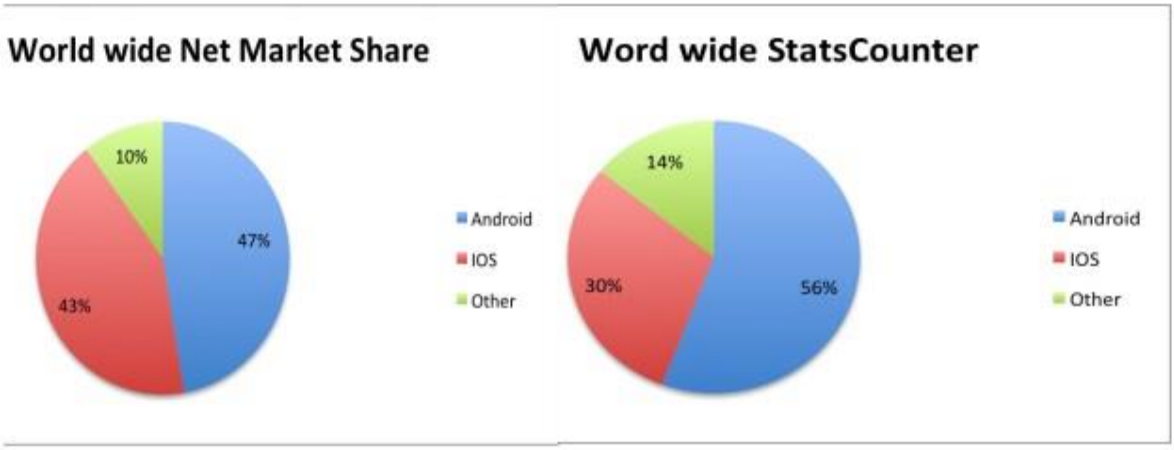


Figure 2.0 - Statistics from Net market share and StatCounter

Operating System	4Q16 Units	4Q16 Market Share (%)	4Q15 Units	4Q15 Market Share (%)
Android	352,669.9	81.7	325,394.4	80.7
iOS	77,038.9	17.9	71,525.9	17.7
Windows	1,092.2	0.3	4,395.0	1.1
BlackBerry	207.9	0.0	906.9	0.2
Other OS	530.4	0.1	887.3	0.2
Total	**431,539.3**	**100.0**	**403,109.4**	**100.0**

Figure 2.1- Comparison of 2015's Q4 market to that of 2016 - The Verge

We aim to release a user-friend ease running application for both android and ios to make sure that all individuals that want to control their electricity usage all are well served.

2.2.1. Android:

Android is claimed to be the most popular mobile operating system worldwide and provides a very open marketplace to distribute apps (Android, 2015). Android has many devices in which its operating system runs on from various companies such as Samsung, HTC, Motorola, ASUS and many others. The devices created by these companies operate on many different screen sizes and versions of the operating systems. A download is available from Google Play's support page that gives an A-Z list of all the devices that Android supports, the number of devices is approximately

8,725 (Google, 2015b). Due to there being a great deal of devices, testing can be a long process when developing for Android.

Google Play is Android's main source of mobile application distribution although Android does not restrict developers to publish Android applications on other application stores. To publish an application, a registration fee of $25 is required. This is a one time fee per account and allows developers to publish their applications on the Google Play store. Google allows Android applications to be uploaded by developers on to the Play Store in short time, usually the same day as it was published (Google, 2015a). The applications are scanned automatically for malicious and poorly implemented software as they are uploaded to maintain quality applications on the application store (Mills, 2012). This is a major benefit to developers as the software can be released very quickly, therefore minimizing the gap between development and release. As a result, this lowers the cost of development especially for companies that hire many developers for a particular application.

2.2.2. IOS:

The usage of IOS devices is prominent in the UK with over 50% of data traffic on IOS (StatCounter, 2015a). In comparison to Android there are far fewer device variants on IOS, various versions of iPad, iPhone, iPod Touch in which under 20 devices run IOS 8.0. As there is such a small number of devices testing is far quicker as well as the there is an emulator for each device build into the IDE that Apple provides. The Apple App store is the only place where applications can be downloaded to mobile devices. The development for IOS is much more restricted than Android.

To publish an application to the App store an annual fee of $99 is required to enrol on the developer program. This then gives the developer a certificate to authorise the apps they develop. Once the application is developed it has to go through the review process. This takes an average of 8 days according to the crowd sourced site (Shiny Development, 2015) as of 14th of April. If the application is not rejected it then will be released on to the application store, although if it is rejected the entire review process has to be repeated once the issue is resolved (Apple, 2015a). In comparison to Android the release time can take much longer as if an app especially if some guidelines are missed during development.

The reason for Apple doing this is to ensure the quality of the applications on its App store to keep the companies good reputation. This can be frustrating to developers and seriously impact development times but ensures only higher quality applications can be released.

2.3 Design patterns:

What is a design pattern?

In software engineering, a design pattern is a general repeatable solution to a commonly occurring problem in software design. A design pattern is not a finished

design that can be transformed directly into code. It is a description or template for how to solve a problem that can be used in many different situations.

Why are design patterns important?

For a very long time there was a serious problem in software engineering: you hire a newcomer to a project and no matter how well they know the programming language, it takes them months to get up to speed with how things are done in your project before they can be productive. In hardware engineering, they solved this problem a very long time ago. They have a common terminology called 'schematic diagrams'. You hire a hardware engineer, give them the schematics of your hardware project in the morning, allow them to study it, and by evening before it is time to call it a day they can pick up the soldering gun and become productive. The software industry has been trying to come up with ways to improve this. Standardization of programming languages was one way. Standard libraries (class libraries nowadays) has been another way; however, one of the most important ways has perhaps been design patterns. So, are design patterns important? Very.

Java Design Pattern Categories

Java Design Patterns are divided into three categories: creational, structural, and behavioral design patterns.

Creational Design Patterns:
Creational design patterns provide solutions to instantiate an object in the best possible way for specific situations.

Singleton patterns:
Singleton patterns, the most popular pattern, restricts the instantiation of a class and ensures that only one instance of the class exists in the java virtual machine.

Factory pattern:
Factory pattern is used when we have a super class with multiple subclasses and, based on input, we need to return one of the sub-classes. This pattern removes the responsibility of instantiation of a class from the client program to the factory class.

Abstract Factory Pattern:
Abstract Factory pattern is similar to Factory pattern and it's factory of factories. If you are familiar with factory design pattern in java, you will notice that we have a single Factory class that returns the different sub-classes based on the input provided, and factory class uses if-else or switch statement to achieve this. In Abstract Factory pattern, we get rid of the if-else block and have a factory class for each subclass, and then an Abstract Factory class that will return the sub-class based on the input factory class.

Builder Pattern:

This pattern was introduced to solve some of the problems with Factory and Abstract Factory design patterns when the Object contains a lot of attributes. Builder pattern solves the issue with large numbers of optional parameters and an inconsistent state by providing a way to build the object step-by-step and provide a method that will return the final Object.

Prototype Pattern:

Prototype pattern is used when the Object creation is a costly affair, requires a lot of time and resources, and you have a similar object that already exists. This pattern provides a mechanism to copy the original object to a new object, and then modify it according to your needs. This pattern uses java cloning to copy the object. Prototype design pattern mandates that the Object which you are copying should provide the copying feature. It should not be done by any other class; however, whether to use shallow or deep copy of the Object properties depends on the requirements and is a design decision.

Observer pattern:

Observer pattern is used when there is one-to-many relationship between objects such as if one object is modified, its dependent objects are to be notified automatically. Observer pattern falls under behavioral pattern category.

2.4. Database Software

In today's day and age words like 'mobile application' and 'cloud' are creating quite a stir. Each day there is some new application in the market with the aim of delivering the best performance to their users. Mobile backend services help achieve this by building a better infrastructure.

What is BaaS?

These days most mobile applications need a backend that's connected to the internet. BaaS (Backend-as-a-Service) is the service that helps enable that, helping you manage the centralized database and allowing your users to share content via the cloud. BaaS provides the backend for mobile applications, an API, and tools for various programming languages to integrate with their application backend.

As opposed to SaaS (Software-as-a-Service) that is targeted to end users, BaaS is targeted at developers.

Also scaling apps is difficult as and when your audience grows, to do this effectively, you need a backend. Earlier backends needed to be developed from scratch but owing to technological advancements, you can now directly integrate them into your application.

Types of BaaS for Mobile Apps

There are two types of Backend-as-a-Service (BaaS) namely Consumer BaaS and Enterprise BaaS.

Consumer BaaS: This focuses more on lightweight applications and games.

Enterprise BaaS: This focuses more on mobilizing sensitive and critical business data.

When it comes to different BaaS services, there are several leading solutions like Amazon Web Services (AWS), Kinvey, Google Firebase and Apple CloudKit.

We choose to use **Firebase.**

Why Firebase?

Firebase was an independent service before being acquired and launched by Google as a part of 2016 UI. Firebase aims to help web developers in building better apps and thereby helping them grow in their business, by taking care of the backend of the applications. Here are some features that help in determining why Firebase is the best BaaS available:

1. Platform: Firebase provides a real-time database platform that writes events as and when it occurs. It also provides database management API.
2. Front-End: Firebase runs on Angular, Ember, Vue.js and React on the front end.
3. Hosting: It includes features like custom domains for free, Global CDN, Users, DB and Auto-provisioned SSL certificates.
4. Clean Dashboard: A strong design is another feature of Firebase, the dashboard is clean and lists all the features on the left sidebar.
5. Analytics: Firebase has a free and built-in Analytics for all Firebase apps, which allows developers to see how ad campaigns are performing, log events to track username and age, view active users in your application etc.

Conclusion

With features like Cloud Messaging, Remote Config, and a Real-Time Database, Firebase scores points. Choosing the right BaaS is crucial and it comes down to what your project requires.

2.5. Machine Learning (Regression Analysis)

Regression is one of the most important and broadly used machine learning and statistics tools out there. It allows you to make predictions from data by learning the relationship between features of your data and some observed, continuous-valued response. Regression is used in a massive number of applications ranging from predicting stock prices to understanding gene regulatory networks.

2.5.1. Why do we use Regression Analysis?

There are multiple benefits of using regression analysis. They are as follows:

1. It indicates the significant relationships between dependent variable and independent variable.
2. It indicates the strength of impact of multiple independent variables on a dependent variable.

Types of Regression Analysis

There are two types of methods commonly used for regression :

1. Linear Regression

Linear regression tries to fit points to a line generated by an algorithm. This optimized line is capable of predicting values for certain input values and can be plotted.

2. Logistic Regression

Logistic regression comes from the fact that linear regression can also be used to perform classification problem but the logistic regression is not linear. The idea of logistic regression is to make linear regression produce probabilities. It's always best to predict class probabilities instead of predicting classes. Unlike Linear regression which calculate a linear function and then a threshold in order to classify.

2.5.2 How does it work?

2.5.2.1 Cost function

We want to set the parameters in order to achieve a minimal difference between the predicted and the real values.
We make a hypothesis that defines our predicted output in terms of input features and other parameters called weights.
consider we have just one feature for clarification, then our hypothesis will be as follows:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Where
h(x) : Predicted output.

x : input feature.
theta : adjustable weights to achieve higher accurate predicted output.

We can measure the accuracy of our hypothesis function by using a cost function. This takes an average difference (actually a fancier version of an average) of all the results of the hypothesis with inputs from x's and the actual output y's.

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2 = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$$

So, how we could minimize this cost Function?

There are a variety of algorithms that reduce a cost function. One of the most famous algorithms is Gradient Descent.

2.5.2.2 Gradient Descent

When there are one or more inputs you can use a process of optimizing the values of the coefficients by iteratively minimizing the error of the model on your training data.

