

# **Electricity Monitoring And Management**

A Graduation Project Report Submitted to  
the Faculty of Engineering at Cairo University  
In Partial Fulfillment of the Requirements for the  
Degree of Bachelor of Science in  
Electronics and Communications Engineering

Faculty of Engineering, Cairo University

## ***Prepared By***

**Mohamed Ahmed Mohamed Atya**

**Mahmoud Samy**

**Hamed Abd El Aziz**

**Mahmoud Tesar**

**Amir Ahmed Moselhy**

## ***Supervised By***

***Dr. Hassan Mostafa***

***Assistant Professor At The Faculty of Engineering, Cairo University.***

# Table Of Contents:

Acknowledgment .....	VII
Abstract .....	VIII
Chapter One: Introduction.....	1
Overview .....	2
Electricity Management .....	2
Device .....	2
Registration.....	3
Platforms.....	3
Motivation .....	4
Problem .....	4
Solution .....	5
Statistics .....	5
Business Opportunities.....	5
Evolution.....	6
Chapter Two : Used Technologies Background .....	7
1- HTML5.....	8
Definition .....	8
Advantages.....	8
1- Legacy/Cross Browser Support .....	8
2- Smarter Storage .....	8
3- Cleaner Code .....	8
4- Better Interactions .....	9
Examples .....	10
1- An image height and width using attributes and CSS.....	10
2- An image as a link.....	11
3- Uploading an audio.....	11
4- Uploading a video .....	11
5- Input type password .....	12
6- Input type color – with color picker .....	12
7- Local Storage (Store a counter permanently) .....	13

8-	Tables with different style using class I.....	14
2-	CSS3 .....	15
	Definition .....	15
	Examples .....	16
1-	Element and ID selector .....	16
2-	Define font properties.....	17
3-	Set the background color of different elements.....	18
4-	Set an image as the list-item marker .....	19
5-	Set the width, style and color of the border .....	20
6-	Pseudo - classes (match the first I element in all p elements).....	21
7-	Pseudo – classes (match all I elements in all first child p elements).....	21
3-	JavaScript .....	22
	Definition .....	22
	JavaScript Uses .....	22
1-	Java script can validate input .....	22
2-	JavaScript Can Change HTML Content:.....	23
3-	JavaScript Can Change HTML Attributes:.....	23
4-	JavaScript Can Change HTML Styles (CSS):.....	24
4-	jQuery .....	26
	Definition .....	26
	What Can jQuery Do? .....	26
	jQuery UI .....	27
	jQuery Mobile .....	28
5-	Json .....	29
	Definition .....	29
	Why JSON?.....	29
	Using XML .....	29
	Using JSON .....	29
	JSON Example: .....	30
	Output.....	30
6-	Phone Gap .....	31
	Introduction to PhoneGap .....	31

PhoneGap and Apache Cordova .....	32
How PhoneGap Works .....	33
7- Android .....	36
Definition .....	36
What can an Android phone do? .....	36
8- IOS.....	37
Definition .....	37
9- Windows Phone .....	37
Definition .....	37
Chapter Three : Implementation.....	39
Three main layers of every System Architecture.....	40
System Development Layer .....	41
Bootstrap.....	41
Introduction to Bootstrap.....	41
What is Bootstrap and why it is used?.....	41
Responsive Web Design.....	43
What is Responsive Web Design? .....	43
How Does Responsive Web Design Work?.....	43
Related Issues: .....	44
Mobile Application .....	45
Chapter Four : Work Outputs And User Experience.....	46
1- User Sign Up Flow Chart .....	47
2- User Login Flow Chart.....	48
3- Application Diagram .....	49
User Experience:.....	50
1- Sign Up Screen. ....	50
2- Log In Screen.....	51
3- Dash Board Screen.....	51
4- My Devices Screen. ....	53
5- Tarrifs Screen.....	54
6- Profile Screen.....	55
Chapter Five : Conclusion .....	56

Chapter 6: Related Works And Previous Attempts.....	58
Alternate Hybrid Application Frameworks.....	59
Worklight .....	59
Appcelerator Titanium.....	59
Qt Creator.....	60
Xamarin .....	60
Previous Attempts And Similar Smart Meter Solutions .....	61
Wattson Solar Plus .....	61
Onzo Energy Kit.....	61
Power Cost Monitor .....	62
Wattvision Power Monitor .....	62
Vattenfall Smart Meter App .....	63
Hohm and PowerMeter.....	63
Appendix .....	65
Code:.....	66
Globals.....	66
Sign Up.....	66
Log in .....	69
Charts .....	72
Devices .....	80
Tarrifs.....	84
REFERENCES.....	89
1- Motivation.....	89
2- Used Technologies Background .....	89
3- Responsive web design .....	89
4- Bootstrap .....	89
5- Related works and previous attempts .....	90

# List Of Figures

Figure 1 : Device Components.....	2
Figure 2 : Problem in Egypt.....	4
Figure 3: Real and changed image size.....	10
Figure 4 : jQuery And Its Tagline .....	26
Figure 5 : jQuery Mobile .....	28
Figure 6 : Adobe PhoneGap .....	31
Figure 7 : PhoneGap Build Process .....	31
Figure 8 : Apache Cordova.....	32
Figure 9 : Apache Cordova Android Emulator.....	32
Figure 10 : PhoneGap application architecture .....	33
Figure 11 : PhoneGap as a framework filling the gap.....	33
Figure 12 : Apache Cordova Native Application Architecture .....	34
Figure 13 : PhoneGap Build Logo.....	35
Figure 14 : Android Logo.....	36
Figure 15 : iOS Logo.....	37
Figure 16 : Windows Phone Logo .....	37
Figure 17 : System Architecture Diagram .....	40
Figure 18 : Bootstrap Framework.....	42
Figure 19 : Figure 3.3 Responsive Layout .....	44
Figure 20 : IBM WorkLight.....	59
Figure 21 : Appcelerator Titanium.....	59
Figure 22 : Xamarin development.....	60
Figure 23 : Wattson Solar Plus.....	61
Figure 24 : Onzo Energy Kit .....	61
Figure 25 : Power Cost Monitor.....	62
Figure 26 :Wattvision Power Monitor .....	62
Figure 27 : Vattenfall Smart Meter App.....	63
Figure 28 : Google PowerMeter .....	63

# *Acknowledgment*

All our gratitude and deep thank to our parents and all the family members for their courage, supports and motivations they had given to us to finish this project.

All our respect and appreciation to our supervisor Dr. Hassan Mostafa for his situations, trust, useful comments, recommendations and guidance which helped us a lot to show this thesis.

We would like to thank Valuemotiv company and Engineer Ahmed Gamiel for their help and support.

We would also like to thank our communication department colleagues who had offered us advice and helped us at the beginning of our project.

# *Abstract*

In the last few years here in Egypt there is a quickly rising problem which is Electricity shortage and its rising prices.

Our project is all about this, it is sponsored by a company that shows great interest in solving the electricity problem in Egypt.

This company has a device that can calculate the electricity consumed by the place and it will be released soon.

It consists of two main parts:

- 1- Sensor that is attached to the new digital meters to help calculating usage data.
- 2- Gateway that receives this data from the sensor and sends it to our online server continuously every 5 minutes.

We are supposed to do two main things as our project.

1<sup>st</sup>- A web application where he can register his device and monitor his data daily, weekly, monthly and yearly so he nearly knows his consumption, corresponding cost then try to reduce this usage to reduce the cost.

2<sup>nd</sup>- A mobile application with the same features.

Finally the client can access his device through both the web application and the mobile application.



## ***Chapter One: Introduction***

**The introduction is classified into four sections:**

**1- The main idea of our project (OVERVIEW).**

**2- Motivation.**

**3- Business Opportunities.**

**4- Evolution.**

**In this chapter we will introduce our project, tell the motivations to proceed in it, state some of the business opportunities using this project and state its expected evolution.**

## Overview

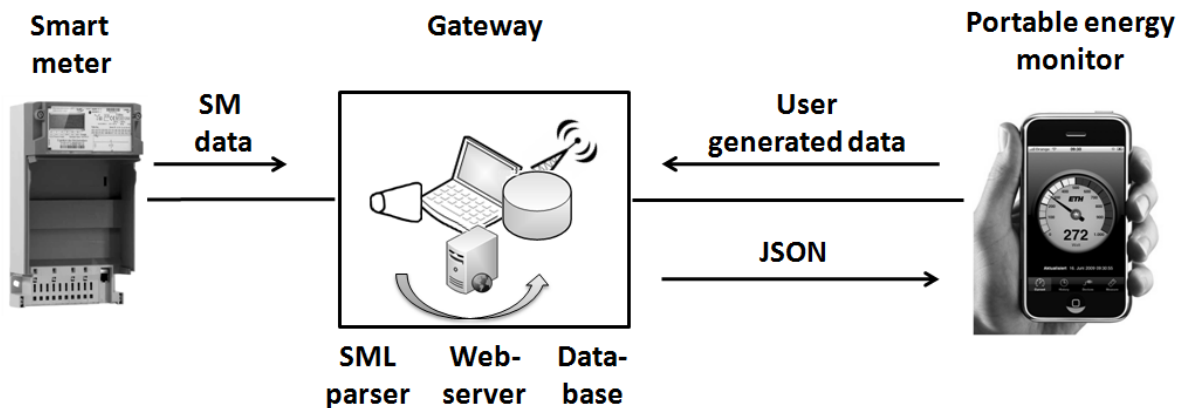
### Electricity Management

Let's start talking about the first part which is the project idea .... Our project idea is Electricity Management and Monitoring; How to control and rationalize your electric usage at home, office and building.

Now a question arises!! ,,,, How can we do that!!

All of us have the electricity meter in or out of our apartments, we will use this meter to analyze and monitor our daily, weekly and monthly usage.

### Device



**Figure 1:Device Components**

Weiss, M., Mattern, F., Graml, T., Staake, T., Fleisch, E.: Handy feedback: connecting smart meters with mobile phones. In: Proc. MUM 2009, pp. 1–4. ACM, New York (2009)

We have a device that consists of two main parts: a sensor and a gateway, this sensor reads the amount of electricity used in the place using the digital electricity meter of this place.

The device transmits the consumption levels within short intervals; so anyone can judge through the detailed data graphs whether the user is at home or not.

We will place a sensor externally on this digital meter which helps in measuring usage and calculating the corresponding cost, This sensor will be connected to the home internet through a gateway; This gateway receives the readings of the sensor every five minutes and transmits the readings and values of our usage to our online server over specified IP, When

the server receives these measurements we get a graphical representation to our consumption in unit of minutes.

When the user knows his usage cost; It's assumed to be easy for him to control and keep his average consumption within the mean.

Later we will know the detailed process and steps for this simple description.

## Registration

Now we will discuss somehow the commercial side of our product.

Each device has a serial number and an API; each user registers with his ID and password on the website and then registers the serial number and API of all the devices he owns in case of having more than one device, So he will be able to see and get his consumption graphs or history.

## Platforms

The user can also do the same on his mobile whatever his platform is:

- 1- ANDROID.
- 2- IOS.
- 3- WINDOWS PHONE.

Energy savings and high security are important advantages that motivate people to work on this issue and develop various ways to solve and avoid the power shortage problem.

## Motivation

In the last few years here in Egypt there is a quickly rising problem which is electricity shortage and its rising prices.

## Problem



**Figure 2: Problem in Egypt**

<http://www.vetogate.com/1205902>

The problem of electricity is one of the most important problems facing Egypt which led to the existence of economic and social crises. The repeated drop of electricity is a Phenomenon experienced by the Egyptian society.

Years ago, experts warn a sudden increase of consumption and the pressure that will fall on networks to generate electricity because of the increased temperatures and high work rates consumption and also with the widespread use of air conditioners.

One important reason that leads to the problem of power shortage is the lack of generated electricity compared to the required quantities for consumption. Since the normal value to meet the consumption is estimated at about 30 000 MegaWatts but the actual production is no more than 23 500 MegaWatts.

Due to the absence of regular maintenance of power stations and shortage of fuel needed to operate the stations at full capacity.

And because of people preventing passage of gas lines for the establishment of a high pressure towers next to their homes for fear of disease.

The blackout in Egypt was due to several factors and not one specific factor, such as temperature, which greatly affects the ability of the stations, as well as lack of fuel and the targeting of electrical networks and attempt to sabotage, noting that summer period is considered one of the most periods in consumption of electricity due to the severity of the heat in it, what leads to increased loads.

## **Solution**

The apartment is the building block of each stage, We can summarize the usages of many apartments which are on the same building and make them all in one graph for this building then try to summarize many near buildings to form a region and evolve to a whole city.

We can take a massive advantage of this fusion where we can know which are the highest-cost regions and force them to decrease their usage and rationalize them or to increase the fees of usage for this specified region; In order not to force the other regions to decrease their use to decrease the overall usage of the city or the area.

## **Statistics**

Now we will state some statistics which show the shortening of the consumed and the produced amounts of electricity in Egypt.

We all suffered from the shortage of power and we spent many hours per day without electricity which was a big deal for us and a main reason to motivate us to work in solving this problem by our product and participate in rationalizing our electricity consumption.