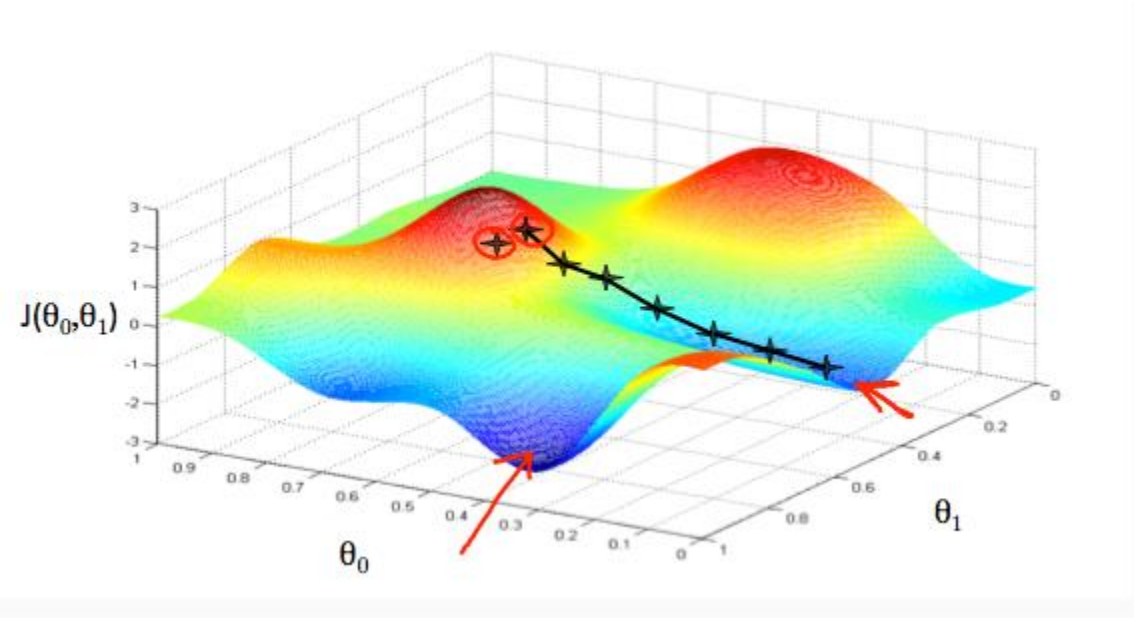
This operation is called Gradient Descent and works by starting with random values for each coefficient. The sum of the squared errors are calculated for each pair of input and output values. A learning rate is used as a scale factor and the coefficients are updated in the direction towards minimizing the error. The process is repeated until a minimum sum squared error is achieved or no further improvement is possible.When using this method, you must select a learning rate (alpha) parameter that determines the size of the improvement step to take on each iteration of the procedure.

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

Where:

Theta j : a specific weight.
Alpha : Learning rate.
J(theta) : our cost function.

Imagine that we graph our hypothesis function based on its fields theta_0 and theta_1 (actually we are graphing the cost function as a function of the parameter estimates). We put theta_0 on the x axis and theta_1 on the y axis, with the cost function on the vertical z axis. The points on our graph will be the result of the cost function using our hypothesis with those specific theta parameters.
The graph below depicts such a setup.



We will know that we have succeeded when our cost function is at the very bottom of the pits in our graph, when its value is the minimum. The red arrows show the minimum points in the graph.

The way we do this is by taking the derivative (the tangent line to a function) of our cost function. The slope of the tangent is the derivative at that point and it will give us a direction to move towards. We make steps down the cost function in the direction with the steepest descent.

When specifically applied to the case of linear regression, a new form of the gradient descent equation can be derived, where m is the size of the training set. All parameters must be updated simultaneously.

repeat until convergence: {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} ((h_\theta(x_i) - y_i) x_i)$$

}

Choosing the value of alpha is crucial. If it is too small the algorithm will be slow, if it is too large it will fail to converge.

From that we get a general form to apply that on multiple features

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)} \qquad \text{for j} := 0...n$$

2.5.2.3 Feature Scaling and Mean Normalization:

To make sure that all values of features are on a same scale and have the same mean it's necessary to use Feature Scaling and Mean Normalization.

Feature scaling involves dividing the input values by the range (i.e. the maximum value minus the minimum value) of the input variable, resulting in a new range of just 1. Mean normalization means that features will be rescaled so that they'll have the properties of a standard normal distribution with mean = 0, standard deviation =1

$$z = \frac{x - \mu}{\sigma}$$

.

2.5.2.4 Learning rate

To choose a suitable learning rate, gradient descent has to be plotted and "debugged".

Make a plot with number of iterations on the x-axis. Now plot the cost function, J(θ) over the number of iterations of gradient descent. If J(θ) ever increases, then you probably need to decrease α.

If J(θ) stops to decrease significantly in an iteration step convergence can be declared.

PROPOSED SYSTEM

Switchack is small device installs in your home's electrical panel to measure the current exactly like your electrical meter but with more details, because it measures the current passes through each fuse. Switchack also connects your electrical panel to your mobile phone through our Android/IOS mobile application, so you can monitor the usage of each room (Fuse) and know how much energy you consume, you can ON/OFF some devices wherever you are. Switchack also estimates your bills and recommends how to reduce your usage to reduce your bills.

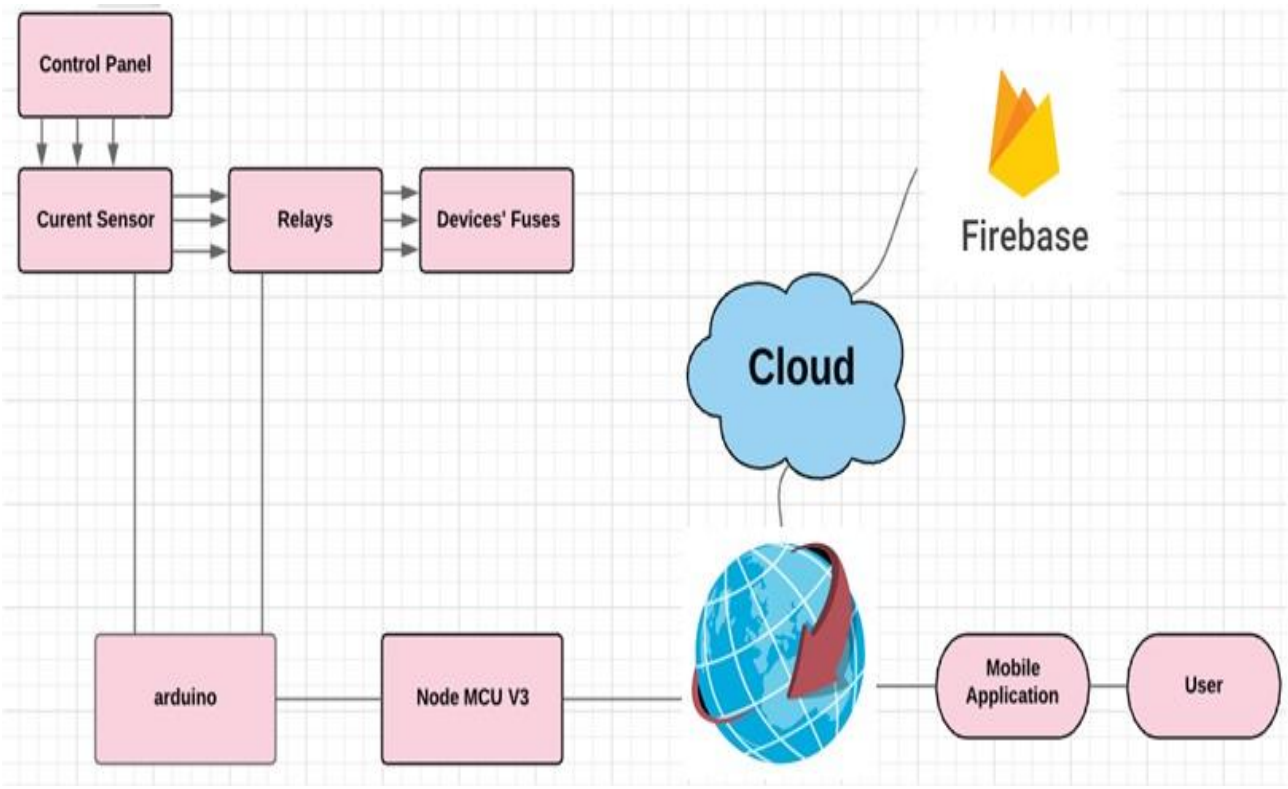


Figure 3.0 - System block diagram

3. System Design

3.1. Hardware Module

3.1.1 Hardware Functions:

- Measure consumed current.
- Send readings to cloud database (Firebase).
- Controlling Fuses (and some devices).

3.1.2. Hardware components:

(A) Arduino

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on a computer, used to write and upload computer code to the physical board.

The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board – you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.

(B) NodeMCU

The NodeMcu is an open-source firmware and development kit that helps you to prototype your IOT product within a few script lines.

Features:

- Open-source
- Interactive
- Programmable
- Low cost
- Simple
- Smart
- WI-FI enabled

Advantages:

Less than $2 WIFI MCU ESP8266 integrated and easy to prototyping development kit. It provides the best platform for IOT application development at the lowest cost.

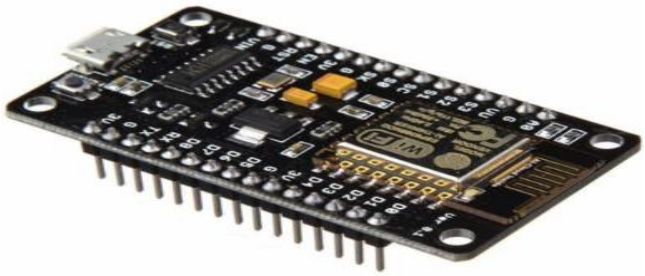


Figure 3.1 - NodeMCU

(C) ACS712 Current Sensor module

The ACS712 Module uses the famous ACS712 IC to measure current using the Hall Effect principle. The module gets its name from the IC (ACS712) used in the module, so for you final products use the IC directly instead of the module.

These ACS712 module can measure current AC or DC current ranging from +5A to -5A, +20A to -20A and +30A to -30A. You have to select the right range for your project since you have to trade off accuracy for higher range modules. This modules outputs Analog voltage (0-5V) based on the current flowing through the wire; hence it is very easy to interface this module with any microcontroller. So if you are looking for a module to measure current using a microcontroller for you project then this module might be the right choice for you.

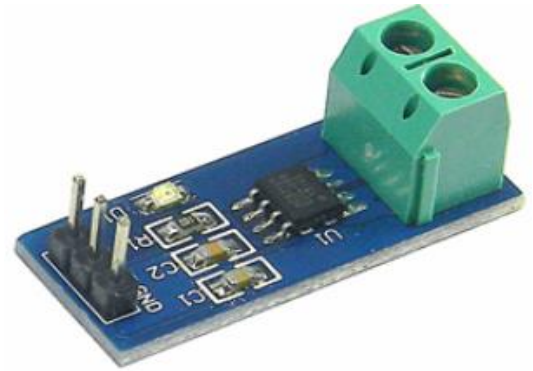
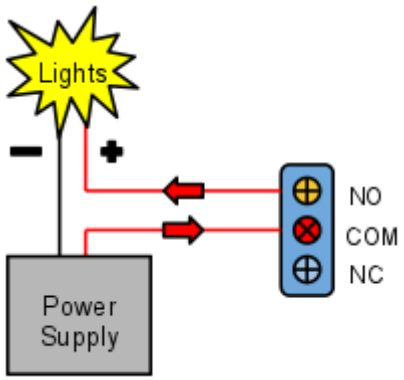


Figure 3.2 - ACS712 Current Sensor Module

(D) Relay module

Relays work on electromagnetism, When the Relay coil is energized it acts like a magnet and changes the position of a switch. The circuit which powers the coil is completely isolated from the part which switches ON/OFF, This provides electrical isolation. This is the reason we can control a relay using 5V's from an arduino and the other end of it could be running an 230V appliance, the 230V end is completely isolated from the 5V arduino circuitry.



(E) Voltage Sensor

Through discrete components and voltage divider we implement a voltage sensor to measure the actual voltage value delivered at homes.

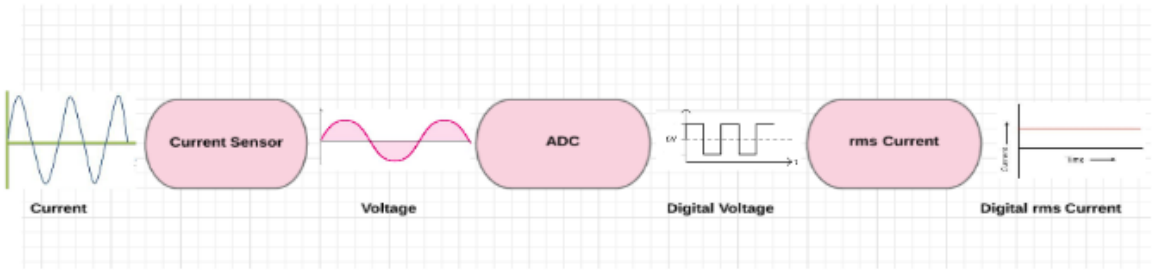
Used components :

1. Resistors (100k ohm , 1k ohm)
2. Capacitor (1uf)
3. Zener Diode
4. Diode

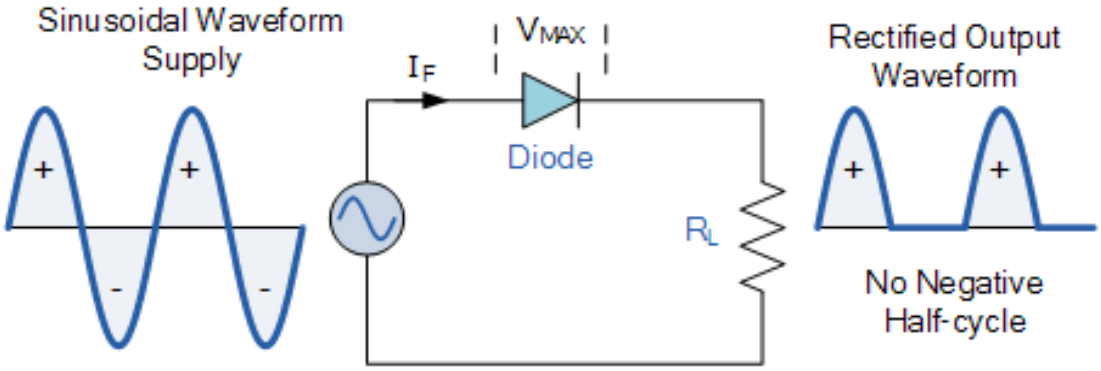
3.1.3 Hardware Implementation

(A) Measure AC Voltage:

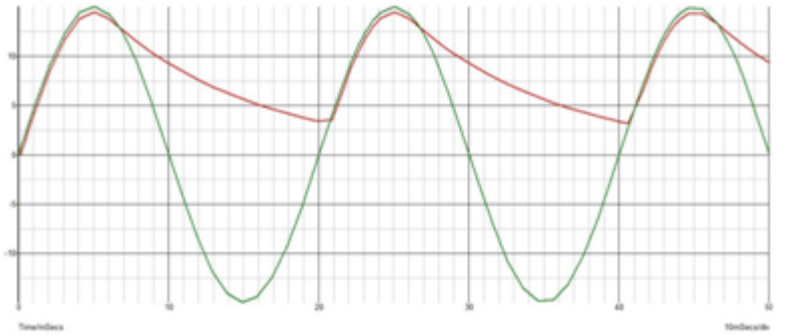
1. Reduce the ac voltage to acceptable level less than 5 volt (using voltage divider).



2. half wave rectifier to take only positive cycle.



3. Capacitor is connected to smooth out the ac signal because it contain large ripples.

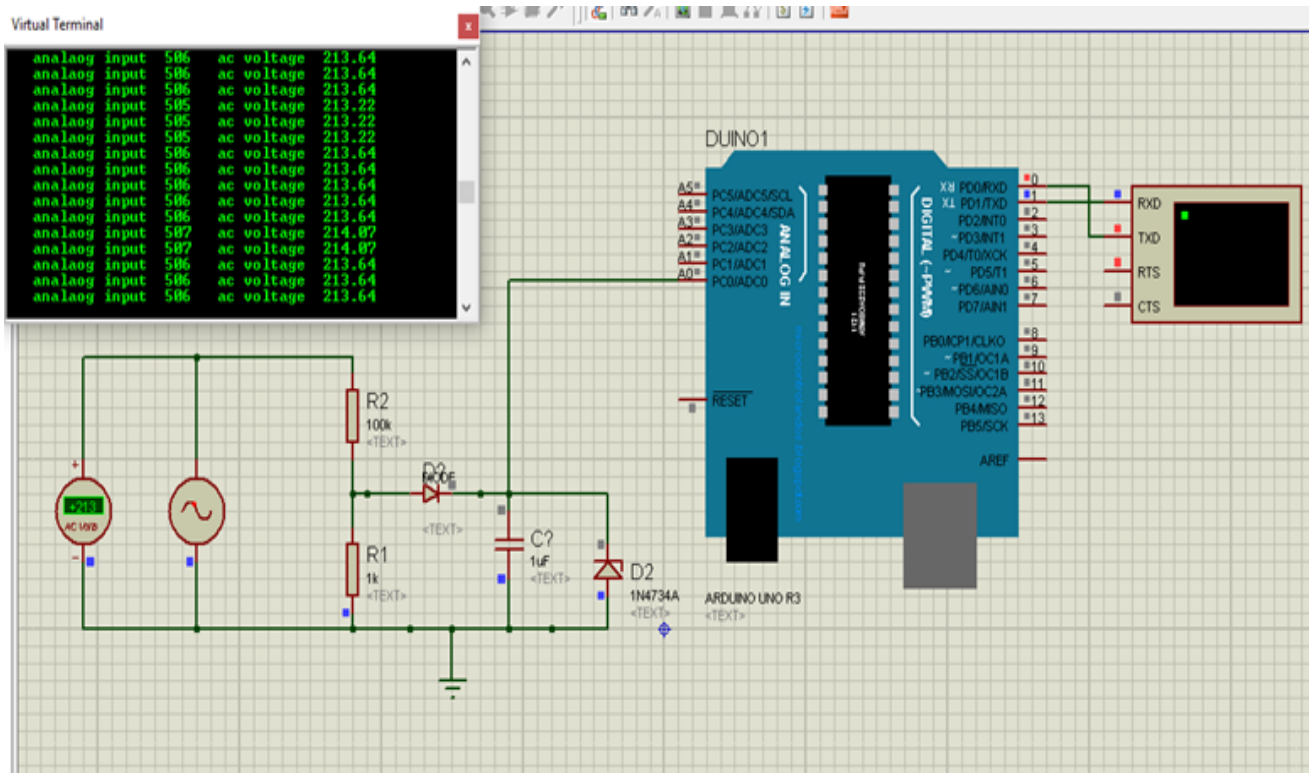


Input and Output Waveforms

4. Calculate voltage with Arduino Read analog value at pin A0

Ac Volt = analog value * 429 / 1024

```
int m;// initialise variable m
float n;//initialise variable n

void setup()
{
  pinMode(A0,INPUT); // set pin a0 as input pin
  Serial.begin(9600);// begin serial communication between arduino and pc
}

void loop()
{
   m=analogRead(A0);// read analog values from pin A0 across capacitor
   n= m * .4189;// converts analog value(x) into input ac supply value using this formula ( explained in woeking section)
   Serial.print("   analaog input  " ) ; // specify name to the corresponding value to be printed
   Serial.print(m) ; // print input analog value on serial monitor
   Serial.print("   ac voltage  ") ; // specify name to the corresponding value to be printed
   Serial.print(n) ; // prints the ac value on Serial monitor
   Serial.println();

}
```



(B) Measure Power Factor

What is power factor?

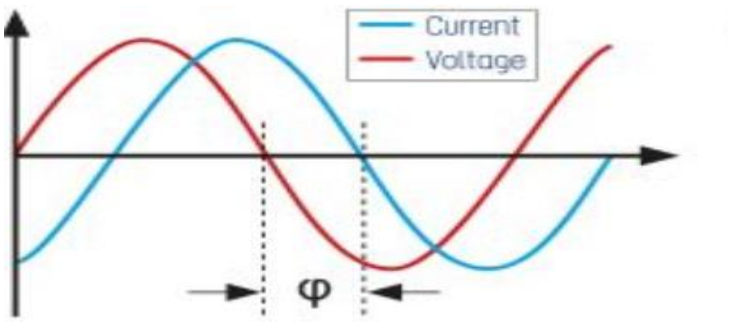
Power factor is defined as the ratio of the real power flowing to the load to the apparent power in the circuit, and is a dimensionless number in the closed interval of −1 to 1.

A power factor of less than one means that the voltage and current waveforms are not in phase. Real power is the capacity of the circuit for performing work in a particular time apparent. power is the product of the current and voltage of the circuit. In an electric power system, a load with a low power factor draws more current than a load with a high power factor for the same amount of useful power transferred. The higher currents increase the energy lost in the distribution system, and require larger wires and other equipment. Because of the costs of larger equipment and wasted energy, electrical utilities will usually charge a higher cost to industrial or commercial customers where there is a low power factor.

How to measure power factor?

Power factor = cos (Φ)
Firstly calculate phase shift between current and voltage

1. Calculate the time difference (td) between two waves and calculate period = 1/frequency.
2. Phase shift (Φ) = td *360 / period
3. Then calculate power factor = cos (Φ)



3.2. Software Module

3.2.1. Architecture:

Model–view–controller (MVC) is an architectural pattern commonly used for developing user interfaces that divides an application into three interconnected parts. This is done to separate internal representations of information from the ways information is presented to and accepted from the user.
The MVC design pattern decouples these major components allowing for efficient code reuse .

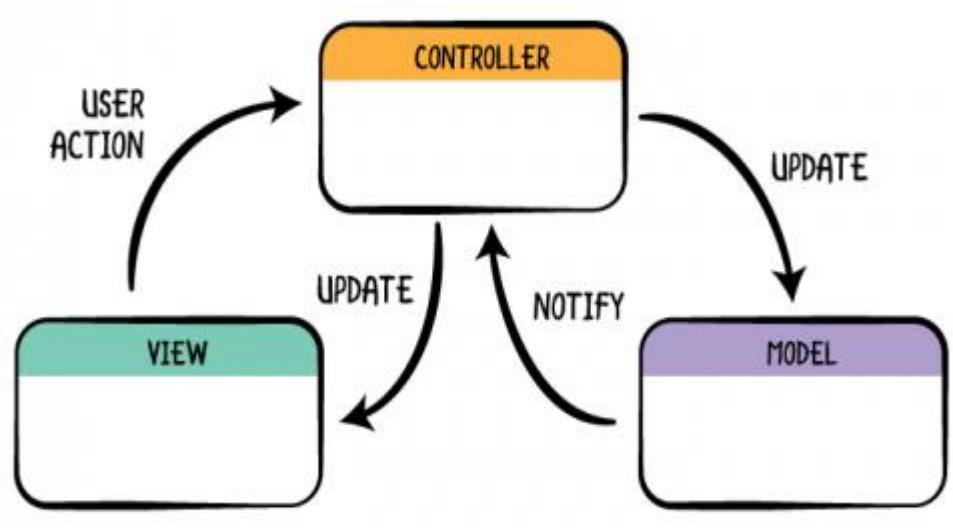


Figure 3.1- MVC Architecture

3.2.2 Technologies

A. Android version:

The development will first be done for Android devices. We use the « Android studio » IDE, with Google's Android SDK. We use the Java language, allowing to get the most out of Android and MP Android chart for animation and graphics.

Design patterns used in Android Application:

1. Singleton Pattern:

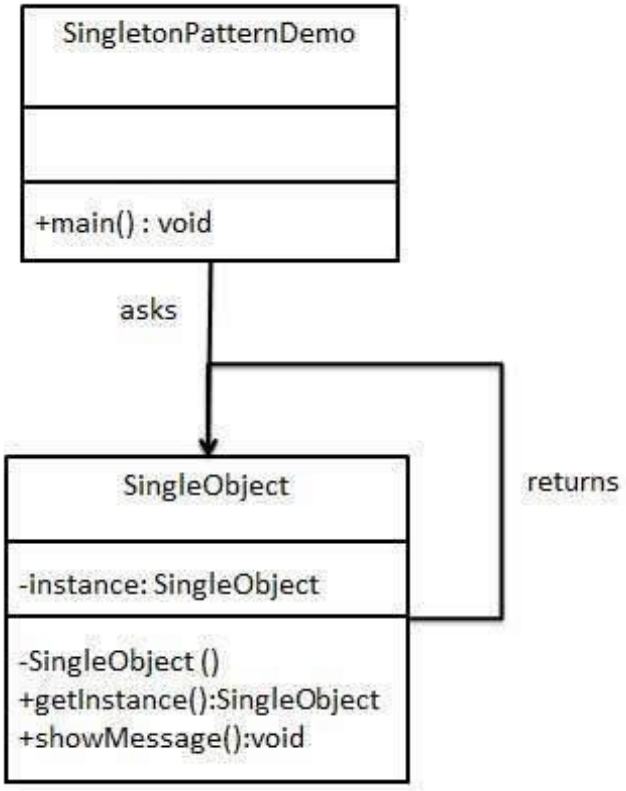


Figure 3.2 Singleton pattern

we use **Singleton pattern** to ensure that there is only one instance of a **class** is created in the Java Virtual Machine. It is **used** to provide global point of access to the object.
We use singleton pattern in two main parts in our code:

We use it in " House" class as for sure we serve one house a time:

```
public class House {

  private final List<PieEntry> pieEntries;

  private static final House ourInstance = new House();
  private final List<Entry> entries;
  private final LineDataSet dataSet;
  private float thisMonthReading;

  public static House getInstance() {
     return ourInstance;
  }
```

We also use it in "FirebaseUtils" the function which retrieve data from the cloud database to ensure that we retrieve readings just one time which helps us the improve

the performance:

```
private static final FirebaseUtils ourInstance = new FirebaseUtils();
private float totalLatestReading;
private float totalReading;
private FirebaseUtils() {
  final FirebaseDatabase database = FirebaseDatabase.getInstance();
  database.setPersistenceEnabled(true);
  myRef = database.getReference();
```

2. Observer Pattern:

The observer pattern is a software design pattern in which an object, called the subject, maintains a list of its dependents, called observers, and notifies them automatically of any state changes, usually by calling one of their methods, we use it to update our fragment views while keeping the principle of loose coupling between objects that interact with each other. It allows sending data to other objects effectively without any change in the Subject or Observer classes.