Not only electricity shortage but also our need to know in which tier and level we are consuming power and how we are paid for our usage , Which device consumes the most power in our home or place ; All of these reasons and needs were a big motivation too to choose this project .

## **Business Opportunities**

This web and mobile application can be used by the company producing the device we've talked about earlier so each user having the device will have his own account in both so he can keep up with changes and updates with his usage wherever he is.

## Evolution

There is another advantage for our project but this advantage will appear in the next stages when we can graph the usage of a region or an area , We expect our usage then we check how near was our expectation , So by this way we can find and detect if there is any overloads in the used quantity due to electricity theft in streets and illegal shops which had greatly appeared in this period of time ; Therefore we can arrest those people by this way of detection of extra usage which we don't consume.

Later we can compute the usage of a whole building and may be updated for a whole region, so we have a graph for a city and then a graph for a country to see if the consumed level of electricity is equal to, greater or less than the produced level.

So we can judge and plan to build more electricity stations or no need for this and just try to keep the produced amount constant with little deviation.

Finally our product also helps in anti-theft of electricity not just monitoring it.

## ***Chapter Two: Used Technologies Background***

**In this chapter we are going to introduce some technologies and programming languages we've used. If you see yourself a developer ninja you can skip this chapter and come back upon facing a problem in our code or functions.**

**Let's have fun ...**

# 1- HTML5

## Definition

HyperText Markup Language(HTML) is a new encoded hypertext language which describes the internal structure of a web page which based on mixed pictures, animation, sound and video. It also converting any text into active readable text can be displayed and used on the web by organizing the words into smaller processed portions which helping you recognize the significance of font sizes, spacing, capitalization and punctuation.

HTML5 is the latest iteration of HTML that designed with high simplicity and flexibility. It consists of multiple encoded markers called tags which differentiate the text portions and separate them to be readable texts. HTML tag defines what the programmer means and writes inside the portion in other words the nature of a portion provided by vital information about it.

Application Programming Interfaces (APIs) are standardized to help the web applications work with HTML5 through an editor where we will be able to use a web app working with stored files on computer, store offline application data ,edit web documents directly in the browser, and more but it will not be covered in detail in this book.

## Advantages

Here are some features of HTML5 that make it the most widely used language for describing and designing any web page

### 1- Legacy/Cross Browser Support

HTML5 is being built to make things easier and friendly supported by the most of the popular browsers (Chrome, Firefox, Safari, IE9 and Opera).

### 2- Smarter Storage

Local storage feature is a mixture between a client-side database and regular old cookies, but it's better than cookies as it can stores across multiple windows. Local storage has better performance and security and data will be saved after closing the browser so you don't have to worry about the user deleting cookies as it's essentially a client side data base.

### 3- Cleaner Code

HTML5 allows you to write simple, clear, elegant and readable code, HTML5 allows you to keep your CSS organized and to code your markup cleaner with the new <section>,<aside>, <header>,<article>, <footer> and <nav>tags.



- **<article>** – this tag defines an article, a user comment or a post, so an independent item of content
- **<section>** – this is another important new syntax, as it can define any kind of sections in your document. It works pretty much like a *div* which separates different sections.
- **<aside>** – the *aside* tag marks content aside from the page content, which for example could be a lateral sidebar
- **<canvas>** – the *canvas* tag is quite exciting, as it allows drawing graphics via scripting (mostly JavaScript, but some others can be employed as well)
- **<embed>** – this new tag defines a container for interactive content (plugin) or external application
- **<header>, <footer>** – you won't need to manually name IDs for headers and footers, as now you have a pre-defined tag for them
- **<nav>** – the navigation can now be placed in the markup in between the *nav* tags, which will automatically make your lists act like navigation
- **<audio>, <video>** – these two obviously mark sound or video content, which will now be easier to run by devices.

#### 4- Better Interactions

As the user wants to enjoy/interact the content instead of looking at it, we want more interactions with `<canvas>` tag, with a great APIs allow you to build more dynamic web application some native APIs examples (document editing, browser history management, (DND) drag and drop and offline storage database).

## Examples

There are some examples on how we can upload an image, audio, video, input type password, input type color with color picker, local storage and tables...etc.

**The desired uploaded media should be found in the same created web page directory on your PC or Laptop.**

### 1- An image height and width using attributes and CSS

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
width:100%;
}
</style>
</head>
<body>
```

```
<p>It is better to use the style attribute (instead of the width and height attributes), because it prevents internal or external styles sheets to change the original size of an image:</p>
```

```


```

```
</body>
</html>
```

It is better to use the style attribute (instead of the width and height attributes), because it prevents internal or external styles sheets to change the original size of an image:



**Figure 3: Real and changed image size**

## 2- An image as a link

```
<!DOCTYPE html>
<html>
<body>

<p> <a href="http://www.facebook.com"></a></p>

</body>
</html>
```



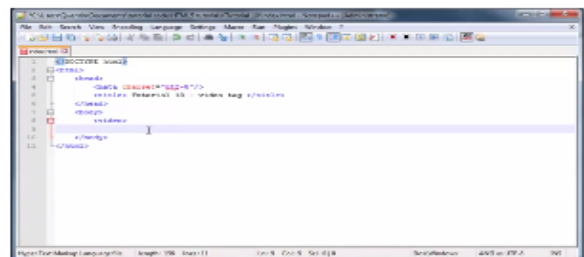
## 3- Uploading an audio

```
<!DOCTYPE html>
<html>
  <body>
    <audio controls loop>
      <source src="a5startk.mp3" type="audio/mpeg"/>
    </audio>
  </body>
</html>
```



## 4- Uploading a video

```
<!DOCTYPE html>
<html>
  <body>
    <video width="300" height="300" controls>
      <source src="New video.mp4" type="video/mp4"/>
    </video>
  </body>
</html>
```



## 5- Input type password

```
<!DOCTYPE html>
<html>
<body>

<form action="">
User name:<br>
<input type="text" name="userid">
<br>
User password:<br>
<input type="password" name="psw">
</form>

<p>The characters in a password field are masked (shown as asterisks or circles).</p>

</body>
```

User name:

User password:

The characters in a password field are masked (shown as asterisks or circles).

## 6- Input type color - with color picker

```
<!DOCTYPE html>
<html>
<body>

<p>
Depending on browser support:<br>
A color picker can pop-up when you enter the input field.
</p>
<form action="action_page.php">
  Select your favorite color:
  <input type="color" name="favcolor" value="#AA00AA">
  <input type="submit">
</form>
</body>
</html>
```

Depending on browser support:

A color picker can pop-up when you enter the input field.

Select your favorite color:

## 7- Local Storage (Store a counter permanently)

```
<head>
<script>
function clickCounter() {
    if(typeof(Storage) !== "undefined") {
        if (localStorage.clickcount) {
            localStorage.clickcount =
Number(localStorage.clickcount)+1;
        } else {
            localStorage.clickcount = 1;
        }
        document.getElementById("result").innerHTML = "You have
clicked the button " + localStorage.clickcount + " time(s).";
    } else {
        document.getElementById("result").innerHTML = "Sorry, your
browser does not support web storage...";
    }
}
</script>
</head>
<body>
<p><button onclick="clickCounter()" type="button">Click me!</button>
</p>
<div id="result"></div>
<p>Click the button to see the counter increase.</p>
<p>Close the browser tab (or window), and try again, and the counter
will continue to count (is not reset).</p>
</body>
```

Click me!

Click the button to see the counter increase.

Close the browser tab (or window), and try again, and the counter will continue to count

## 8- Tables with different style using class I

```
<html>
<head>
<style>
table, th, td {
border: 1px dotted ;
padding: 5px;
}
table {
border-spacing: 20px;
}
</style>
</head>
<body>
<table style="width:100%">
<tr>
<td>Hamed</td>
<td>Abd El Aziz</td>
<td>90</td>
</tr>
<tr>
<td>Mohamed</td>
<td>Attya</td>
<td>94</td>
</tr>
<tr>
<td>Mahmoud</td>
<td>Samy</td>
<td>85</td>
</tr>
</table>
</body>
</html>
```

Hamed	Abd El Aziz	90
Mohamed	Attya	94
Mahmoud	Samy	85

## 2- CSS3

### Definition

Cascading style sheet (CSS) is a set of instructions each one of them can teach the browser how to draw a specific element on the page. They can be used to define table sizes, text styles, font properties, box (padding, bordering and margin), changing background images, styling links and lists, and other aspects of Web pages that may be done through HTML5 or by both of them.

CSS helps the Web developers create the same look for different pages of a website. In HTML Web developers were defining the style of each block of text and of each table, while in CSS the Web developers have to define the commonly used styles only once in a CSS document. Once the required style is defined in a cascading style sheet, the Web developers can use it in any page which references the same CSS file. For example, a Web developer wants to increase the default text size of 30 pages of a Web site from 12pt to 15pt to make it more readable, that will be easy on him as if he referenced the same style sheet for all the pages, he only needs to change the text size on the style page and all the pages will show the larger text.

CSS is a new feature added to HTML that gives both users and Web Developers controllability over displaying the Web pages. CSS can be used to define the style of table cells, colors of a table's border, rounded corners, these created style sheets define the appearance of different elements, like headers and links

## Examples

### 1- Element and ID selector

```
<html>
<head>
<style>
.center {
  text-align: center;
  color: red;
}
#paragraph1 {
  text-align: left;
  color: blue;
}
</style>
</head>
<body>
<h1 class="center">Red and center-aligned heading</h1>
<p class="center">Red and center-aligned paragraph.</p>
<p id="paragraph1"> Hello World! </p>
<p>This paragraph is not affected by the style.</p>
</body>
</html>
```

## Red and center-aligned heading

Red and center-aligned paragraph.

Hello World!

This paragraph is not affected by the style.



## 2- Define font properties

```
<html>
<head>
<style>
h1{
font-family:Tahoma;
font-weight:bold;
font-style:italic;
font-size:30px;
}
p{
font-family:Arial;
font-weight:100;
font-style:normal;
font-size:30px;
}
</style>
</head>
<body>
<h1>
Soccer is the most popular sport all over the world.
</h1>
<p>
El-Ahly is my favourite team.
</p>
</body>
</html>
```

***Soccer is the most popular sport all over the world.***

El-Ahly is my favourite team.

### 3- Set the background color of different elements

```
<html>
<head>
<style>
h1 {
  background-color:#A52A2A;
}

p {
  background-color: #e0ffff;
}

div {
  background-color: #40E0D0;
}
</style>
</head>
<body>
<h1>CSS header background color example!</h1>
<div>
This is a text inside a div element.
<p>This paragraph has its own background color.</p>
We are still in the div element.
</div>
</body>
</html>
```

## CSS Header background color example!

This is a text inside a div element.

This paragraph has its own background color.

We are still in the div element.

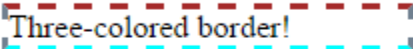
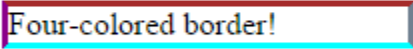
#### 4- Set an image as the list-item marker

```
<html>
<head>
<style>
ul {
  list-style: square inside url("square.gif");
}
</style>
</head>
<body>
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>
</body>
</html>
```

- Coffee
- Tea
- Coca Cola

## 5- Set the width, style and color of the border

```
<html>
<head>
<style>
p.two {
  width: 200px;
  border-style: outset;
  border-color: #A52A2A #0000ff;
  border-width: thick;
}
p.three {
  width: 200px;
  border-style: dashed;
  border-color: #A52A2A #708090 #00FFFF;
  border-width: medium;
}
p.four {
  width: 200px;
  border-style: solid;
  border-color: #A52A2A #708090 #00FFFF #8B008B;
  border-width: 1px;
}
</style>
</head>
<body>
<p class="two">Two-colored border!</p>
<p class="three">Three-colored border!</p>
<p class="four">Four-colored border!</p>
<pre><b>Note:</b> The "border-color" property does not work if it is used alone.
  Use the "border-style" property to set the borders first.</pre>
</body>
</html>
```

A rectangular box with the text "Two-colored border!". The box has a thick, double-line border. The top and bottom lines are dark red, and the left and right lines are dark blue.A rectangular box with the text "Three-colored border!". The box has a dashed border with three colors: dark red on the top and bottom, dark blue on the left and right, and cyan on the top and bottom.A rectangular box with the text "Four-colored border!". The box has a solid border with four colors: dark red on the top and bottom, dark blue on the left and right, cyan on the top and bottom, and magenta on the left and right.

**Note:** The "border-color" property does not work if it is used alone.  
Use the "border-style" property to set the borders first.

## 6- Pseudo - classes (match the first I element in all p elements)

```
<html>
<head>
<style>
p i:first-child {
    color: green;
}
</style>
</head>
<body>
<p><i>Electricity</i> management project. <i>Electricity</i> monitoring project.</p>
<p><i>Electricity</i> management project. <i>Electricity</i> monitoring project.</p>
<p><b>Note:</b> For i:first-child to work in IE8 and earlier, a DOCTYPE must be declared.</p>
</body>
</html>
```

*Electricity* management project. *Electricity* monitoring project.

*Electricity* management project. *Electricity* monitoring project.

**Note:** For i:first-child to work in IE8 and earlier, a DOCTYPE must be declared.

## 7- Pseudo - classes (match all I elements in all first child p elements)