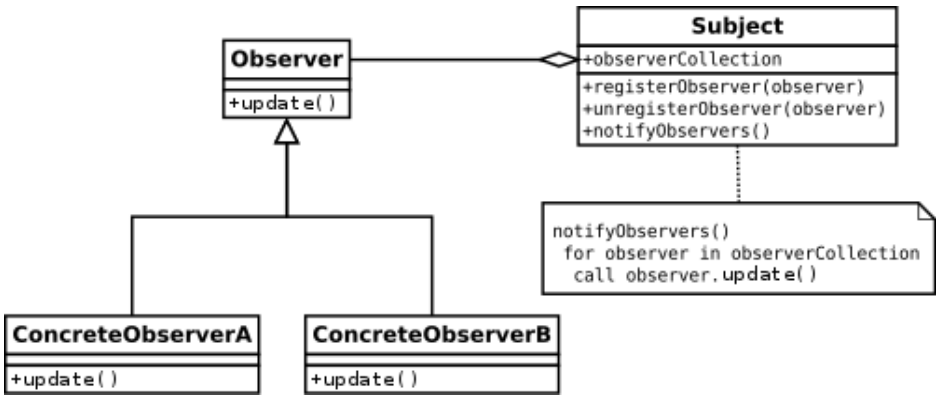


Figure 3.3- Observer pattern

In the next section, we will implement our observer:

```
@Override
public void update(Observable observable, Object o) {
  latestReadingTextView.setText(new
DecimalFormat("#.##").format(FirebaseUtils.getInstance().getTotalLates
tReading()));
  totalReadingTextView.setText(new
DecimalFormat("#.####").format(House.getInstance().getThisMonthRea
ding() / 1000) + " kWh");
  mAdapter.notifyDataSetChanged();
}
```

We use it for “ Rooms”:

```
//Registering observer
FirebaseUtils.getInstance().addObserver(this);
update(null, null);
```

We also use observer for “Readings” to update charts:

```
@Override
public void update(Observable o, Object arg) {


  if (House.getInstance().getPieEntries().size() > 0)
     pieChart.setData(pieData);

  pieData.notifyDataChanged();
  pieChart.notifyDataSetChanged();
```

B. IOS version:

We used the « xcode 9 » IDE to develop the application. We used Swift language for programming, we used core animation, core graphics and core data to integrate charts, we also use cocoapods for integration between IOS - Firebase - third parties.

Pods 〉 Podfile 〉 No Selection

```
# Uncomment the next line to define a global platform for your project
# platform :ios, '9.0'

target 'GraduationProject' do
  # Comment the next line if you're not using Swift and don't want to use dynamic frameworks
  use_frameworks!

  # Pods for GraduationProject
  pod 'IQKeyboardManagerSwift'
  pod 'Firebase/Core'
  pod 'Firebase/Database'
  pod 'Firebase/Auth'
  pod 'Firebase/Storage'
  pod 'Firebase/Messaging'

  pod 'SwiftCharts', '~> 0.6.1'
  pod 'PieCharts'

end
```

- MVC design pattern is used here in the iOS Application , and this is an example of that:

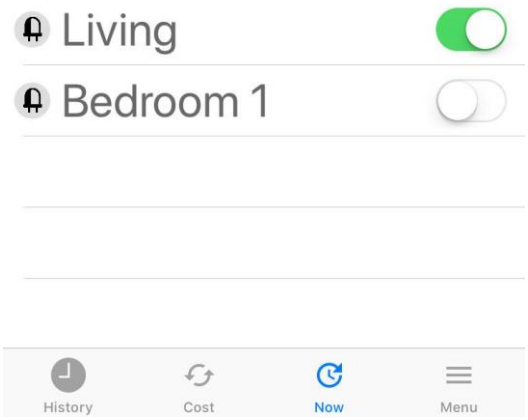
1. Here is our model including the Singleton variable:

```
//
//  RoomsModel.swift
//  GraduationProject
//
//  Created by hesham ghalaab on 6/27/18.
//  Copyright © 2018 hesham ghalaab. All rights reserved.
//

import Foundation

class RoomModel {
    static var rooms = [RoomModel]()

    var id: String
    var power: Bool
    var room_name: String
    var totalReadingsOfRoom: Double
    var totalCostOfRoom: Double

```

2. We get the data and then make our array of objects from the model.

```swift

func getRooms(){
    // networking Layer is here
    let ref = Database.database().reference()
    ref.child(Defaults.userId).child("rooms").observe(.value, with: { (snapShot) in
        if !snapShot.exists(){
            self.alertUser(withTitle: "Oubs!", message: "error occured, please try again later")
            return
        }

        let values = snapShot.value as! NSDictionary
        self.nowRooms = []
        for value in values{
            let id = value.key as? String ?? ""
            let _roomValue = value.value as! NSDictionary
            let power = _roomValue["power"] as? Bool ?? false
            let room_name = _roomValue["room_name"] as? String ?? ""
            print("room_name: \(room_name), power: \(power), roomId: \(id)")

            // creating our array of object
            let room = RoomModel(id: id, room_name: room_name, power: power, totalReadingsOfRoom: 0, totalCostOfRoom: 0)
            self.nowRooms.append(room)
        }

        self.tableView.reloadData() // updating the view
    }) { (error) in
        Helper.hideLoading(self.view)
        self.alertUser(withTitle: "Oubs!", message: error.localizedDescription)
    }
}

```

3. Now we are ready to update our view with data :



4. we made our iOS application according to Human interface guideline

iOS Design Themes

As an app designer, you have the opportunity to deliver an extraordinary product that rises to the top of the App Store charts. To do so, you'll need to meet high expectations for quality and functionality.

Three primary themes differentiate iOS from other platforms:

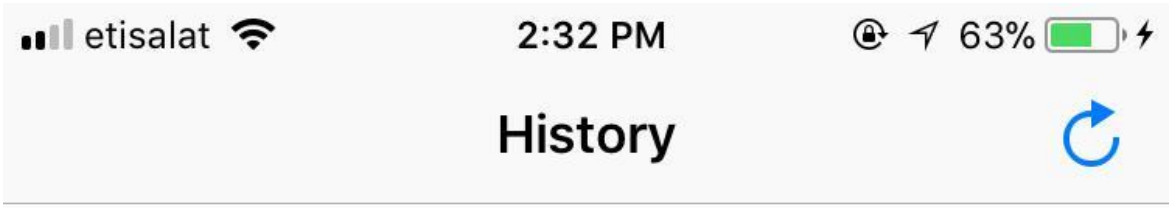
- **Clarity.** Throughout the system, text is legible at every size, icons are precise and lucid, adornments are subtle and appropriate, and a sharpened focus on functionality motivates the design. Negative space, color, fonts, graphics, and interface elements subtly highlight important content and convey interactivity.
- **Deference.** Fluid motion and a crisp, beautiful interface help people understand and interact with content while never competing with it. Content typically fills the entire screen, while translucency and blurring often hint at more. Minimal use of bezels, gradients, and drop shadows keep the interface light and airy, while ensuring that content is paramount.
- **Depth.** Distinct visual layers and realistic motion convey hierarchy, impart vitality, and facilitate understanding. Touch and discoverability heighten delight and enable access to functionality and additional content without losing context. Transitions provide a sense of depth as you navigate through content.

Color

Color is a great way to impart vitality, provide visual continuity, communicate status information, give feedback in response to user actions, and help people visualize data. Look to the system's color scheme for guidance when picking app tint colors that look great individually and in combination, on both light and dark backgrounds.

Navigation Bars

A navigation bar appears at the top of an app screen, below the status bar, and enables navigation through a series of hierarchical screens. When a new screen is displayed, a back button, often labeled with the title of the previous screen, appears on the left side of the bar. Sometimes, the right side of a navigation bar contains a control, like an Edit or a Done button, for managing the content within the active view. In a split view, a navigation bar may appear in a single pane of the split view. Navigation bars are translucent, may have a background tint, and can be configured to hide when the keyboard is onscreen, a gesture occurs, or a view resizes.



Tab Bars

A tab bar appears at the bottom of an app screen and provides the ability to quickly switch between different sections of an app. Tab bars are translucent, may have a background tint, maintain the same height in all screen orientations, and are hidden when a keyboard is displayed. A tab bar may contain any number of tabs, but the number of visible tabs varies based on the device size and orientation. If some tabs can't be displayed due to limited horizontal space, the final visible tab becomes a More tab, which reveals the additional tabs in a list on a separate screen.

3.2.3. Interaction:

A. System Sequence diagram:

If the user use the mobile application for the first time, he has to login using his serial number and password.

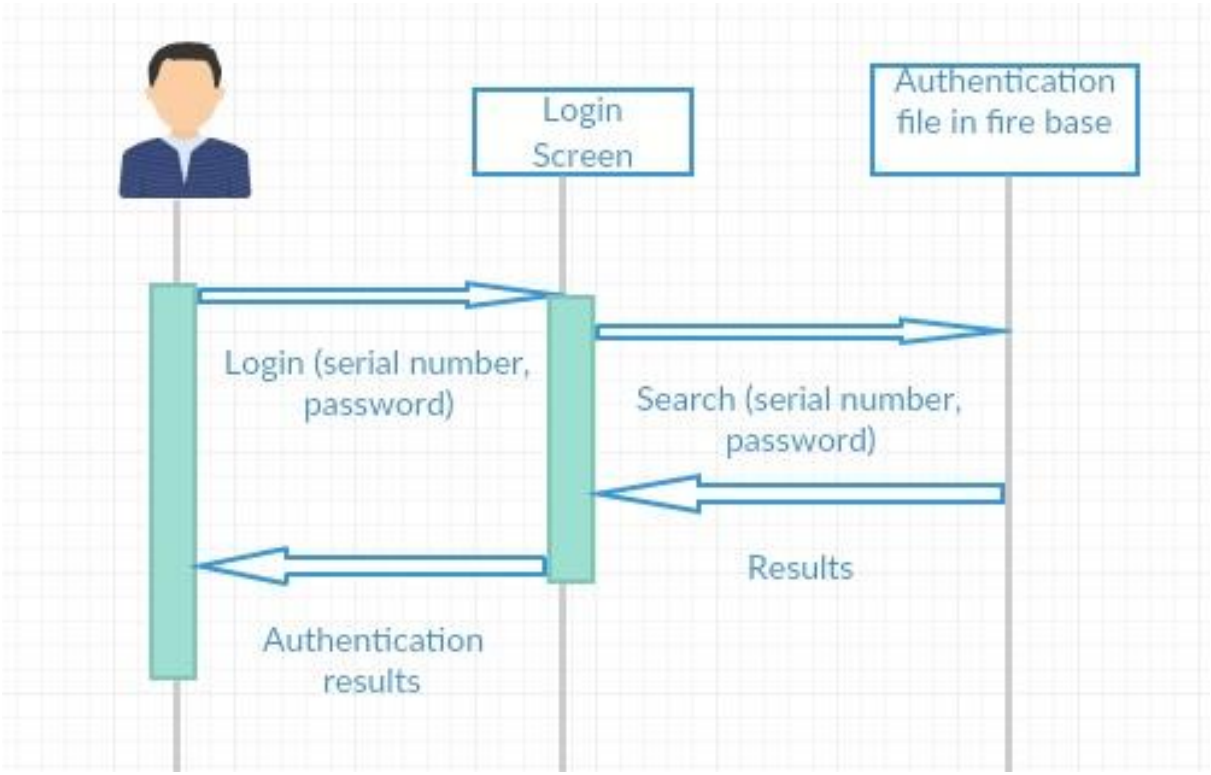


Figure 3.4- Login sequence diagram

Switchack mobile application has three main fragments:

1. **History Fragment**

This screen provides user with a histogram to show his usage during the month to help him to recognize when his usage is exceed his limits.

2. **Cost Fragment**

This Screen provides the user with pie chart with the expected cost of each fuse consumption

3. **Now Fragment**

It’s the main screen where user can know the current usage in his whole appartement, and can control some devices and fuses, can OFF/ON remotely any fuse.

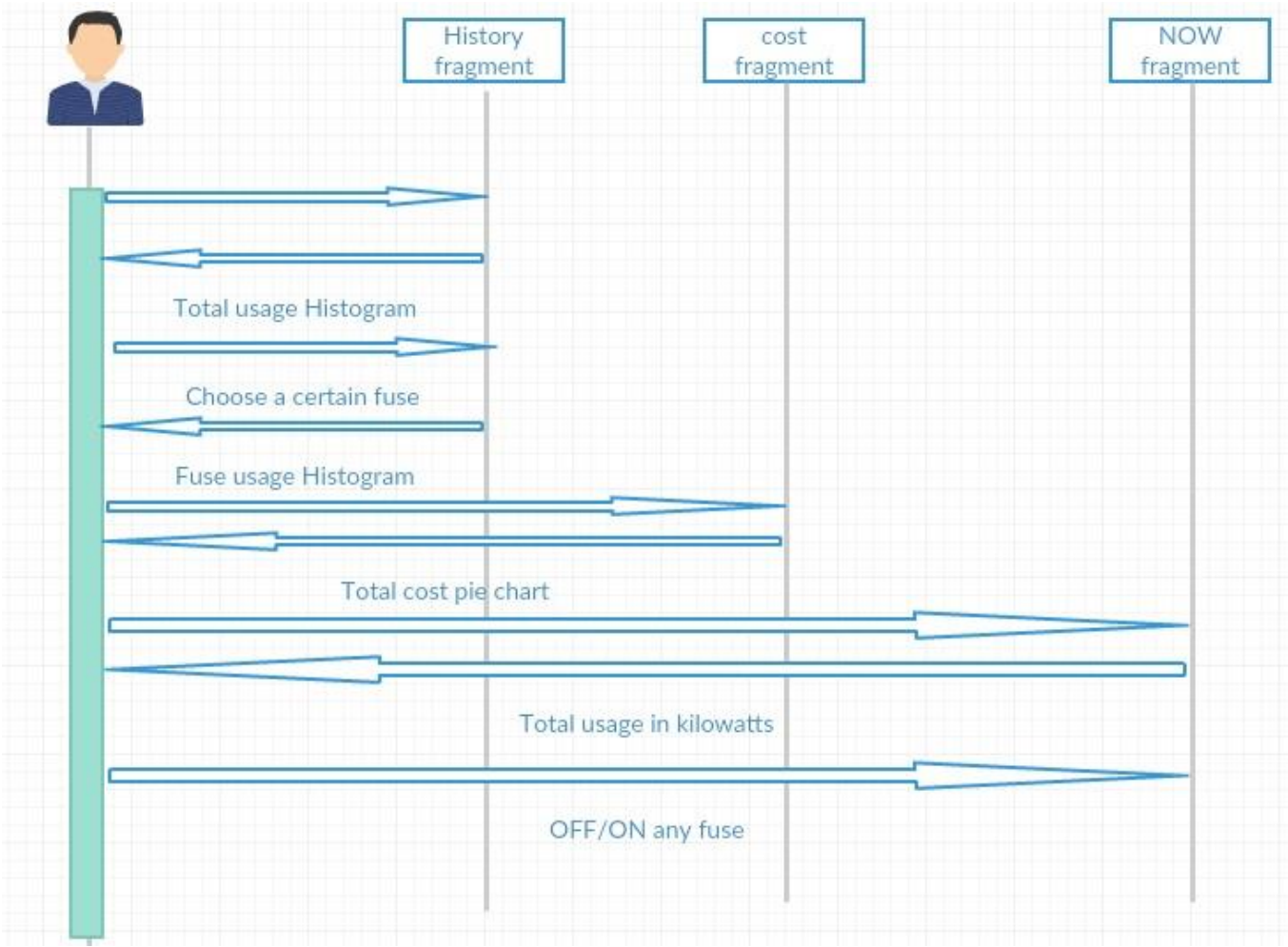


Figure 3.5 - Application screens sequence diagram

B. System Activity diagram:

Activity diagram is one of the most important diagrams in UML to describe the dynamic aspects of the system. Each activity describes an operation of the system. Firstly the user must login to access the control of the home using the application, using the serial number of the device and his password, the application check our database and if the user is a registered one, then he has four main screens he can access, this activity diagram helps you to understand the function of each screen.

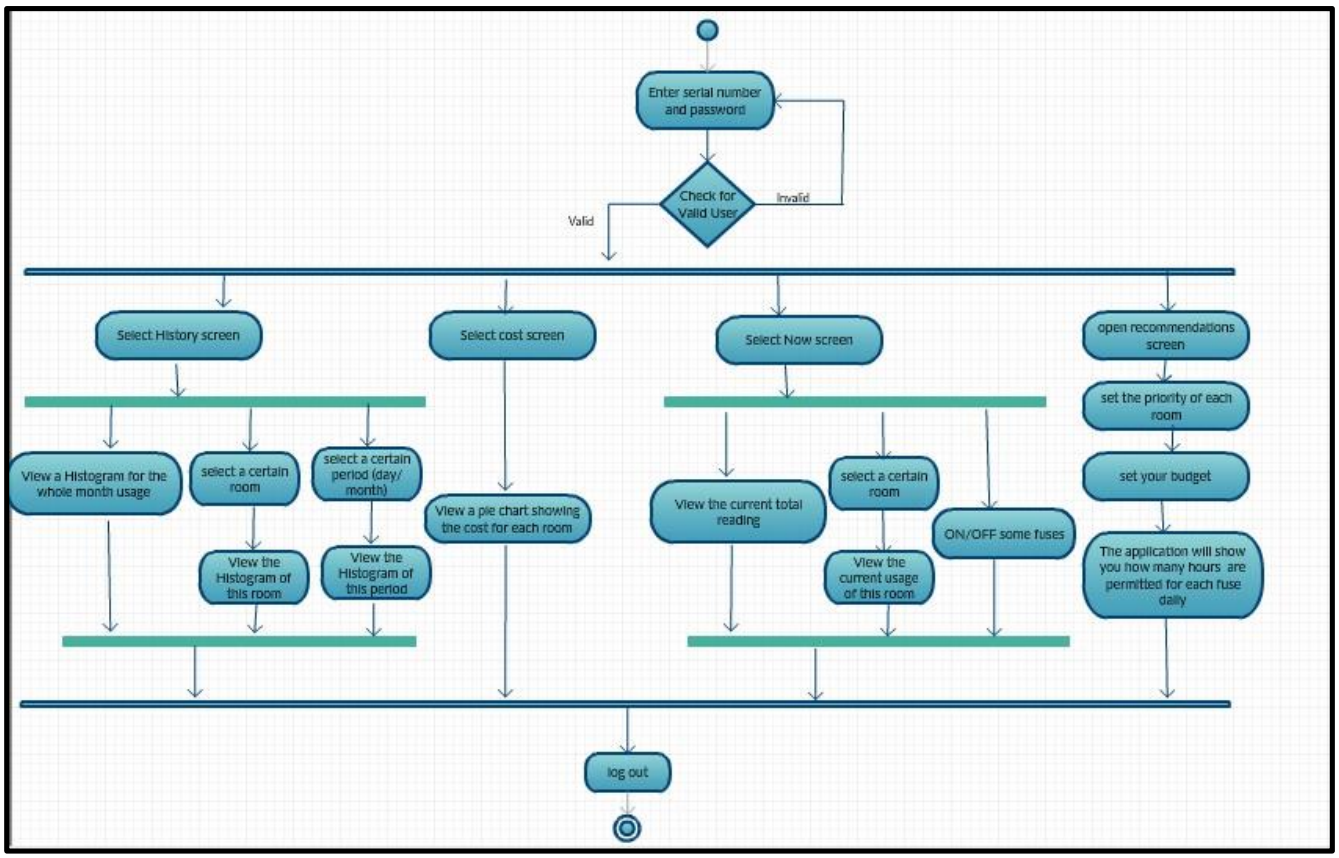


Figure 3.6 - System Activity diagram

C. Data Flow diagram:

This diagram shows the way information flows through a process or system.

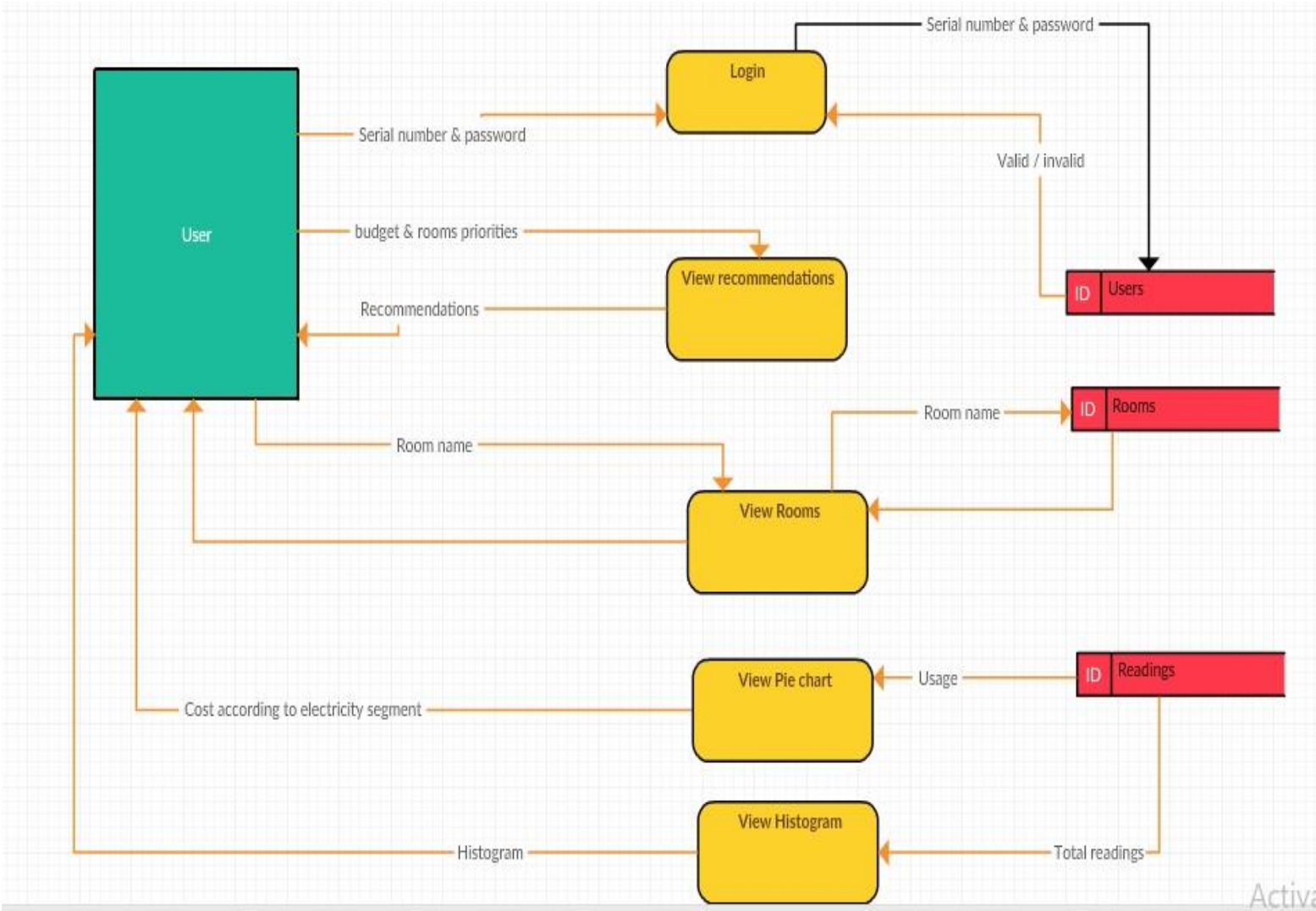


Figure 3.8- System DFD

3.3 Database:

Specifying the requirements of the application and analyzing them allowed us to get to the second step which is designing the database, we choose to use firebase as a communication layer between our hardware module and mobile application for many reasons mentioned before.

3.3.1. Database Structure:

The diagram below represents the entities that form the database of our application on the firebase.

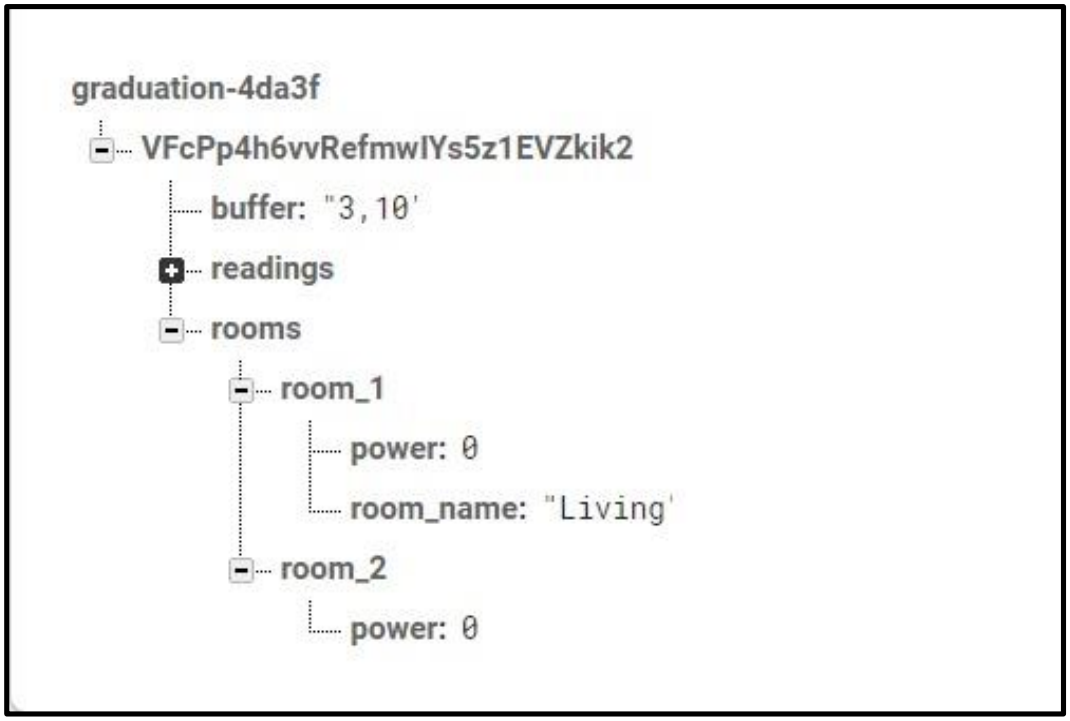


Figure 3.7- Database structure

It consists of:

1. **Rooms:** This entity refers to the rooms in the apartment using our system, each room has two parameter, Power which indicate this fuse is ON or OFF, and room name which is editable by the user.