```
<html>
<head>
<style>
p:first-child i {
    color: red;
}
</style>
</head>
<body>
<p><i>Electricity</i> management project. <i>Electricity</i> monitoring project.</p>
<p><i>Electricity</i> management project. <i>Electricity</i> monitoring project.</p>
</body>
</html>
```

*Electricity* management project. *Electricity* monitoring project.

*Electricity* management project. *Electricity* monitoring project.

**Note:** For i:first-child to work in IE8 and earlier, a DOCTYPE must be declared.

## 3- JavaScript

### Definition

JavaScript is a programming language used to make web pages interactive. It runs on your visitor's computer and doesn't require constant downloads from your website. JavaScript is often used to create polls and quizzes.

### JavaScript Uses

#### 1- Java script can validate input

```
<html>
<body>
<p>Please input a number between 1 and 10:</p>
<input id="numb" type="number">
<button type="button" onclick="myFunction()">Submit</button>
<p id="demo"></p>
<script>
function myFunction() {
    var x, text;
    // Get the value of the input field with id="numb"
    x = document.getElementById("numb").value;

    // If x is Not a Number or less than one or greater than 10
    if (isNaN(x) || x < 1 || x > 10) {
        text = "Input not valid";
    } else {
        text = "Input OK";
    }
    document.getElementById("demo").innerHTML = text;
}
</script>
</body>
</html>
```

Please input a number between 1 and 10:

Input OK

Please input a number between 1 and 10:

Input not valid

## 2- JavaScript Can Change HTML Content:

This example uses the method to "find" an HTML element (with id="demo") and changes the element content (**innerHTML**) to "Hello JavaScript":

```
<html>
<body>
<h1>What Can JavaScript Do?</h1>
<p id="demo">JavaScript can change HTML content.</p>
<button type="button"
onclick="document.getElementById('demo').innerHTML = 'Hello JavaScript!'">
Click Me!</button>
</body>
</html>
```

JavaScript can change HTML content

Click Me!

Hello JavaScript!

Click Me!

## 3- JavaScript Can Change HTML Attributes:

This example changes an HTML image, by changing the src attribute of an <img> tag:

```
<html>
<body>
<h1>JavaScript Can Change Images</h1>

<p>Click the light bulb to turn on/off the light.</p>
<script>
functionchangeImage() {
var image = document.getElementById('myImage');
if (image.src.match("bulbon")) {
```

```
image.src = "pic_bulboff.gif";
    } else {
image.src = "pic_bulbon.gif";
    }
}
</script>
</body>
</html>
```



Click the light bulb to turn on/off the light.



Click the light bulb to turn on/off the light.

#### 4- JavaScript Can Change HTML Styles (CSS):

Changing the style of an HTML element, is a variant of changing an HTML attribute:

```
>html<
```

```
>body<
```

```
>h1>What Can JavaScript Do?</h1<
```

```
<p id="demo">JavaScript can change the style of an HTML element.</p>
```

```
>script<
```



```
functionmyFunction(){
var x = document.getElementById("demo
x.style.fontSize = "25px
x.style.color = "red
{
/>script<
>button type="button" onclick="myFunction()">Click Me!</button<
/>body<
/>html<
```

JavaScript can change the style of an HTML element.

Click Me!

JavaScript can change the style of an HTML element.

Click Me!

## 4- JQuery

### Definition

jQuery is a JavaScript library designed to simplify development. It solves the complexity of many tasks, and cross-browser problems. It is very popular framework which is installed in around 60% of the 10.000 most visited Web sites.

jQuery performs so many tasks and functions with a "write less, do more" philosophy. Using jQuery, JavaScript programming is easier and more funnier. Advanced features are easily added allowing more sophisticated programs and web sites.



Figure 4: jQuery And Its Tagline

Various web browsers run JavaScript codes differently, many tasks that may run in Firefox for example, may be unresponsive in Internet Explorer, one of the main advantages of using jQuery that it spares many hours of code testing on different browsers and different machines.

Since jQuery is based on JavaScript – a client-side language, it manipulates the web page after it has been sent to the browser. The Ajax technique (Asynchronous JavaScript and XML) can be used to send and retrieve data from a server without refreshing the entire web page.

jQuery is open source, so it can be used by anyone.

### What Can jQuery Do?

jQuery hosts an array of file sources designed to allow developers build websites with greater ease.

The library tasks can be accomplished through single lines of code, which are converted to tens and hundreds of lines of complex JavaScript code. Many features can be used for doing a lot of things like:

- HTML/DOM manipulation
- CSS manipulation
- Effects and animations
- HTML event methods
- AJAX calls
- Plugins and utilities

DOM is an API that can be used to interface with the HTML document and interact with its objects. jQuery made it robust and efficient to select a precise piece of the document that requires to be inspected or manipulated.

jQuery allow changing the location of any element, as well as its appearance and behavior. Several special effects can be easily made with jQuery such as: Hide, Show, Animate, Fade In, Fade Out, Slide Down, Slide Up, and many others.

Events are defined to trigger or start the effect, events such as:

- load: The effect happens when an element has finished loading.
- mouseover: The effect happens when the cursor moves on a certain location on a page.
- mouseout: The effect happens when the cursor moves off a certain location on a page.
- change: The effect happens when a specific value of something changes.

jQuery use Ajax techniques to allow creating highly interactive web sites and applications. Ajax (Asynchronous JavaScript and XML) are a several technologies that offer several features, most importantly it allows fetching data from a server without refreshing the page. The way information can be retrieved behind the scene makes web applications feel and act like a desktop application.

Just as PhoneGap, Plugins can be included that offer expanding the jQuery basic library. Plugins are JavaScript programs that can be used to create fascinating effects. Many hundreds of plug-ins are free and available to use, others are constantly being developed to add new abilities.

## jQuery UI

jQuery UI is a set of plugins that add new functionalities to the jQuery core library. It contains a very useful set of widgets, interactions, themes, and animated visual effects. It's easy to use and very popular just like the basic jQuery library. The corresponding mobile-optimized plugin is jQuery Mobile.

## jQuery Mobile

It is a JavaScript library and web framework, built on jQuery and jQuery UI foundation. It helps designers and developers to create mobile web experiences easily.

jQuery is mainly focused on development for smartphone and tablet platforms, with its wide variety. Hence it is optimized for lightweight and has limited dependencies to enhance speed.

It has also a flexible theming framework that allows the creation of custom themes.

jQuery is currently sponsored and supported from many of the biggest companies in the industry, including the following: Adobe, HP Palm, Mozilla Corporation, Nokia, BlackBerry/Rim, DeviceAtlas, and dotMobi.



**Figure 5 :jQuery Mobile**

## 5- Json

### Definition

JSON is short for JavaScript Object Notation, and is a way to store information in an organized, easy-to-access manner. In a nutshell, it gives us a human-readable collection of data that we can access in a really logical manner.

JSON: JavaScript Object Notation.

JSON is a syntax for storing and exchanging data.

JSON is an easier-to-use alternative to XML.

### Why JSON?

For AJAX applications, JSON is faster and easier than XML:

### Using XML

- Fetch an XML document
- Use the XML DOM to loop through the document
- Extract values and store in variables

### Using JSON

- Fetch a JSON string
- JSON .Parse the JSON string

**The JSON format is syntactically identical to the code for creating JavaScript objects.**

**Because of this similarity, instead of using a parser (like XML does), a JavaScript program can use standard JavaScript functions to convert JSON data into native JavaScript objects.**

## JSON Example:

```
<html>
<body>
<h2>JSON Object Creation in JavaScript</h2>
<p id="demo"></p>
<script>
var text = '{"name":"JohnJohnson","street":"Oslo West 16","phone":"555 1234567"}';
varobj = JSON.parse(text);
document.getElementById("demo").innerHTML =
obj.name + "<br>" +
obj.street + "<br>" +
obj.phone;
</script>
</body>
</html>
```

## Output

**John Johnson**  
**Oslo West 16**  
**555 1234567**

## 6- Phone Gap

### Introduction to PhoneGap

PhoneGap is a framework for building cross-platform applications using standard web technologies such as Hyper Text Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript. PhoneGap is very efficient and is open source and completely free to use, which might explain its popularity. The creators of PhoneGap have implemented it as a combination of native and web application technologies, the type of the application created using PhoneGap is called hybrid

application.

PhoneGap fills the gap between web development standards and mobile devices, to make it simpler for web developers to develop mobile applications. It allows the use of device hardware features, with JavaScript APIs provided to give access to capabilities like the camera, GPS / location, device information, sound and many others.

PhoneGap continues to add support for new devices and APIs, and add support for new hardware platforms. More than any other cross-platform mobile development tool, PhoneGap currently supports the following operating system platforms:

- Apple IOS (both iPhone and iPad)
- Google Android
- Microsoft Windows Phone
- HP/Palm webOS
- Nokia Symbian
- RIM BlackBerry
- Samsung bada
- Mozilla Firefox
- Linux Tizen (SDK 2.x)
- Ubuntu Touch



Figure 6 : Adobe PhoneGap

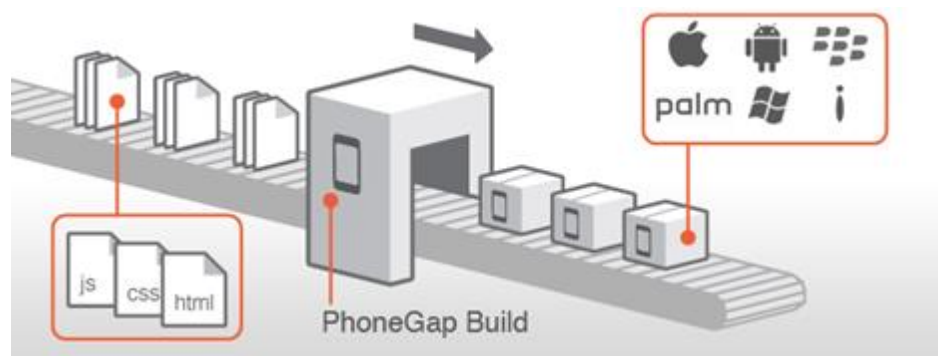


Figure 7 :PhoneGap Build Process

## PhoneGap and Apache Cordova

PhoneGap was originally invented by Nitobi ([www.nitobi.com](http://www.nitobi.com)), which started the project as a way to simplify cross-platform mobile development. It went on to win awards and has been used to develop many apps.

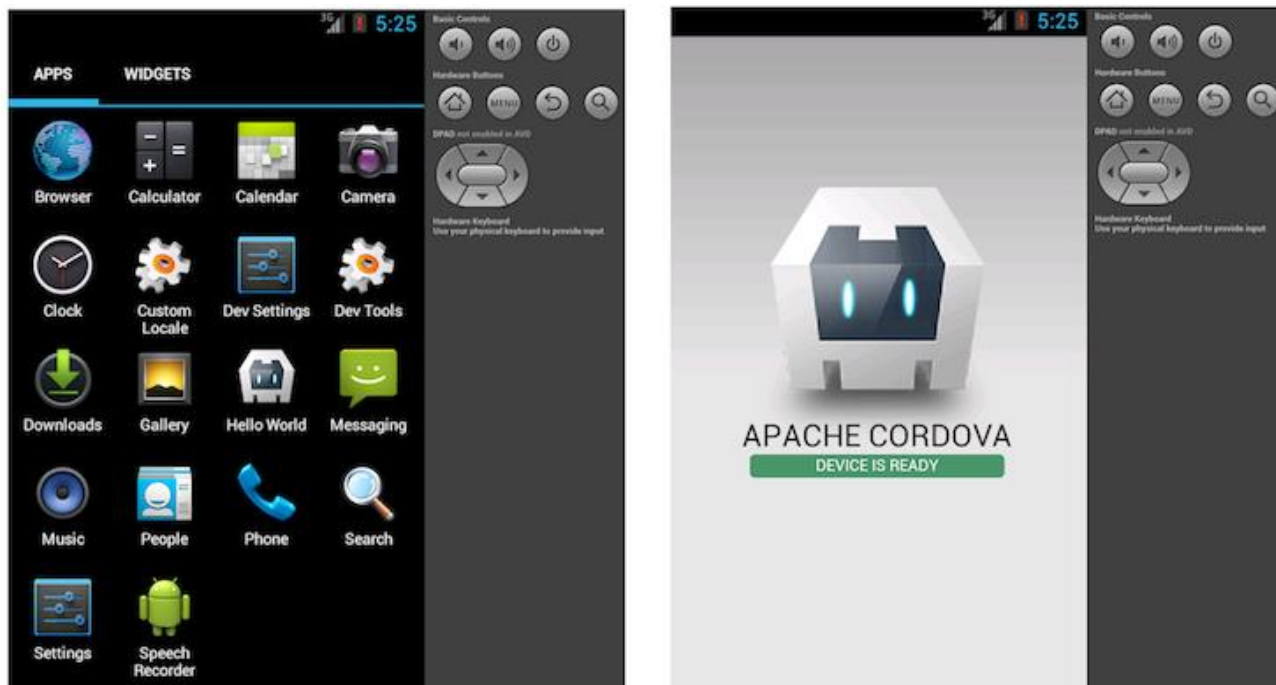
In 2011, when Nitobi was to be acquired by Adobe, the PhoneGap code was contributed to the Apache Software Foundation (ASF) under the project name Cordova, which is the name of the street in Vancouver where Nitobi's offices were located. The move to the Apache Software Foundation allowed everybody to easily contribute to the project, including companies and organizations.

Adobe PhoneGap is an open licensed distribution of Apache Cordova with some extra stuff added to it, in its core the Cordova container and API plugins upon which PhoneGap and its related services (compile, debug, emulate and build services) are built. The primary differences between Cordova and PhoneGap are the command-line tools.

Adobe is currently working on an enterprise version of PhoneGap, and is still a main investor in the project. More contributors include Google, IBM, Microsoft, Nokia, RIM, Intel, and Hewlett-Packard.



**Figure 8 : Apache Cordova**



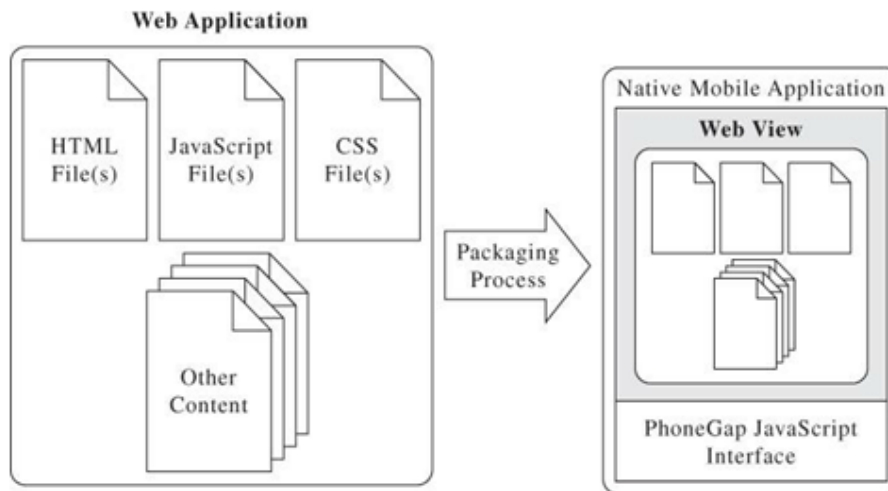
**Figure 9 : Apache Cordova Android Emulator**



## How PhoneGap Works

PhoneGap allows building native applications for mobile devices using HTML, JavaScript and CSS. These assets run in a Web view inside a native application container on the target platform. A Web view is a wrapper around the built-in browser of a mobile device that render web content, and allow communication with the native APIs and device OS.

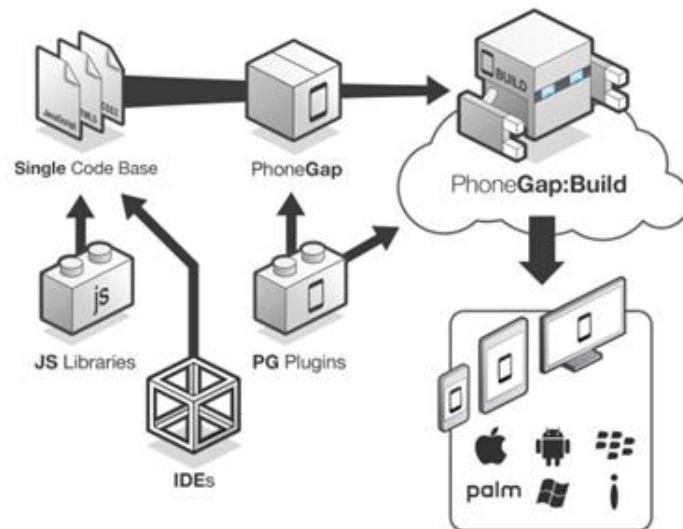
A developer creates a web application for mobile device, and PhoneGap packages the web application into a native application for each supported mobile platform. As Figure 10.



**Figure 10 :PhoneGap application architecture**

When the application launches, it loads the application's startup page into the web view and passes control to allow the user to interact with the web application.

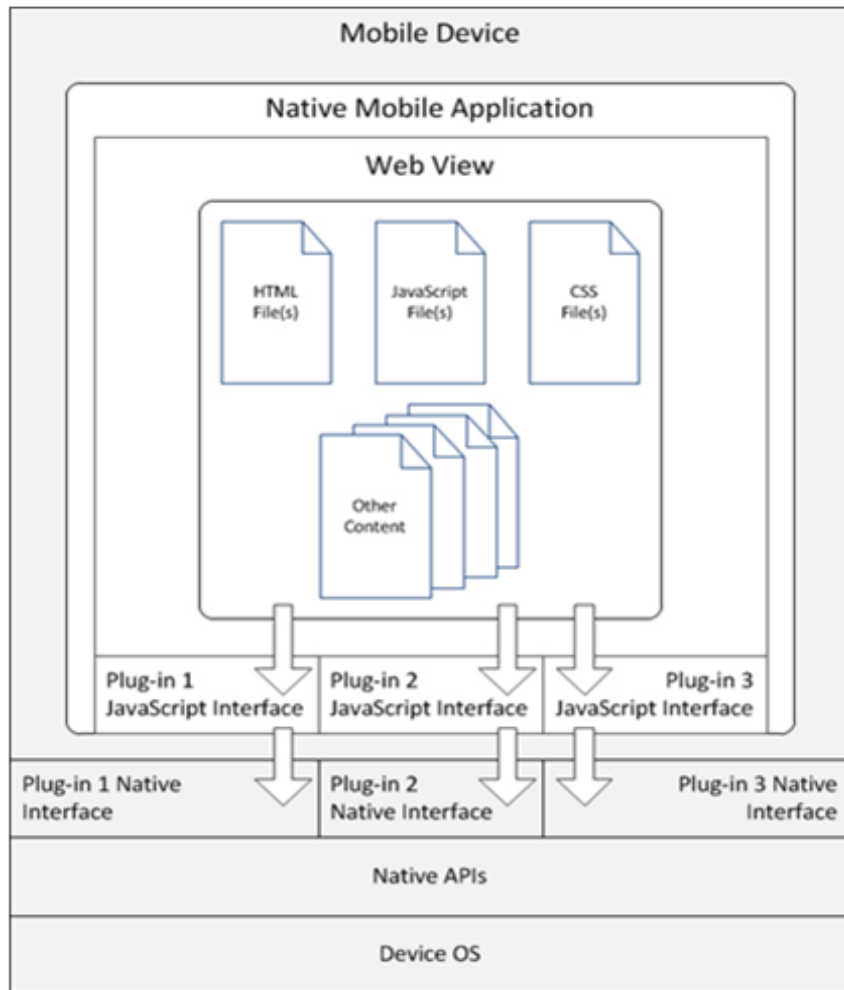
The normal web applications do not typically have access to device-level APIs. PhoneGap provide a suite of JavaScript APIs that a developer can use to allow a web application running within the Cordova container to access native device hardware features.



**Figure 11 :PhoneGap as a framework filling the gap**

**Cordova currently support the following feature APIs:**

- Accelerometer
- Camera
- Capture
- Compass
- Connection
- Contacts
- Device
- Events
- File
- Geolocation
- Globalization
- InAppBrowser
- MediaNotification
- Splashscreen
- Storage



**Figure 12 : Apache Cordova Native Application Architecture**

Plugins can be included to extend the existing functionality of the native-app Cordova container. Starting with the PhoneGap 3.x release, each of the APIs is implemented as a separate plugin, and can be installed and uninstalled using the command line interface (CLI) utility.

PhoneGap use a set of tools and SDKs (software development kits) for each of the platforms to process and manage the creation of application projects, building native applications, and testing applications on emulators.

PhoneGap Developer is an available mobile application that we have used, it gives developers the ability to code changes to their app from desktop/laptop, then serve and see those changes reflected on device immediately without having to rebuild it through the CLI, reinstall or code sign it.

In addition, PhoneGap Build is a cloud based build service, made to help users compile and package mobile applications when they upload their HTML5, CSS and JavaScript files.



**Figure 13 :PhoneGap Build Logo**

## 7- Android

### Definition

Android is the name of the mobile operating system made by American company; Google. It most commonly comes installed on a variety of smartphones and tablets from a host of manufacturers offering users access to Google's own services like Search, YouTube, Maps, Gmail and more.

This means you can easily look for information on the web, watch videos, search for directions and write emails on your phone, just as you would on your computer, but there's more to Android than these simple examples

### What can an Android phone do?

Android phones are highly customizable and as such can be altered to suit your tastes and needs with wallpapers, themes and launchers which completely change the look of your device's interface. You can download applications to do all sorts of things like check your Facebook and Twitter feeds, manage your bank account, order pizza and play games. You can plan events on from your phone's calendar and see them on your computer or browse websites on your desktop and pick them up on your phone.

Another neat feature of Android is that it automatically backs up your contacts for you. When you set up an Android phone you'll need to create a Google Account or sign in with an existing one. Every time you save a number to the address book of your Android phone it will be synced to your Google Account.

The benefit of this is that if you lose your phone all of your numbers will be saved. The next time you get an Android phone (or an iPhone or Windows Phone if you prefer) and sign in with your Google Account, all of your contacts and friend's numbers will be displayed in your new phone's address book immediately, no need to transfer or back them up anywhere else.

Syncing is a way for your phone to keep all your information; websites, contacts, calendar entries and apps up-to-date. This can happen over your phone's mobile data or WiFi connection, seamlessly, in the background.



**Figure 14 : Android Logo**

## 8- IOS

### Definition

iOS is Apple's mobile operating system used to run the popular iPhone, iPad and iPod Touch devices. Formerly known as the iPhone OS, there are over 900,000 iOS applications available for download in the Apple app store, the most popular app store of any mobile device. iOS utilizes a multi-touch interface where simple gestures are used to operate the device, like swiping your finger across the screen to move to the next page or pinching your fingers to zoom out.

iOS is the foundation of iPhone , iPad and iPod touch. It comes with a collection of apps that let you do the everyday things, and not-so-everyday things, a ways that are intuitive, simple and fun. And it's loaded with useful features you will wonder how you ever did without



Figure 15 :iOS Logo

## 9- Windows Phone

### Definition

A smartphone operating system from Microsoft, introduced in 2010. Using the touch interface first deployed on Microsoft's Zune music player, Windows Phone is a vast departure from the earlier Windows Mobile platform. Both Windows Mobile and the first versions of Windows Phone were based on the Windows CE operating system. Although Windows Phone arrived several years after the iPhone and Android platforms, most major apps and tens of thousands more are available for the platform.



Figure 16: Windows Phone Logo

## Types:

### **Windows Phone 7:**

The first WinPhone release was Windows Phone 7 in 2010. In 2011, Version 7.5, code-named Mango, added app switching and refined Office, Bing and social network integration. The Tango 2012 update included improved multitasking and text messaging.

### **Windows Phone 8:**

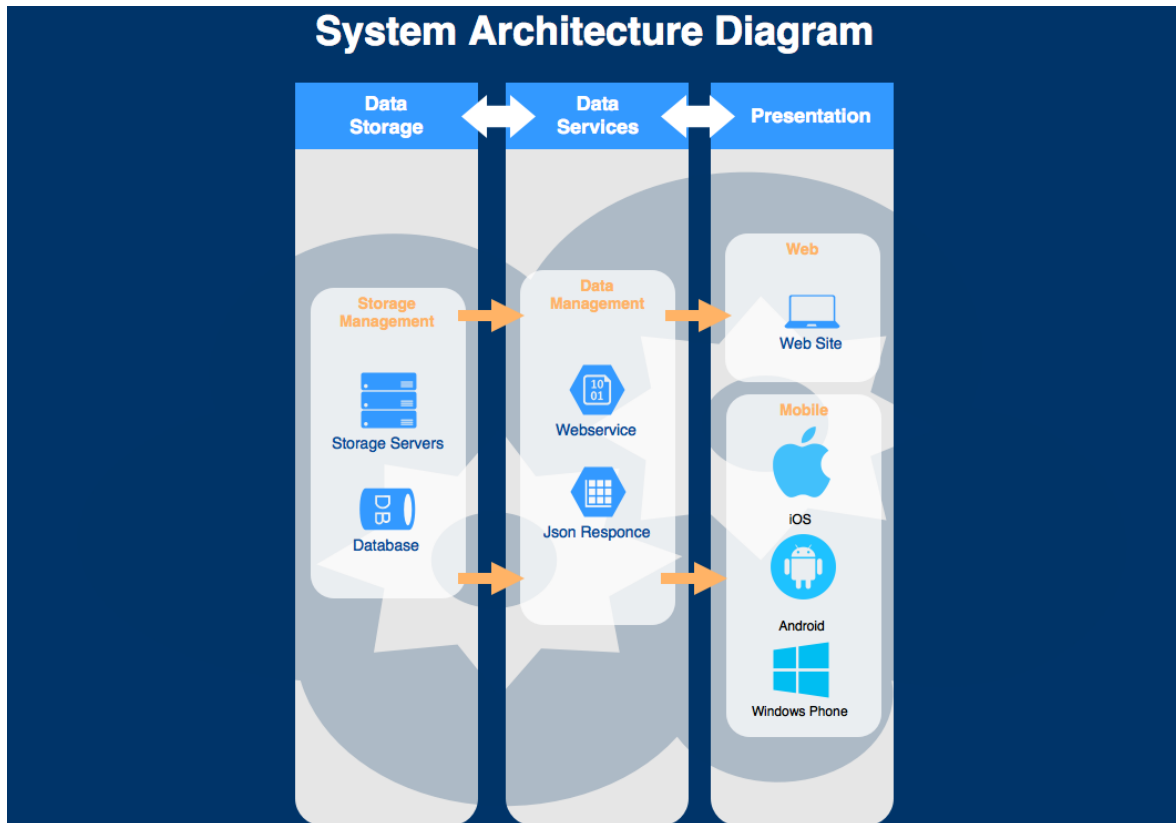
introduced in late 2012, Windows Phone 8 replaces the Windows CE operating system with a phone version of Windows 8 in order to align product architectures for developers.

## *Chapter Three: Implementation*

**And now let's get down to business as in this chapter we will explain how we could build our web and mobile application from the ground up.**

## Three main layers of every System Architecture

- 1- Data Storage Layer or Server Layer.
- 2- Web Service Layer.
- 3- Presentation Layer.



**Figure 17: System Architecture Diagram**

We have three main transitions between the three main layers. First we have the Data Storage layer or the Server Layer represented by the Storage Servers and the Data Base where the storage managed, then there is the Data Services layer represented by the Webservice and the Json Response where the data is managed and finally we have the presentation layer represented by the web application and the mobile application compatible with the most 3 used operating system in mobile phones (Google Android, Apple iOS and Microsoft Windows Phone) .