   JSON code for rooms table:

```
"rooms" : {
  "room_1" : {
    "power" : 1,
    "room_name" : "Bedroom"
  },
  "room_2" : {
    "power" : 1
  }
```

2. **Readings:** This entity carries all the readings and act like real time database, each reading also related to certain room.

   JSON code for readings table:

```
{
"VFcPp4h6vvRefmwIYs5z1EVZkik2" : {
  "buffer" : "19,16",
  "readings" : {
```

```
        "1520267691791" : "2224.20,67683.00",
        "1520267691884" : "2224.20,67720.40",
        "1520267693056" : "2224.20,67680.80",
        "1520267693498" : "2224.20,60882.80",
        "1520267694556" : "2224.20,67689.60",
        "1520267696355" : "2224.20,67687.40",
        "1520267696740" : "2224.20,60889.40",
        "1520267697828" : "2224.20,67687.40",
        "1520267700056" : "2224.20,60889.40",
        "1520267701135" : "2224.20,67685.20",
        "1520267702197" : "2224.20,67687.40",
        "1520267703290" : "2224.20,60887.20",
        "1520267704387" : "2224.20,60889.40",
        "1520267705557" : "2224.20,67685.20",
        "1520267706656" : "2224.20,60889.40",
        "1520267707758" : "2224.20,60887.20",
        "1520267708855" : "2224.20,67685.20",
        "1520267712055" : "2224.20,60885.00",
        "1520267712366" : "2224.20,60891.60",
        "1520267713435" : "2224.20,67685.20",
        "1520267714481" : "2224.20,60889.40",
    }
```


3. **Users:** This entity contains the information of our users, their serial numbers and passwords.

Why do we use “Buffer”?

Since Arduino is unaware of time, so it just send the reading instantly to firebase, we deal with this problem by using buffer which receives the readings from arduino and adds timestamp to it and then passes it to readings entity.

Note: his functionality is implemented using firebase cloud functions. By continuously checking for edits on buffer and then creating a key value pair in readings whenever the buffer value changes

```
1
2 const functions = require('firebase-functions');
3
4 const admin = require('firebase-admin');
5 admin.initializeApp(functions.config().firebase);
6
7
8 exports.addTimeStamp = functions.database.ref('/{userID}/buffer').onUpdate((event) => {
9     const buffer = event.data.val();
10    return event.data.ref.parent.child('readings').child(Date.now()).set(buffer);
11 });
12
13
```

3.3.2. Database Functions:

Those are the **main functions** we use in firebase

1. Authentication: It would usually take months to set up and maintain your own authentication system, but Firebase Authentication lets us build an authentication service in a few lines of code that will handle everything, so users can sign in using their serial number and password.

2. Real-time database: this is the most important function we used in firebase, because we want to ensure that our clients is always updates with what is happening in their home electricity.

3. Offline capabilities: By using this function, we can keep a track of the connection state and update our app accordingly.

4. Cloud functions: We use this function to add timestamp to each reading sent by arduino before passing it to readings entity

3.4. Machine Learning:

We want to transform our solution from ordinary straight forward and just regular monitoring or controlling to a more intelligent solution.
This would be achieved in our problem by making our system a recommender one.

Thus, We aim to recommend a certain behavior by which users could lower their bill.

3.4.1 Importance

We found that first act could be taken to reduce bills is reducing our wasted consumption. So, we should recommend to a user an optimum period of time for every device (or a circuit breaker) that will nearly result in eliminating any wasted consumption.

3.4.2 Implementation

We predict optimum period of time in hours, this achieved through linear regression analysis. As we have a dataset containing a detailed consumption of all circuit breakers over time, we could use average consumption of a certain circuit breaker as an input feature.
In addition to this feature we added a predetermined priority from the user through his mobile application as a feature, and estimated overall budget which is also predetermined from the user.

1. **Hypothesis:**

With those features explained above we get the following hypothesis:

$$hi(x) = theta\_0 + theta\_1 * x1 + theta\_2 * x2 + theta\_3 * x3$$

Where:

hi(x): predicted output (number of hours recommended for circuit breaker i).
Theta_n : prediction weights.
i: index (1,2,....etc).
x1 : Priority of circuit breaker i.
x2 : determined budget of overall consumption.
x3 : average consumption of circuit breaker i.

Now we need to adjust theta_i values for accurate prediction. This would be accomplished through the following section.

2. **Cost Function:**

Here we aim to minimize the error, which is regularly the difference between predicted output but in linear regression problems we try to achieve higher accuracy, so we put the previous hypothesis in a function called cost function.
Our cost function will be as follows:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2 = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$$

Where:
J(theta) : the accuracy of our prediction.
M : total number of entries in the training set.
Yi : actual output at index i.

This could be achieved by the following Matlab code:

```
function J = ComputeCost(X, Y, theta)

m = length(Y);
J = 0;

% Compute the cost for a specific value of Theta
J =(1/(2*m))*(sum(((X*theta)- Y).^2));


end
```

It is called mean square error function.

3. Gradient Descent:

So we have our hypothesis function and we have a way of measuring how well it fits into the data.

Now we need to estimate the parameters in the hypothesis function. That's where gradient descent comes in.

The gradient descent algorithm is:

repeat until convergence: {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_0^{(i)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_1^{(i)}$$

$$\theta_2 := \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_2^{(i)}$$

...

}

We replicate the equation of theta as many as thetas we have. Important note we should consider is that, we must ensure simultaneous update of theta values to avoid interference between old theta values and new ones.

4. Feature Scaling & Mean Normalization

We can speed up gradient descent by having each of our input values in roughly the same range.This is because θ will descend quickly on small ranges and slowly on large ranges, and so will oscillate. inefficiently down to the optimum when the variables are very uneven. Two techniques to help with this are feature scaling and mean normalization.

$$x_i := \frac{x_i - \mu_i}{s_i}$$

Where
x: entries of the dataset.
Mu: the mean of dataset.
S (sigma) : the standard deviation of dataset.

```
function [X, mu, sigma] = FeatureNormalize(X)

mu_1 = mean(X(:,1));
mu_2 = mean(X(:,2));
mu = [mu_1, mu_2];

sigma_1 = std(X(:,1));
sigma_2 = std(X(:,2));
sigma = [sigma_1, sigma_2];

x1 = (X(:,1) - mu_1)./sigma_1;
x2 = (X(:,2) - mu_2)./sigma_2;

X = [x1 x2];
```

Mean and std are built in functions to calculate mean and standard deviation.

5. Prediction

Our main function that produce theta values (which we will use in mobile apps to predict the optimum time for a circuit breaker) will be as follows:

```
Data = load('Dataset.txt');
X = Data(:,1:3);
Y = Data(:,4);
m = length(Y);

% Print out some data points
fprintf('First 10 examples from the dataset: \n');
fprintf(' x = [%.0f %.0f], y = %.0f \n', [X(1:10,:) y(1:10,:)]');

% Normalizing features and set them to zero mean
fprintf('FeatureNormalize...\n');
[X, mu, sigma] = FeatureNormalize(X);

% Add X0 column to dataset
X = [ones(m,1) X];

% ================================================================

% Choosing theta and minimizing the cost function

% Learning rate and number of iterations initialization
alpha = 0.01;
iterations = 400;

% Theta initialization
theta = zeros(size(X,3),1);

% Run GradientDescent
[theta, J_history] = GradientDescent(X,Y,theta,alpha,iterations);
```

We chose a learning rate (alpha) = 0.01 and will run gradient descent for 400 iteration. Then initialize theta vector to zeros.

Theta vector will be like [theta_0, theta_1, theta_2, theta_3].

Now we could plot the convergence of cost function J(theta_i) and values of optimum theta.

```
% plotting the convergence
figure;
plot(1:numel(J_history), J_history, '-b', 'LineWidth', 2);
xlabel('number of iterations');
ylabel('Cost J');

%Display gradient descent's result
fprintf('Theta computer from GradientDescent = \n');
fprintf(theta);
fprintf('\n');
```

We could take these theta values and use them in the application to make our desired predictions.

3.4.3 Implementation in android app:

We enabled the user to get the optimal practice for his usage and his consumption in order to save money by entering his budget want to spend on electricity consumption.

We ask the user for some inputs like

-Budget

-priorities

To provide him with the recommended plan to spend money on a certain number of hours per month.

Recommendation Machine Learning Equation:

Total Recommended number of hours per each selected fuse

= X1*theta1 + X2*theta2 + X3*theta3

Where

x1: Priority number

X2: fuse old consumption

X3: entered budget

some snippets of android code:

```
public void addButton() {
 LinearLayout buttonLayout = findViewById(R.id.button_layout);
 Button recbutton = new Button(this);
 buttonLayout.addView(recbutton);
 recbutton.setText("Recommendation");
 recbutton.setOnClickListener(new View.OnClickListener() {
 public void onClick(View v) {
       float theta1 = 1;
       float theta2 = 1;
       float theta3= 1;
```

```
LinearLayout recommendedText = findViewById(R.id.layout_recommended_text);
 recommendedText.removeAllViews();
 for (int i = 0; i < priorityList.getChildCount(); i++) {
   Spinner spinner = (Spinner) priorityList.getChildAt(i);
    int roomIndex = spinner.getSelectedItemPosition() - 1;

    if (spinner.getSelectedItemPosition() > 0) {
  float selectedRoomReading = rooms.get(i).getTotalReading()/rooms.get(i).getReadings().size();

  float totalHours = priorityList.getChildCount() - i * theta1 + selectedRoomReading * theta2 +
  seek_bar.getProgress() * theta3;

TextView rtext = new TextView(recommendedText.getContext());
rtext.setText("The Recommended Optimal Use for your " +spinner.getSelectedItem() + " is "
+
  totalHours + "hrs/month");

     recommendedText.addView(rtext);
   }
```

RESULTS

- **Small & Portable Hardware Module**
- **Smart Android & IOS Mobile Applications**

4.1. Hardware

A Smart Device that could be installed easily at any home with the electrical panel connected with the internet to send and receive users' usage with accuracy above 95% .

4.2. Software

4.2.1. Screenshots:

A. Login screen:

This screen only allow users who are already have their own serial number and password to sign in to their system to start use the full functions of SWITCHACK.