## System Development Layer

It is divided into 2 main sub layers

- 1- Design layer (Bootstrap ,HTML5 and CSS3 ).
- 2- Business Layer (JavaScript and JQuery ).

**-For the definition of any of these technologies or their roles please go to chapter two in this book -**

For the Design Layer we used a Bootstrap template with a full HTML5 and CSS3 code, lets now see a short note on Bootstrap Framework.

### Bootstrap

#### Introduction to Bootstrap

One of the biggest challenges in development is building web apps that keep its compatibility across various platforms, and screen sizes. That is what is meant by a responsive design, where the app should have the ability to adapt with the size of the display to grant the optimal viewing experience.

Bootstrap helps designers create professional, responsive, dynamic, and slick designs without browser compatibility problems, and simplify common tasks, so they may focus on the design and the content.

#### What is Bootstrap and why it is used?

Twitter Bootstrap is a rapidly evolving framework with a vast array of HTML components and a grid system that is used to build user interfaces of websites and applications. It was designed to solve many of the front-end problems faced while developing applications.

Bootstrap provides CSS solutions that are tried, tested, and future-friendly. Many features are built-in in the framework, such as the fluid grid layout, responsive design, custom forms elements, and advanced typographic capabilities.

It is among the most popular frameworks used. For its powerful features and reliable designs, it is used by designers that have as much as 10 years of experience in web development. It is also well documented in other languages such as Russian, Spanish, Chinese and Ukrainian languages. The web app project templates in Visual Studio 2013 employ Bootstrap by default.

Interface components that can be made with bootstrap include carousel, forms, buttons, tables, alerts, dropdowns, accordion, navigation, modals and tabs. The components used in Bootstrap have been tested in a variety of web browsers. So web pages are going to scroll well on any device, and for example, tables and graphs, which might be difficult to fit on smart phones and small screens, can be made best viewed using bootstrap.

Using Bootstrap, the design can be aesthetically customized on top of a solid foundation, so it represents quality as well as efficiency. Bootstrap's Leaner CSS (LESS) files can also be customized to our heart's content for more fascinating and innovative designs.

The designs made using Bootstrap are consistent, compatible with browsers, and saves time (with a predefined classes and templates).

Bootstrap is improving, and has an active and helpful community; it is available to download for free on Github.



Figure 18 : Bootstrap Framework

After using this bootstrap template containing the design layer of the web we get down to the business layer where we program all the pages we need in the web app writing a Javascript code for these screens, we use the Web service in this Javascript code then we get a Json response and show this response in the way we need according to what each page looks like.

We also used the concept of Responsive Layout –which is the most important feature in Bootstrap- in our design so it can be compatible with any displaying screen size, let's now see a short note on Responsive Layout Design.

## **Responsive Web Design**

### **What is Responsive Web Design?**

Responsive Web design is an approach whereby a designer creates a Web page that “responds to” or resizes itself depending on the type of device it is being seen through. That could be an oversized desktop computer monitor, a laptop, a 10-inch tablet, a 7-inch tablet, or a 4-inch smart phone screen.

The purpose of responsive design is to have one site, but with different elements that respond differently when viewed on devices of different sizes.

Let's take a traditional “fixed” website. When viewed on a desktop computer, for instance, the website might show three columns. But when you view that same layout on a smaller tablet, it might force you to scroll horizontally, something users don't like. Or elements might be hidden from view or look distorted. The impact is also complicated by the fact that many tablets can be viewed either in portrait orientation, or turned sideways for landscape view.

### **How Does Responsive Web Design Work?**

Responsive sites use fluid grids. All page elements are sized by proportion, rather than pixels. So if you have three columns, you wouldn't say exactly how wide each should be, but rather how wide they should be in relation to the other columns. Column 1 should take up half the page, column 2 should take up 30%, and column 3 should take up 20%, for instance.

Media such as images is also resized relatively. That way an image can stay within its column or relative design element.

## Related Issues:

### 1- Mouse V. Touch:

Designing for mobile devices also brings up the issue of mouse versus touch. On desktop computers the user normally has a mouse to navigate and select items. On a smart phone or tablet, the user mostly is using fingers and touching the screen. What may seem easy to select with a mouse, may be hard to select with a finger on a tiny spot on a screen. The Web designer must take “touch” into consideration.

### 2- Graphics And Download Speed:

Also, there's the issue of graphics, ads and download speed. On mobile devices, it may be wise to display fewer graphics than for desktop views so that a site doesn't take forever to load on a smart phone. Larger ad sizes may need to be exchanged for smaller ads.



Figure 19 : Figure 3.3 Responsive Layout

## **Mobile Application**

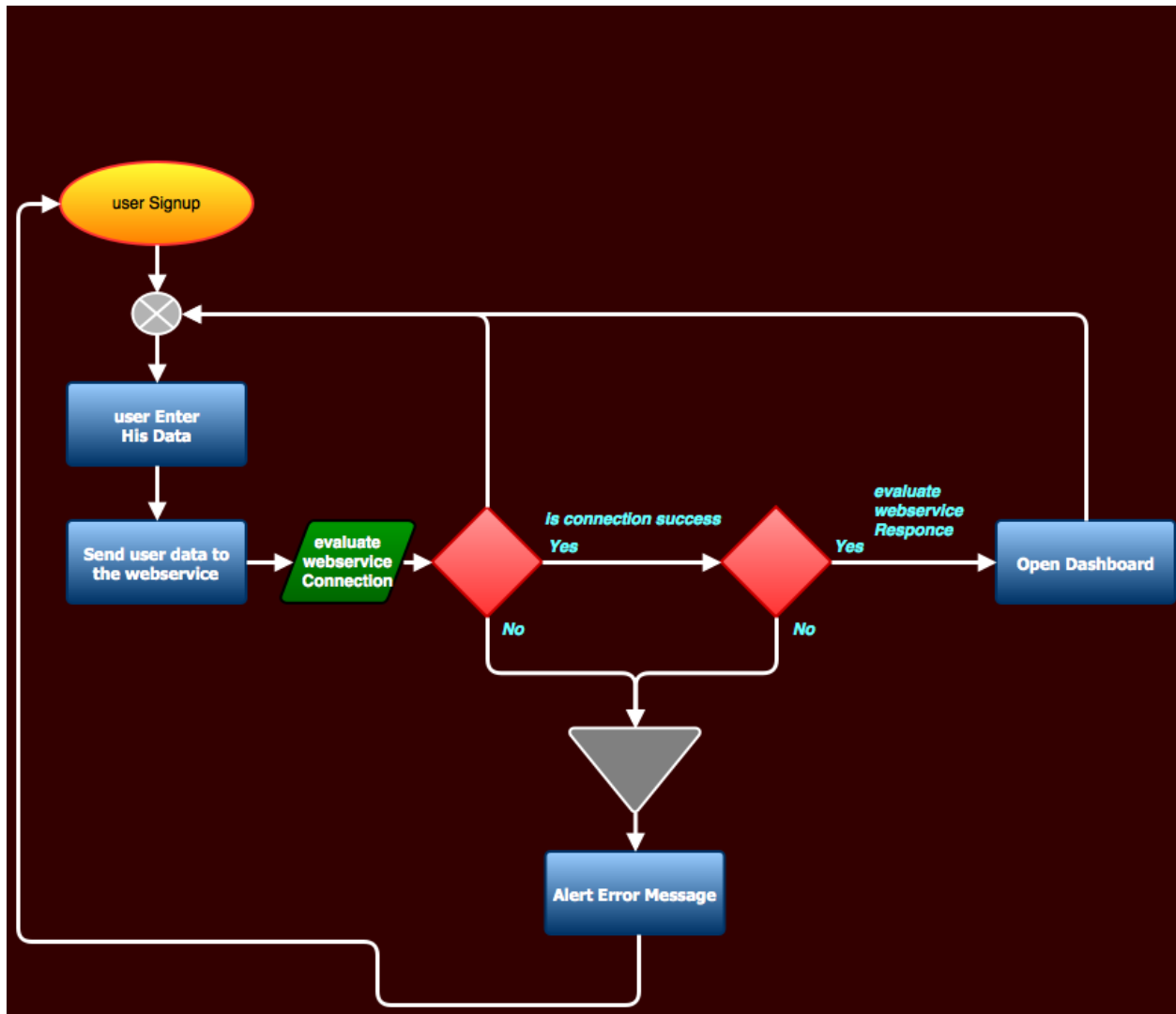
It is mainly based on Cross Platform technology and using PhoneGap Framework to give out a mobile application compatible with the most three worldwide used platforms which are Android, iOS and Windows Phone.

Here we used nearly the same Design Layer of the web application and PhoneGap is for mobile app implementation as it is used to build the three applications.

## *Chapter Four: Work Outputs And User Experience*

**Let's get our hands more dirty as in this chapter we will take a short tour among the outputs of our work and see some flow charts describing both the web and mobile applications. We will also go through a complete user experience using web and mobile application and see how will he move among the different pages. Every user will have many destinations, and pages he is authored to go into.**

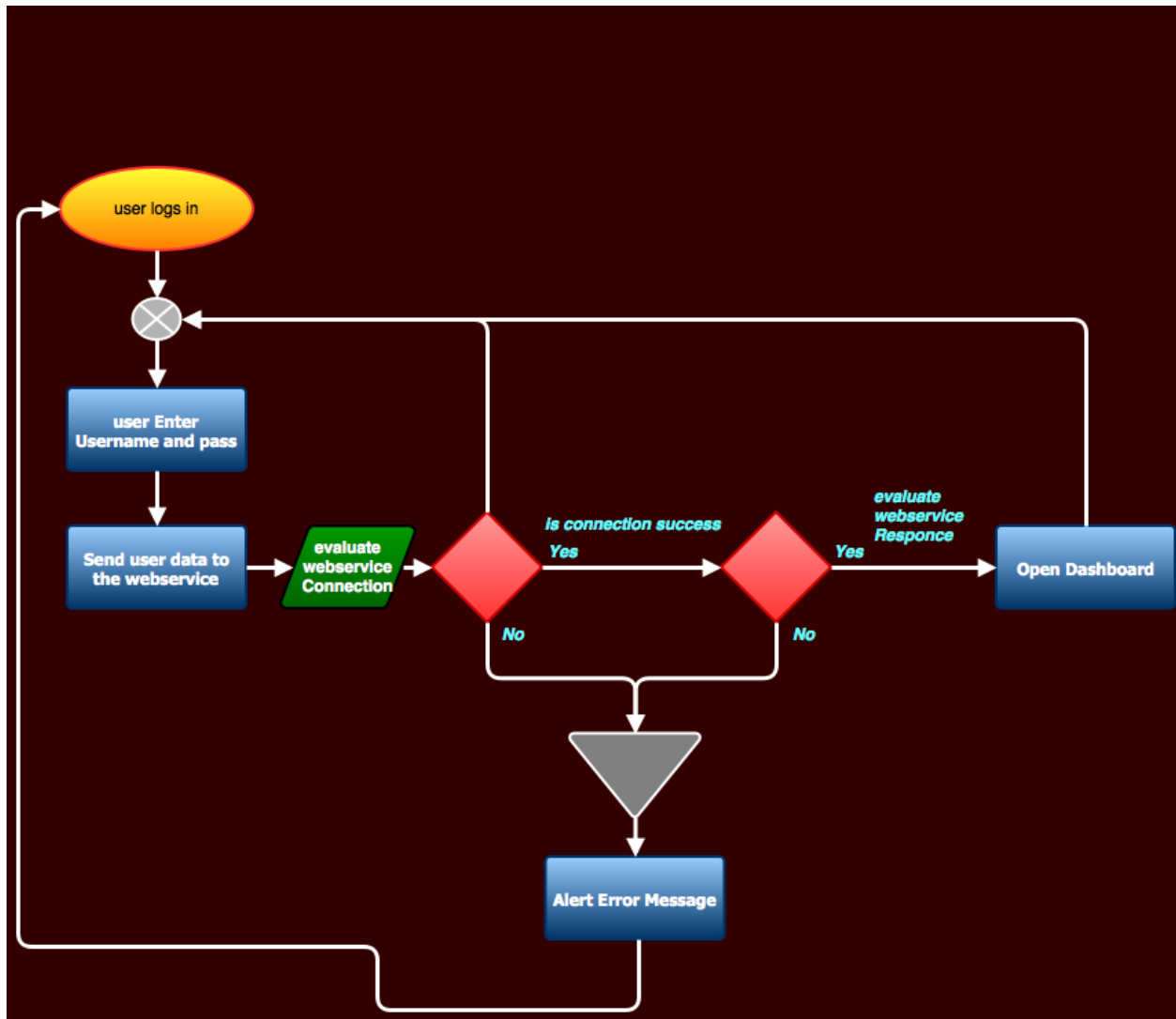
## 1- User Sign Up Flow Chart



The process is as the following :

User enters his data , data is then sent to the web service , web service connection is evaluated if succeeded proceed to web service response evaluation if also succeeded and got a response the dash board is opened . If an error happened in either the connection evaluation or the response evaluation then an alert error message is shown to the user.

## 2- User Login Flow Chart

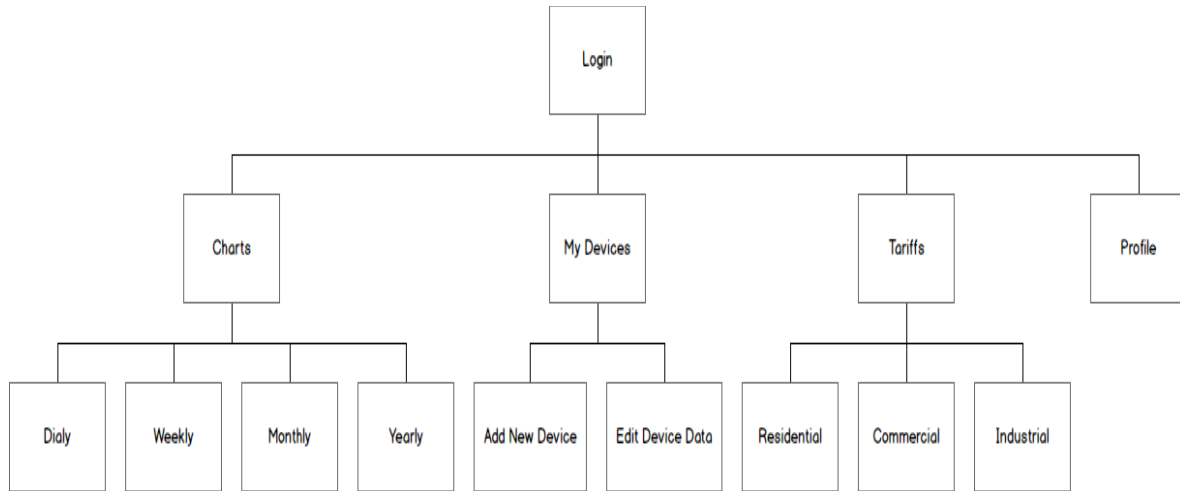


Login process is nearly the same as the sign up process:

User enters his username and password correctly, the data is then sent to the web service , web service connection is evaluated if succeeded proceed to web service response evaluation if also succeeded and got a response the dash board is opened . If an error happened in either the connection evaluation or the response evaluation then an alert error message is shown to the user.



### 3- Application Diagram



After the user successfully logs in he will have 4 places to go.

**1-Charts** : that are shown in the dash board for his usage where he can choose to show his data daily , weekly , monthly or yearly .

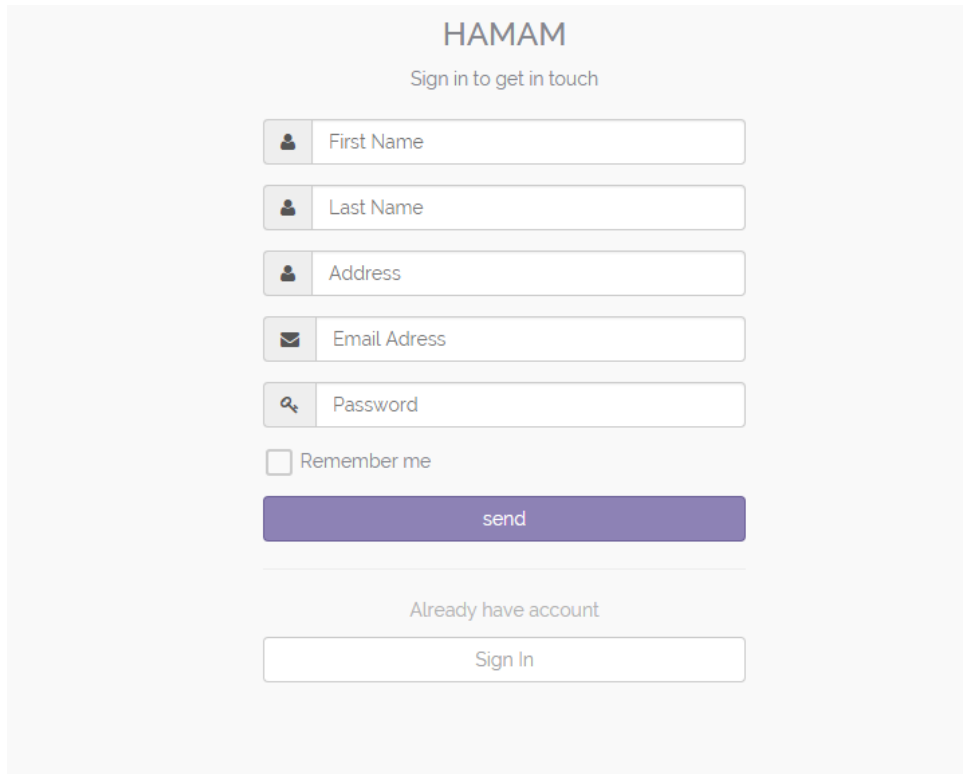
**2-My Devices** : where he can add a new device to his devices or update the data of an existing one .

**3-Tarrifs** : where he can find precisely the cost of each tier in his usage as assigned by the government -this data is entered by the admin and edited transparently each time the government changes the usage tarrifs- . Tarrifs are divided into three sections (Residential like home apartments, Commercial like shops, restaurants .. etc and Industrial like factories) .

**4-User Profile** : Where he can check the data of his profile as first and last name, his address, E-mail and he is capable of editing these data.

## User Experience:

### 1- Sign Up Screen.



The image shows a sign-up screen for 'HAMAM'. At the top, the brand name 'HAMAM' is centered in a bold, dark font. Below it, the text 'Sign in to get in touch' is centered in a smaller, lighter font. The form consists of several input fields, each with a small icon on the left: 'First Name' (person icon), 'Last Name' (person icon), 'Address' (person icon), 'Email Address' (envelope icon), and 'Password' (key icon). Below these fields is a checkbox labeled 'Remember me'. A prominent purple button with the text 'send' is positioned below the checkbox. A horizontal line separates this section from the bottom part of the screen, which contains the text 'Already have account' centered above a white button with the text 'Sign In'.

HAMAM

Sign in to get in touch

First Name

Last Name

Address

Email Address

Password

Remember me

send

Already have account

Sign In

## 2- Log In Screen.

# HAMAM

Sign in to get in touch

Remember me

send

---

Dont have account yet!

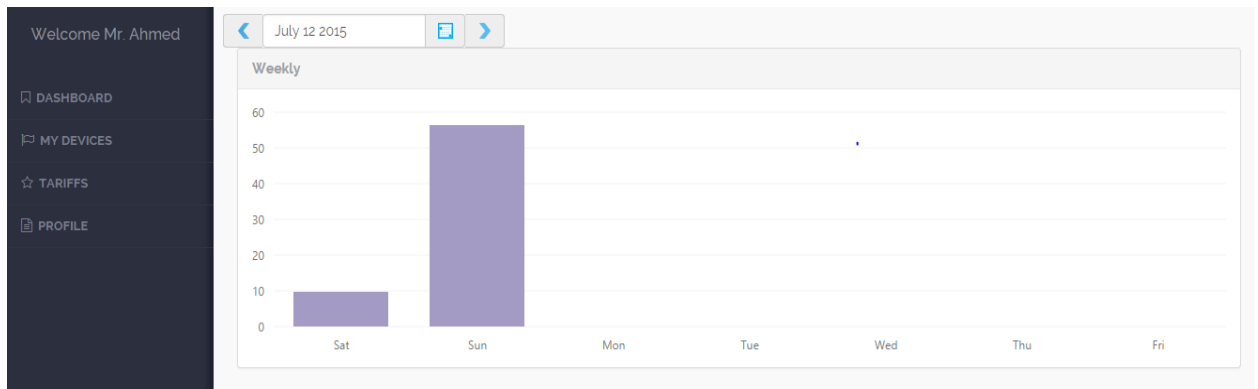
Create Account

## 3- Dash Board Screen.

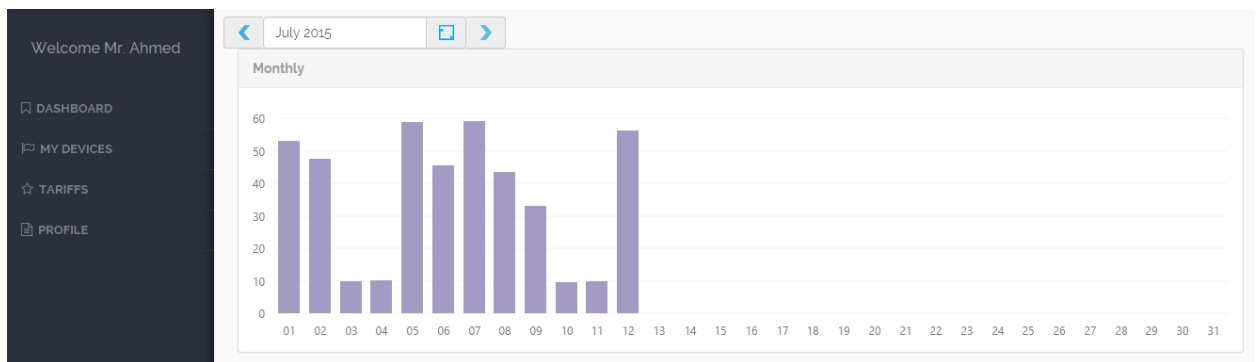
Daily.



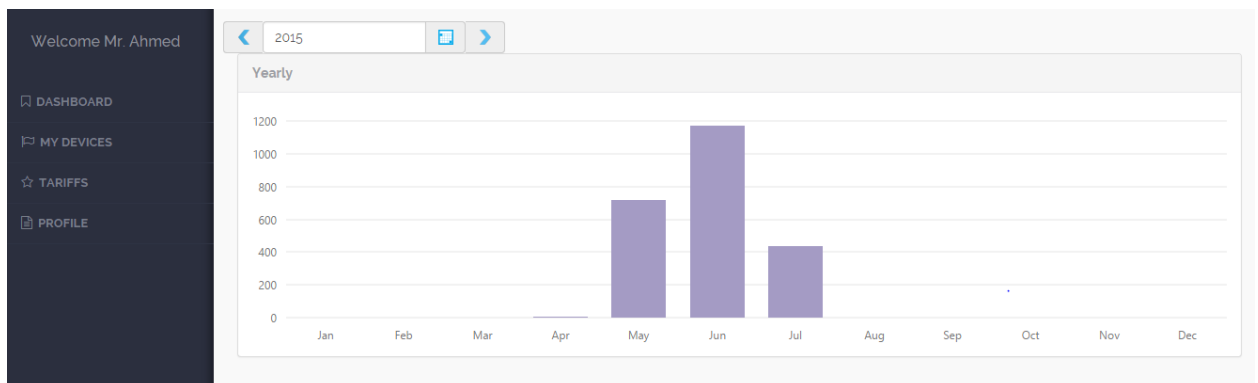
## Weekly.



## Monthly.



## Yearly.



## 4- My Devices Screen.

Welcome Mr. Ahmed

DASHBOARD

**MY DEVICES**

TARIFFS

PROFILE

### My Devices

Data

**Tariff**  
Residential


**Name**  
Home

**Description**  
This is home device

**Serial**  
2f6b7f572ed646agad7f5d7faeeafbef

**API**  
JiIMrJwWj24XPoxWF3BvtOgkNYf2FWo8uOqtKvgoRD1zF6Hh

**Location**



©2015 Google - Map data ©2015 Google, ORION-ME Terms of Use

Edit Device

Copyright © 2015 DTEAM

## 5- Tarrifs Screen.

### Residential.

Welcome Mr. Ahmed

DASHBOARD

MY DEVICES

**TARIFFS**

PROFILE

## Residential

Tier Name	Consumstion Range(KWh)		Price(EGP)	Taxes(EGP)
	From	To		
Tier 1	0	50	0.075	1
Tier 2	51	100	0.145	15
Tier 3	0	200	0.16	3
Tier 4	201	350	0.39	5
Tier 5	351	650	0.39	7
Tier 6	651	1000	0.68	10
Tier 7	1001	0	0.78	15

### Commercial.

Welcome Mr. Ahmed

DASHBOARD

MY DEVICES

**TARIFFS**

PROFILE

## Commercial

Tier Name	Consumstion Range(KWh)		Price(EGP)	Taxes(EGP)
	From	To		
Tier 2	101	250	0.44	4
Tier 3	251	600	0.59	6
Tier 4	601	1000	0.78	8
Tier 5	1001	0	0.83	10

## Industrial.

Tier Name	Consumstion Range(KWh)		Price(EGP)	Taxes(EGP)
	From	To		
Tier 1	51	100	0.3	2
Tier 2	101	250	0.44	4
Tier 3	251	600	0.59	6
Tier 4	601	1000	0.78	8

## 6- Profile Screen.

Welcome Mr. Ahmed

profile

Data

First Name :

Last Name :

Email :

Address :

[Edit Profile](#)

## *Chapter Five: Conclusion*

**During this chapter we will summarize all our steps and compare if we have reached our desired target or not, also we will briefly discuss what we have done and how we got the output of each portion.**



Any system architecture consists of three layers which are:

- 1- Server layer.
- 2- Web services layer.
- 3- Representation layer.

Starting from the design layer which is the code, the code is the base of the project .

We used the HTML5, CSS3, JAVASCRIPT and JQUERY in forming this design, if you are interested in more details; you can refer to chapter two where you can find a complete explanation for the above technologies.

Let us move to the second layer which is the web services layer, what is a web service?

A web service is a link between the design layer and the development layer where your written code can be synchronized with a DATABASE SERVER without directly accessing the database.

We have two sides:

- 1- Server side
- 2- Client side

Firstly, the server side is the upper part of the tree where no one can reach except the web service, the web service asks for a permission to post or get something to or from the database, when the permission is accepted and succeeded; the web service returns a message with the desired data or complete to the next step needed by the client side.

Secondly, the client side is the second leg for the web service which asks for data and waits to see if a response is returned or not.

So we are able to connect to the database (server side) through the different web services, each function which needs the data base permission requests for it through its web service and waits to see if this request fails or passes; then the received message is posted on screen.

Using this web and mobile application a user having the device will be able to see and rationalize his usage and nearly estimate his bill where ever he is, which in return helps in solving the electricity shortage problem in Egypt.

## ***Chapter Six: Related Works And Previous Attempts***

**In this section, we will have a look on similar technologies that are used worldwide, we will also take a tour among some companies and countries which started working on this issue and constructed some business to help the nation solving the shortage of electricity and power as well.**

**Also we will know the reasons of failure in some cases and on the other side we will get the keys of success for other experiments and how they dealt with the faced problems to reach their target and goal.**

**We can get and understand the people's way of thinking by studying and analyzing their treatment for their problems and issues; why they did so and was it worthy or not.**

**This will help u think and be creative in finding solution and introduce your way of thinking as you have seen a lot of opinions and diagnosis.**

## Alternate Hybrid Application Frameworks

Since the release of PhoneGap project, several other products have been released that use a similar approach for building hybrid mobile applications. With web technologies adding more device specific functionalities, the main motivation for these products and technologies is to simplify the building of cross platform applications.

### Worklight

Several enterprise platforms use a similar approach in mobile application development. Worklight is a commercial platform that employs PhoneGap in its core. The initial benefit of using enterprise mobile application platforms is that it includes additional server components and infrastructure with many capabilities. Worklight was owned by IBM in 2012, other commercial platforms include Antenna Volt and AT&T WorkBench.



Figure 20 : IBM WorkLight

### Appcelerator Titanium

Appcelerator titanium is an open source framework used for cross-platform mobile development. It works using web technologies and currently supports IOS and Android. Appcelerator tries to build more native applications than those produced by PhoneGap, with a JavaScript-based development environment where all the application's logic and user interface are entirely coded in JavaScript, which is interpreted on the target platform. The use of native user interfaces increases performance.



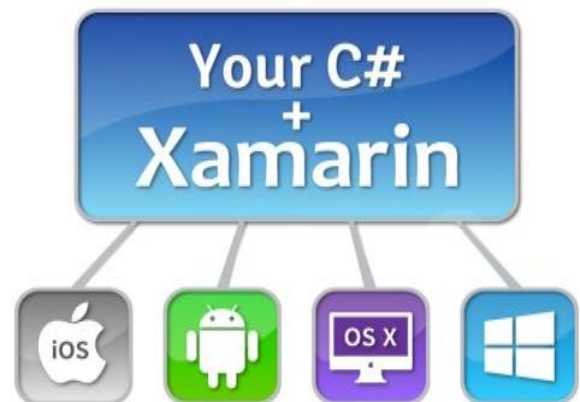
Figure 21 : Appcelerator Titanium

## Qt Creator

Qt Creator is the integrated software development environment that supports C++, JavaScript and QML. It allows developing Qt applications and deploying them across embedded, desktop (Windows, Mac, Linux) as well as mobile devices (iOS, Android, Symbian, Maemo, MeeGo) without rewriting the source code. Qt provides a set of libraries and development tools, and gives support to multiple languages.

## Xamarin

Xamarin platform was released in 2013, and recently it has announced with Microsoft a global partnership. Xamarin takes advantage of the advanced language features of C# and .NET library, to build native applications, with native user interfaces that works on IOS, Android, OS X, and Windows Phone. Xamarin aims for the quality and performance expected of native apps, and can be used with either Xamarin Studio or Visual Studio.



**Figure 22 : Xamarin development**

## Previous Attempts And Similar Smart Meter Solutions

There are various energy monitoring products in the market, we will focus on products that visualize the entire electricity usage of a household. The architecture used in these solutions consists of a smart electricity meter that senses and measures electricity usage, and sends data to a wireless display or a web gateway. The apps are similar in concept to our work, and many are based on web technologies.

### Wattson Solar Plus

A classic wireless energy monitor that displays both the electricity produced and used.

Features:

- Tracks and measures the energy consumption at any time and shows it in Watts, Kg carbon or Pounds.
- Displays the energy generation in numbers and colors.
- Uses sensors that clip around the electricity cables.
- Works with 210V-240V single phase or 400V-460V three phase electricity supplies.
- Works with all generation systems (solar, wind and water) and inverters.



Figure 23 : Wattson Solar Plus

### Onzo Energy Kit

This energy kit tracks how power is used, increasing consumers' understanding, and helping use less energy.

Features:

- A clear, simple wireless display with easily rechargeable batteries.
- Easy to install self-powered sensor.
- Online account and personalized website to examine energy usage.



Figure 24 : Onzo Energy Kit

- USB cable connects the display to PC.

## Power Cost Monitor

Features:

- Displays electricity consumption in dollars and kilowatt hours.
- Estimating function to predict monthly electricity cost.
- Appliance function to show electricity usage by a single appliance.
- Website and smartphone support through a WiFi Bridge accessory.



Figure 25: Power Cost Monitor

## Wattvision Power Monitor

Meter sensor, website and app that records and shows detailed information of usage.

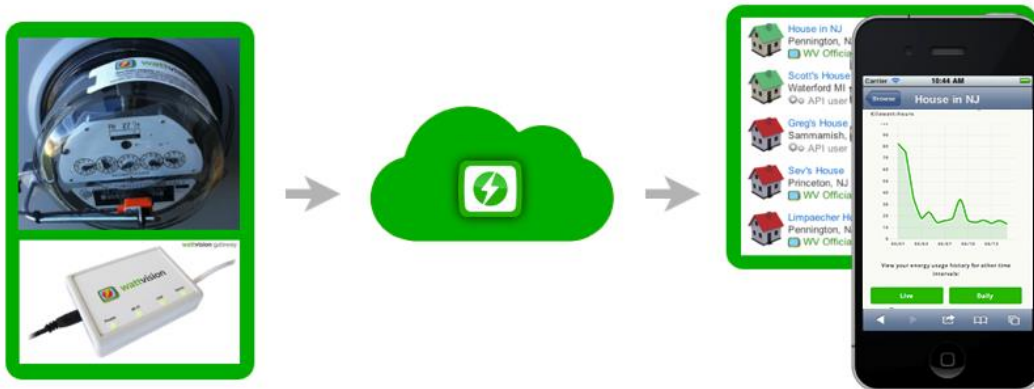


Figure 26 :Wattvision Power Monitor

Features:

- Sensor accuracy within 2 watts.
- Wifi enabled unit to send data directly to the Wattvision website, every 15 seconds.
- Easy setup and use.
- Smartphone version of the website to show energy usage on mobile devices.
- Viewing and comparing days and months of energy data, in watts and dollars.

- Ability to share data with other users, and show energy consumption in similar sites.
- Ranking and rewards that helps lowering energy usage.
- Email alerts on excessive consumption.

## Vattenfall Smart Meter App

Vattenfall is a major power company in northern Europe. By 2009, it has rolled out smart grids all across Sweden. Vattenfall Smart Meter App is a native app for IOS used in its grids.

Features:

- Instantaneous electricity reading.
- View consumption figures in day, week, or month.
- Overview of the current and estimated costs.
- Show the amount of CO2 emissions.
- Free to use with a demo mode available for interested persons outside of project areas.



Figure 27 : Vattenfall Smart Meter

## Hohm and PowerMeter



Figure 6.9 Microsoft Hohm

Microsoft Hohm was a web-based application, intended to work by taking data from a smart meter installed, enabling customers to analyze their usage, and offers energy saving recommendations, helping them save energy and money. It has been discontinued in 2012 for commercial reasons.

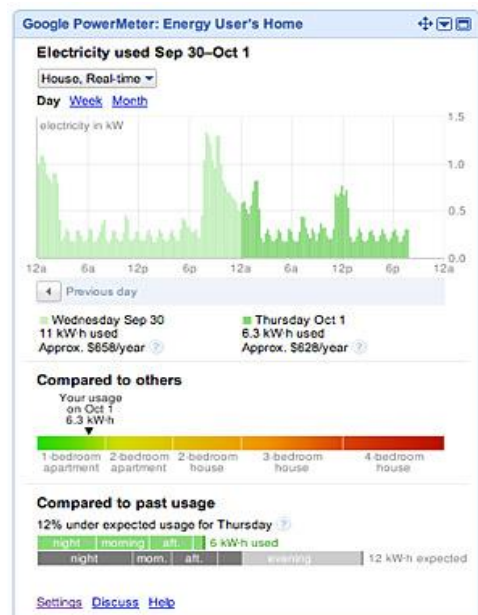


Figure 28 : Google PowerMeter

Google PowerMeter was a software project that allows users to view energy consumption in day, week, or month, and provides some predictive analysis. The user can share his data on iGooglehomepage with his contacts. It was hoped to raise awareness about energy and climate problems. Unfortunately, the project ended in 2011.



# *Appendix*

**In this chapter you will find the code of our design**

## Code:

### Globals