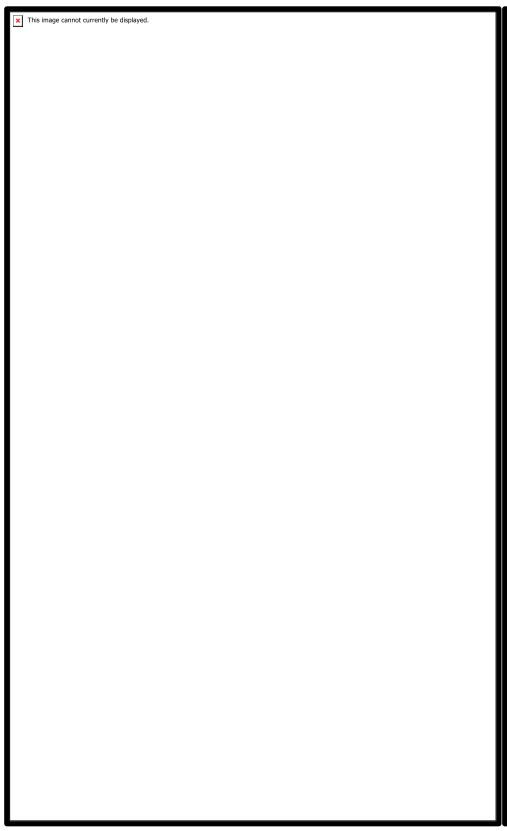


Figure 4.1 login screen

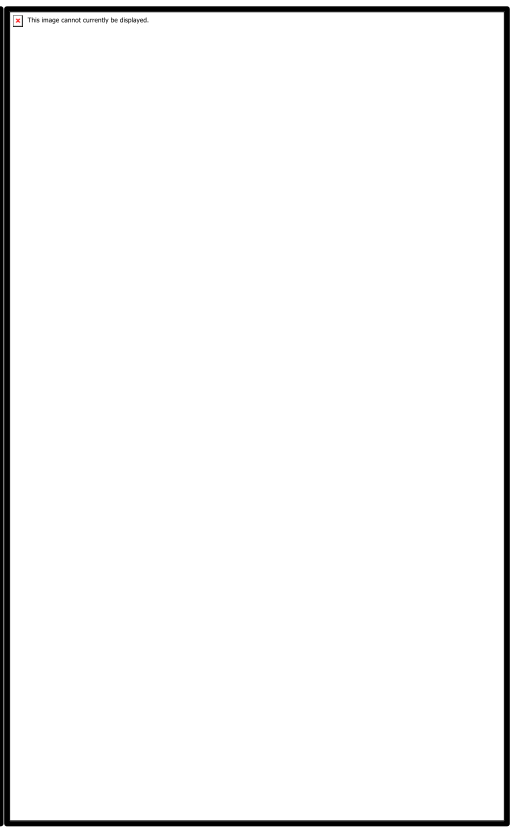


Figure 4.2 password is too short

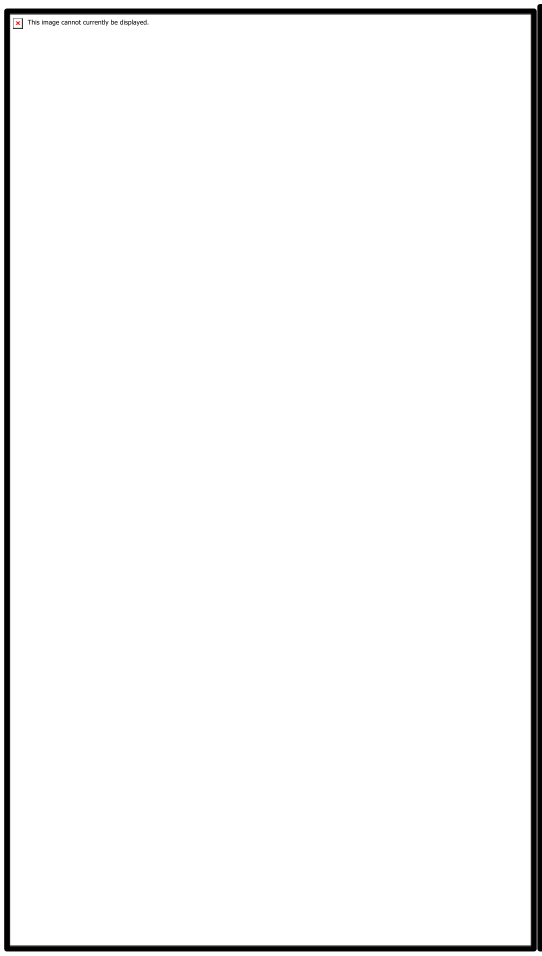


Figure 4.3 unregister user try to sign-in entering

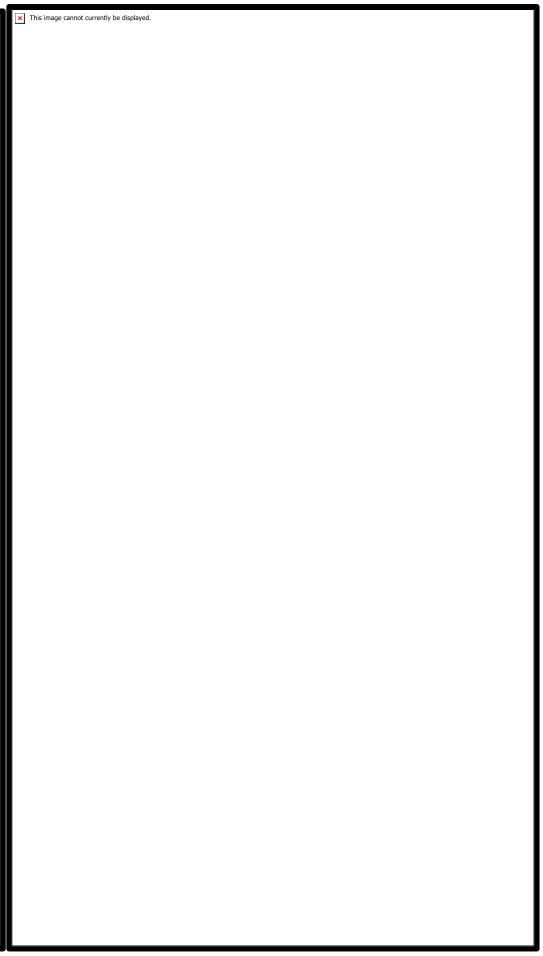


Figure 4.4 user try to sign-in without His data

B. History screen:

This screen allow users to check their usage history during the desired period of time and to check also every rooms' usage when they click on any room/fuse.

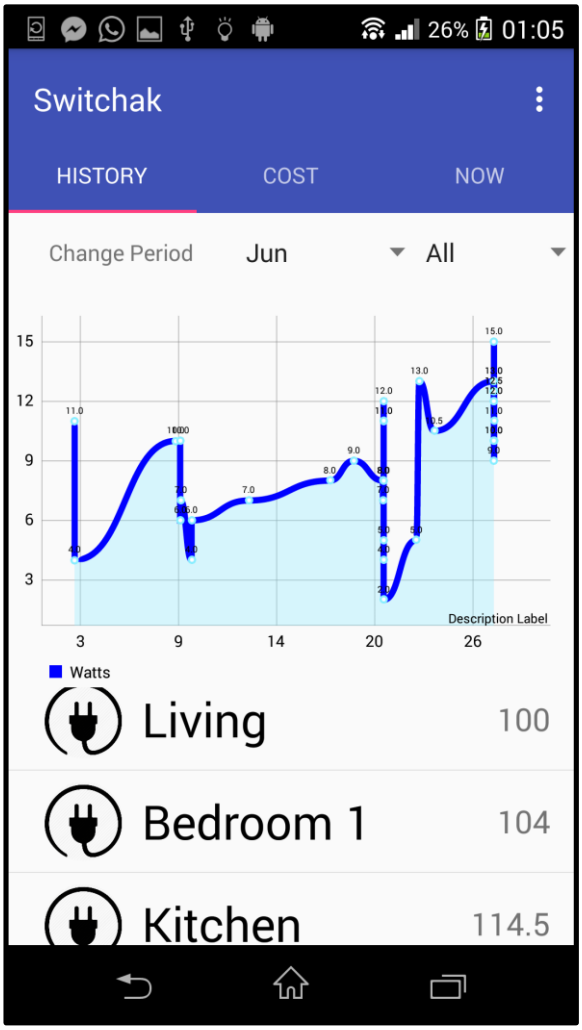


Figure 4.5 History Screen in Android

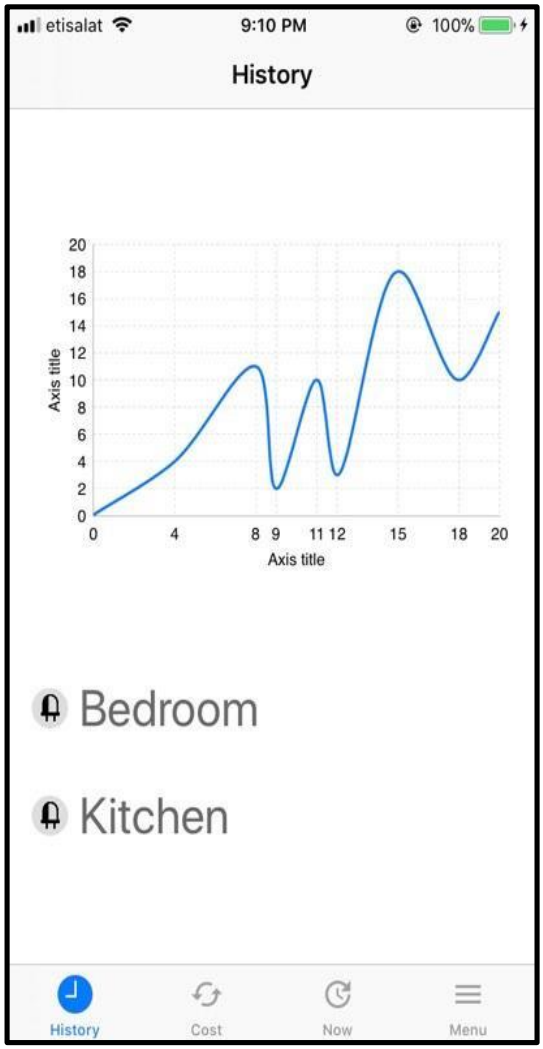
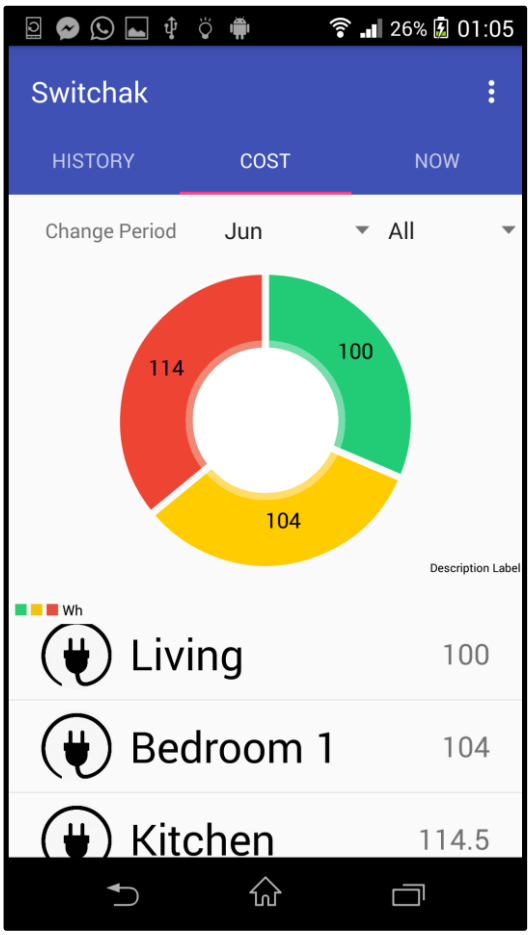


Figure 4.6 History Screen in IOS

C. Cost Screen

This screen allow users to check their usage cost percentage over the whole rooms/fuses entered to check also every rooms' total cost percentage when they click on any room/fuse.



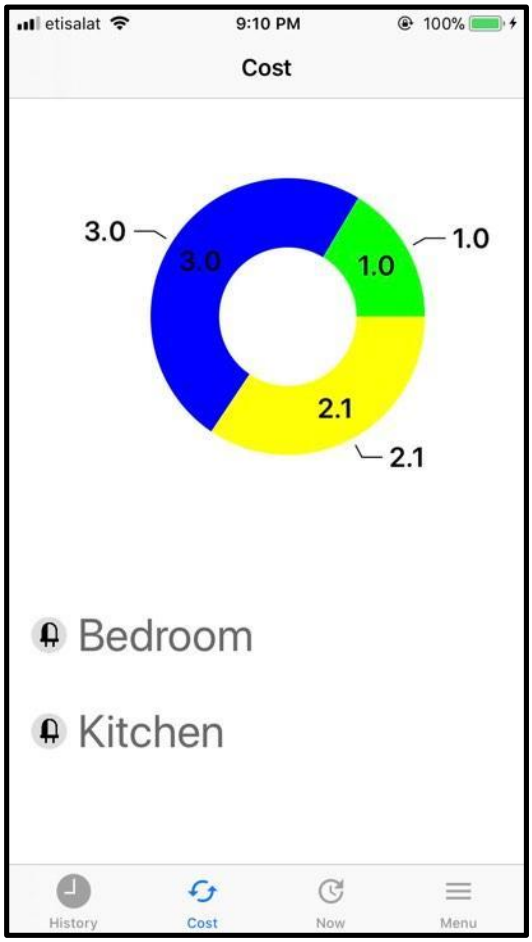


Figure 4.7 Cost screen in Android application
Figure 4.8 Cost screen in IOS application

D. Now Screen

This screen allows every user to see how much kilowatts is he using now during all his system.