```
varGlobals={  
    app_user_name:",          // These variables are global variables which may  
    device_id:",              be accessed during any function through  
};                             the code
```

### Sign Up

```
$(document).ready(function(){ // When the page is ready do this {..... }  
    $("#signup").click(function(){//When I click do this {.....}  
varServiceURL =  
'http://41.43.67.72:8080/WattSee/api/clients/register/'+$('#firstname').val()+ '/' + $('#lastname').  
val()+ '/' + $('#address').val()+ '/' + $('#email').val()+ '/' + $('#password').val()+ '/';  
console.log(ServiceURL);    // This is the URL of the register web service contains  
    $.ajax({the IP and the needed data for this operation like:  
url: ServiceURL,first and last name, address, email and password  
type: 'GET',  
data: "",  
contentType: "application/json",  
success: function (response, textStatus, xhr) {  
console.log(response, "ajax Response");  
varobj = jQuery.parseJSON(response);// Whatever the returned response is, show  
if (obj.id == 0) {it in json  
alert(obj.message);  
    }// Display alert message attached with  
    },with the web service
```

```
error: function (xhr, status, error) {  
  console.log(error, "error");  
  }  
  })  
});
```

```
var ALERT_TITLE = "OK!"; // A template to post Alert
```

```
var ALERT_BUTTON_TEXT = "Ok";
```

```
if(document.getElementById) {  
  window.alert = function(txt) {  
    createCustomAlert(txt);  
  }  
}
```