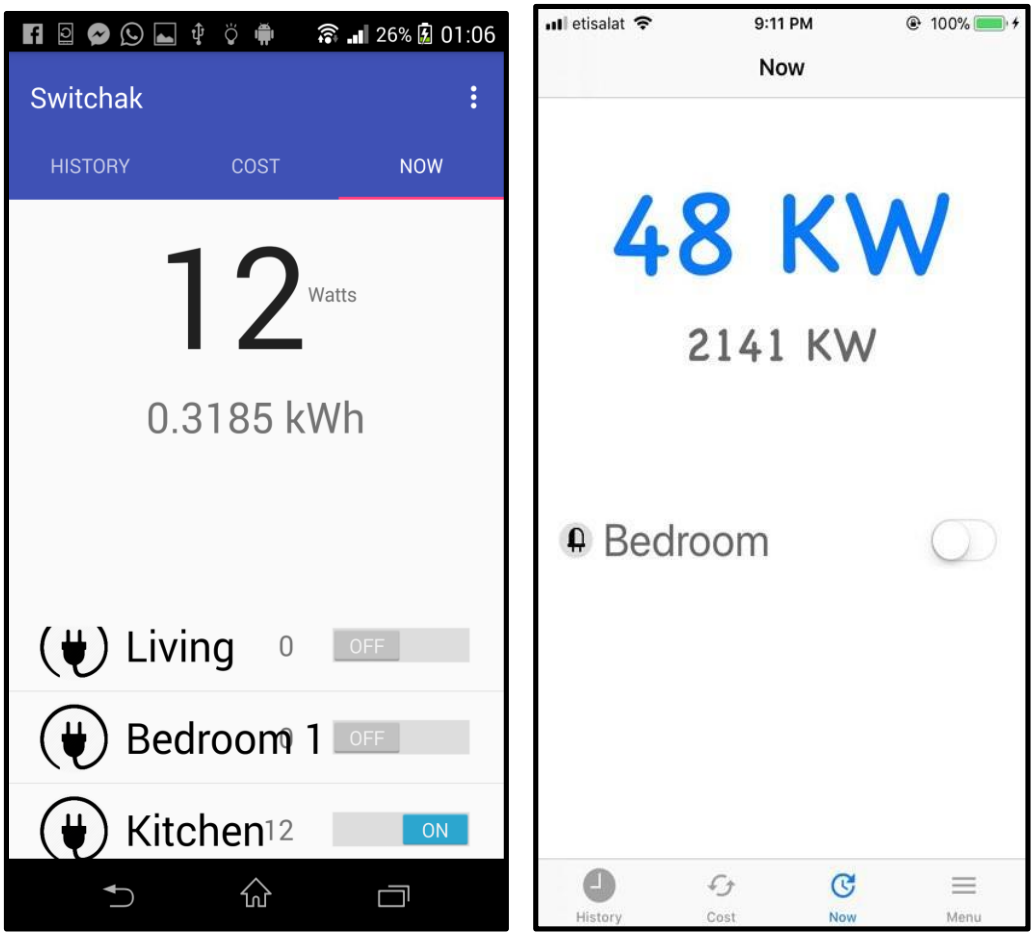
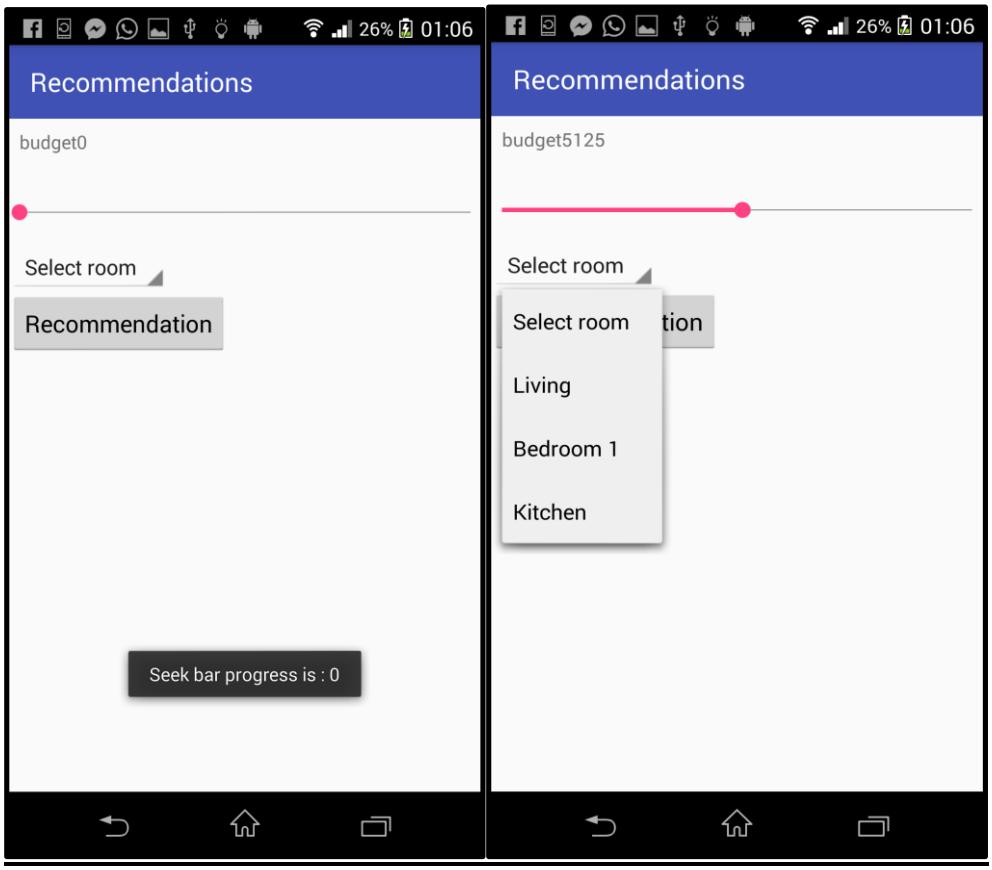


Figure 4.9 Now screen in android application
Figure 4.10 Now screen in IOS application

E. Recommendations Screen

The user start to select the budget amount and his devices' (fuses) priorities in order to get the optimal usage within the budget amount he entered when he click on the recommendation button.



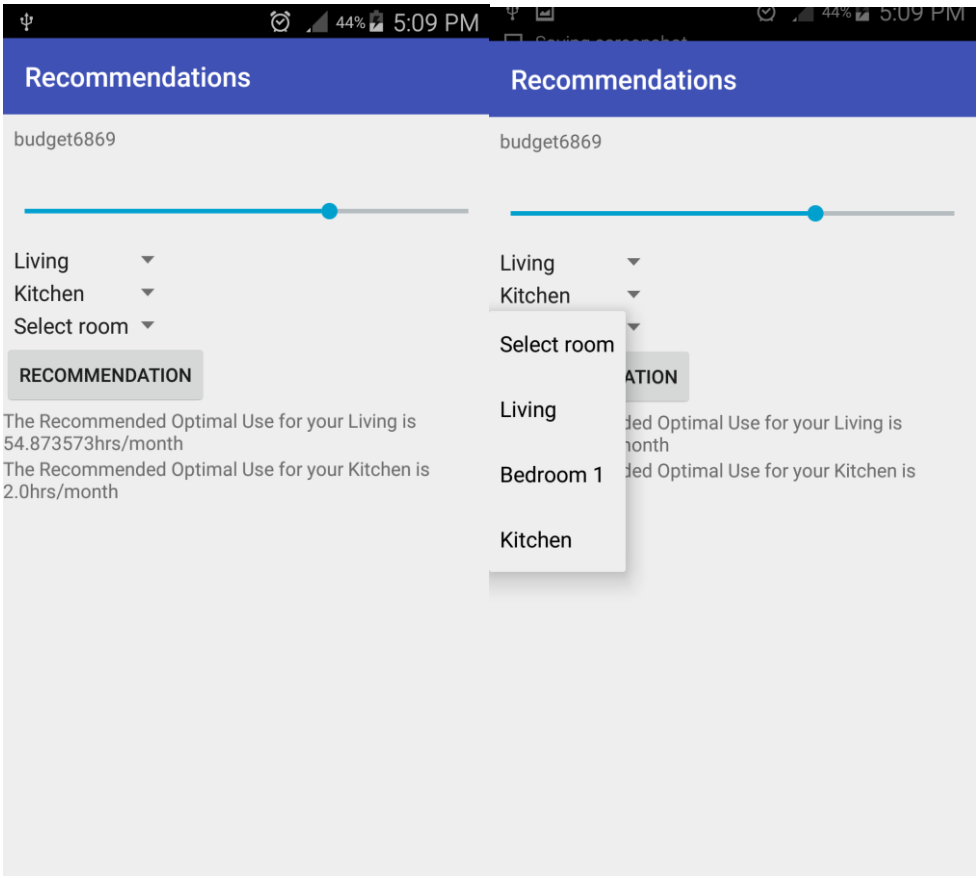


Figure 4.11 Recommendations screen in Android application

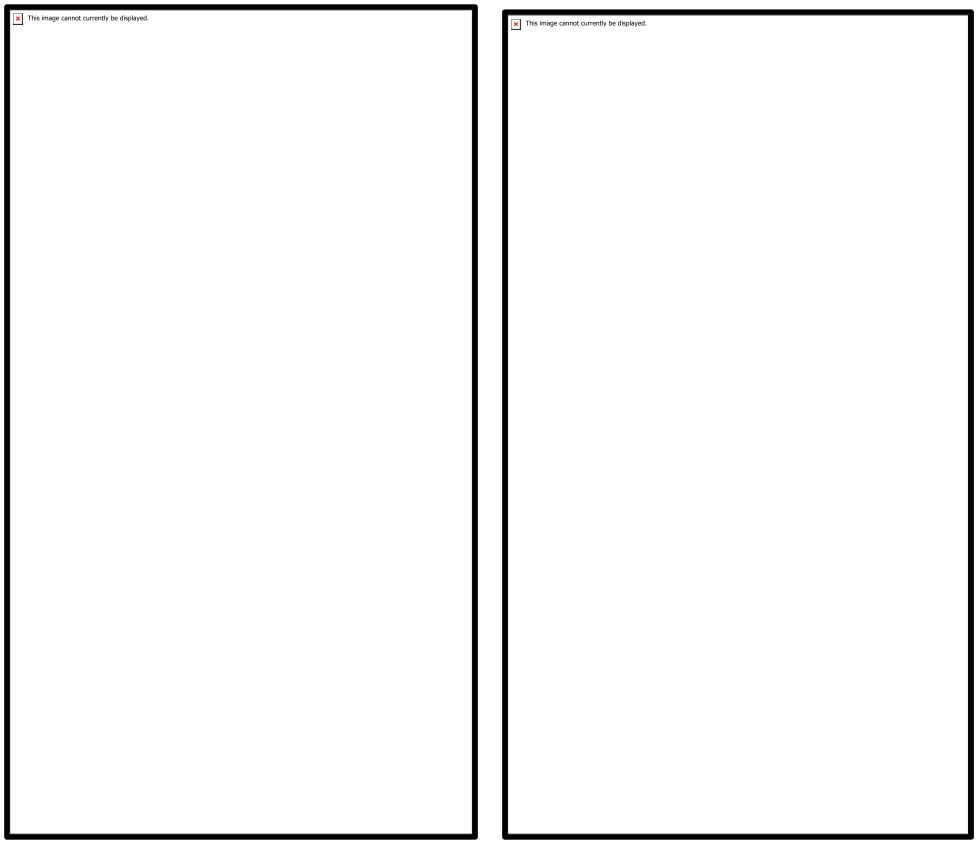


Figure 4.12 Recommendations screen in iOS application

REFERENCES

1. Android Developers. (n.d.). Retrieved June 20, 2018, from https://developer.android.com/
2. Android Basics: User input, from https://udacity.com/
3. Developing Android Apps, from https://udacity.com/
4. "Use of energy in Egypt, past and recent" report, by Eng. Tawfik F.Azer
5. https://www.worlddata.info/africa/egypt/energy-consumption.php
6. https://unstats.un.org/oslogroup/meetings/og-09/docs/day4-session9-Energy_Use_in_Egypt.pdf
7. https://energysmartshow.com/understanding-your-energy-use-at-home/
8. http://feelyourtempo.com/electricity-wastage-problem-consequence-solution/
9. " Java Design Patterns - reusable solution to common problems" by Rohit Joshi
10.
11. https://enterprise.press/stories/2017/07/09/government-announces-new-electricity-prices/
12. https://blog.resellerclub.com/why-is-firebase-the-best-mobile-backend-as-a-service/
13. https://www.proteloinc.com/blog/design-patterns-important-software-engineering/

Hardware References:

1. https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT
2. https://www.ibm.com/blogs/internet-of-things/what-is-the-iot/
3. https://www.hobbyist.co.nz/?q=interfacing-relay-modules-to-arduino
4. https://components101.com/sensors/acs712-current-sensor-module
5. https://learn.sparkfun.com/tutorials/what-is-an-arduino
6. https://nodemcu.readthedocs.io/en/master/
7. University of California: Phase, Period, and Vertical Shifts of Trigonometric Functions
8. University of Nebraska: Measuring Relative Phase Between Two Waveforms Using an Oscilloscope

Machine Learning References:

1. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics) 2nd Edition, Chapter 3

2. Data science, Machine Learning By Andrew Ng from https://www.coursera.org
3. Data Science, Machine Learning: Regression By Emily Fox and Carlos Guestrin ,from https://www.coursera.org/learn/ml-regression
4. A First Course in Machine Learning, Chapter 1.
5. .https://towardsdatascience.com/machine-learning-basics-part-1-concept-of-regression-31982e8d8ced
6. http://sebastianraschka.com/Articles/2014_about_feature_scaling.html
7. https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/