```
function createCustomAlert(txt) {
```

```
  d = document;
```

```
  if(d.getElementById("modalContainer")) return;
```

```
  mObj = d.getElementsByTagName("body")[0].appendChild(d.createElement("div"));
```

```
    mObj.id = "modalContainer";
```

```
  mObj.style.height = d.documentElement.scrollHeight + "px";
```

```
  alertObj = mObj.appendChild(d.createElement("div"));
```

```
    alertObj.id = "alertBox";
```

```

if(d.all&& !window.opera) alertObj.style.top = document.documentElement.scrollTop + "px";
alertObj.style.left = (d.documentElement.scrollWidth - alertObj.offsetWidth)/2 + "px";
alertObj.style.visibility="visible";

    h1 = alertObj.appendChild(d.createElement("h1"));
h1.appendChild(d.createTextNode(ALERT_TITLE));

msg = alertObj.appendChild(d.createElement("p"));
    //msg.appendChild(d.createTextNode(txt));
msg.innerHTML = txt;

btn = alertObj.appendChild(d.createElement("a"));
    btn.id = "closeBtn";
btn.appendChild(d.createTextNode(ALERT_BUTTON_TEXT));
btn.href = "#";
btn.focus();
btn.onclick = function() { removeCustomAlert();return false; }

alertObj.style.display = "block";
}
functionremoveCustomAlert() {
document.getElementsByTagName("body")[0].removeChild(document.getElementById("modalCon
tainer"));
}
});// End of template

```

## Log in

```
$(document).ready(function() { // Ready function

    $("#midosa").click(function() { // On click function

        console.log(Globals.app_user_name);

        var ServiceURL =
        'http://41.43.81.185:8080/WattSee/api/clients/login/'+$('#Email').val()+'/'+$('#Pass').val()+'/'

        console.log(ServiceURL); // URL of login web service

        $.ajax({

            url: ServiceURL,

            type: 'GET',

            data: "",

            contentType: "application/json",

            success: function (response, textStatus, xhr) { // Web service is loaded successfully

                console.log(response, "ajax Response");

                var obj = jQuery.parseJSON(response);

                if (obj.id == 0) {

                    alert(obj.message);

                } else {

                    window.location.href = "charts.html";

                    // Globals.app_user_name=$('#Email').val();

                    localStorage.setItem("username", $('#Email').val());

                }

            },

            error: function (xhr, status, error) {
```

```
console.log(error, "error");
    }
  })
});
```

```
var ALERT_TITLE = "Oops!"; // Alert template
```

```
var ALERT_BUTTON_TEXT = "Ok";
```

```
if(document.getElementById) {
  window.alert = function(txt) {
    createCustomAlert(txt);
  }
}
```

```
functioncreateCustomAlert(txt) {
```

```
  d = document;
```

```
  if(d.getElementById("modalContainer")) return;
```

```
  mObj = d.getElementsByTagName("body")[0].appendChild(d.createElement("div"));
```

```
    mObj.id = "modalContainer";
```

```
  mObj.style.height = d.documentElement.scrollHeight + "px";
```

```
  alertObj = mObj.appendChild(d.createElement("div"));
```

```
    alertObj.id = "alertBox";
```

```
  if(d.all&& !window.opera) alertObj.style.top = document.documentElement.scrollTop + "px";
```

```

alertObj.style.left = (d.documentElement.scrollWidth - alertObj.offsetWidth)/2 + "px";
alertObj.style.visibility="visible";

    h1 = alertObj.appendChild(d.createElement("h1"));
h1.appendChild(d.createTextNode(ALERT_TITLE));

msg = alertObj.appendChild(d.createElement("p"));
    //msg.appendChild(d.createTextNode(txt));
msg.innerHTML = txt;

btn = alertObj.appendChild(d.createElement("a"));
    btn.id = "closeBtn";
btn.appendChild(d.createTextNode(ALERT_BUTTON_TEXT));
btn.href = "#";
btn.focus();
btn.onclick = function() { removeCustomAlert();return false; }

alertObj.style.display = "block";

}

functionremoveCustomAlert() {
document.getElementsByTagName("body")[0].removeChild(document.getElementById("modalCon
tainer"));
}

});// End of template

```

## Charts

```
/******
```

Simple Bar Chart

```
***** /
```

```
$(document).ready(function(){ // Load this function on page startup
```

```
var serial="2f6b7f572ed646a9ad7f5d7faeeafbef";
```

```
var counter=0
```

```
varServiceURL = 'http://41.43.81.185:8080/WattSee/api/records/daily-records/'+serial+'/'+counter+'/';
```

```
$.ajax({// Details of the daily records
```

```
url: ServiceURL,
```

```
type: 'GET',
```

```
data: "",
```

```
contentType: "application/json",
```

```
success: function (response, textStatus, xhr) {
```

```
console.log(response, "ajax Response");
```

```
varobj = jQuery.parseJSON(response);
```

```
if (obj.id == 0) {
```

```
alert(obj.message);
```

```
}else{
```

```
$.("#line-chart").dxChart({// Template of daily chart
```

```
dataSource: [Representation
```

```
{ hour: "0:0", europe: obj.records[0] },
```

```
{ hour: "1:0", europe: obj.records[1] },
```

```
{ hour: "2:0", europe: obj.records[2] },
```



```
{ hour: "3:0", europe: obj.records[3] },
{ hour: "4:0", europe: obj.records[4] },
{ hour: "5:0", europe: obj.records[5] },
{ hour: "6:0", europe: obj.records[6]},
{ hour: "7:0", europe: obj.records[7] },
{ hour: "8:0", europe: obj.records[8]},
{ hour: "9:0", europe: obj.records[9]},
{ hour: "10:0", europe: obj.records[10]},
{ hour: "11:0", europe: obj.records[11]},
{ hour: "12:0", europe: obj.records[12]},
{ hour: "13:0", europe: obj.records[13]},
{ hour: "14:0", europe: obj.records[14]},
{ hour: "15:0", europe: obj.records[15]},
{ hour: "16:0", europe: obj.records[16]},
{ hour: "17:0", europe: obj.records[18]},
{ hour: "18:0", europe: obj.records[19]},
{ hour: "19:0", europe: obj.records[20]},
{ hour: "20:0", europe: obj.records[21]},
{ hour: "21:0", europe: obj.records[22]},
{ hour: "22:0", europe: obj.records[23]},
{ hour: "23:0", europe: obj.records[24]}
],
commonSeriesSettings: {
argumentField: "hour"
},
series: [
```

```

{ valueField: "europe", name: "Europe", color: "#27c24c" },
{ valueField: "americas", name: "Americas", color: "#fad733" },
{ valueField: "africa", name: "Africa", color: "#f05050" }
  ],
  tooltip: {
    enabled: true,
    font: { size: 16 }
  },
  legend: {
    visible: false
  },
  valueAxis: {
    grid: {
      color: '#9D9EA5',
      width: 0.1
    }
  }
}
})// End of daily chart template

```

```

  }
},
error: function (xhr, status, error) {
  console.log(error, "error");
}
})

```

```
$("#bar-chart").dxChart({// Monthly template
```

```
dataSource: [
```

```
    {month: "Jan", oranges: 3},
```

```
    {month: "Feb", oranges: 2},
```

```
    {month: "Wed", oranges: 3},
```

```
    {month: "Mar", oranges: 4},
```

```
    {month: "Apr", oranges: 6},
```

```
    {month: "May", oranges: 11},
```

```
    {month: "Jun", oranges: 4},
```

```
    {month: "Jul", oranges: 4},
```

```
    {month: "Aug", oranges: 4},
```

```
    {month: "Sep", oranges: 4},
```

```
    {month: "Oct", oranges: 4},
```

```
    {month: "Nov", oranges: 4},
```

```
    {month: "Dec", oranges: 4} ],
```

```
valueAxis:{
```

```
grid:{
```

```
color: '#000',
```

```
width: 0.1
```

```
    }
```

```
},
```

```
legend: {
```

```
visible: false,
```

```
},
```

```

series: {
  argumentField: "day",
  valueField: "oranges",
  name: "My oranges",
  visible: false,
  type: "bar",
  color: '#000'
}
}); // End of monthly chart template representation

```

```

$("#bar-chart-1").dxChart({
  dataSource: [// Weekly template

```

```

    {day: "Sat", oranges: 11},
    {day: "Sun", oranges: 4},
    {day: "Mon", oranges: 3},
    {day: "Tue", oranges: 2},
    {day: "Wed", oranges: 3},
    {day: "Thu", oranges: 4},
    {day: "Fri", oranges: 6}],

```

```

  valueAxis: {

```

```

    grid: {

```

```

      color: '#9D9EA5',

```

```

      width: 0.1

```

```

    }

```

```

  },

```

```

  legend: {

```

```

    visible: false,

```

```
    },  
    series: {  
      argumentField: "day",  
      valueField: "oranges",  
      name: "My oranges",  
      visible: false,  
      type: "bar",  
      color: '#a49bc4'  
    }  
  });  
  
  // End of template
```

```
$("#bar-chart-2").dxChart({  
  dataSource: [  
    {day: "01", oranges: 3},  
    {day: "02", oranges: 2},  
    {day: "03", oranges: 3},  
    {day: "04", oranges: 4},  
    {day: "05", oranges: 6},  
    {day: "06", oranges: 11},  
    {day: "07", oranges: 4},  
    {day: "08", oranges: 11},  
    {day: "09", oranges: 11},  
    {day: "10", oranges: 11},  
    {day: "11", oranges: 11},
```

```
{day: "12", oranges: 11},
{day: "13", oranges: 11},
{day: "14", oranges: 11},
{day: "15", oranges: 11},
{day: "16", oranges: 11},
{day: "17", oranges: 11},
{day: "18", oranges: 11},
{day: "19", oranges: 11},
{day: "20", oranges: 11},
{day: "21", oranges: 11},
{day: "22", oranges: 11},
{day: "23", oranges: 11},
{day: "24", oranges: 11},
{day: "25", oranges: 11},
{day: "26", oranges: 11},
{day: "27", oranges: 11},
{day: "28", oranges: 11},
{day: "29", oranges: 11},
{day: "30", oranges: 11},
{day: "31", oranges: 11} ],
```

```
valueAxis://{ monthly chart
```

```
grid:{
```

```
color: '#9D9EA5',
```

```
width: 0.1
```

```
 }
```

```
},
```

```
legend: {
  visible: false,
  },
series: {
  argumentField: "day",
  valueField: "oranges",
  name: "My oranges",
  visible: false,
  type: "bar",
  color: '#a49bc4'
  }
});
```

```
/*****
```

Line Chart

```
*****/
```

```
});
```

## Devices

```
$(document).ready(function(){  
  
  // Save username when got,  
  
  console.log(localStorage.getItem("username")); to be used when needed  
  
  varServiceURL = ' http://41.43.81.185:8080/WattSee/api/devices/client-  
  devices/'+localStorage.getItem("username")+'/';  
  
  $.ajax({  
  
    url: ServiceURL,  
  
    type: 'GET',  
  
    data: "",  
  
    contentType: "application/json",  
  
    success: function (response, textStatus, xhr) {  
  
      console.log(response, "ajax Response");  
  
      varobj = jQuery.parseJSON(response); // Post outputs in JSON format  
  
      if (obj.id == 0) { when received from web service  
  
        alert(obj.message);  
  
      }else{  
  
        $('#Tariff').val(obj[0].tariff.name);  
  
        $('#Name').val(obj[0].name);  
  
        $('#Description').val(obj[0].description);  
  
        $('#Serial').val(obj[0].serial);  
  
        $('#API').val(obj[0].api);  
  
      }  
  
    }  
  
  });  
  
});
```



```

    },
    error: function (xhr, status, error) {
    console.log(error, "error");
    }
    })

//-----//

$("#Edit_Device").click(function(){
console.log(localStorage.getItem("username"));

varServiceURL =
'http://41.43.81.185:8080/WattSee/api/devices/update/'+device_id+'/'+localStorage.getItem("us
ername")+ '/' +$('#Tariff').val()+ '/' +$('#Name').val()+ '/' +$('#Description').val()+ '/' +$('#Address').v
al()+ '/' +$('#Serial').val()+ '/' +$('#API').val()+ '/';

// Attached data needed to be sent with the URL of the web service

$.ajax({
url: ServiceURL,
type: 'GET',
data: "",
contentType: "application/json",
success: function (response, textStatus, xhr) {
console.log(response, "ajax Response");
varobj = jQuery.parseJSON(response);
if (obj.id == 0) {// Alert template
alert(obj.message);
}else{
alert(obj.message);
}
}

```

```

    },
    error: function (xhr, status, error) {
    console.log(error, "error");
    }
    })
    });
//-----//
var ALERT_TITLE = "Oops!";
var ALERT_BUTTON_TEXT = "Ok";

if(document.getElementById) {
window.alert = function(txt) {
createCustomAlert(txt);
}
}

functioncreateCustomAlert(txt) {
    d = document;

if(d.getElementById("modalContainer")) return;

mObj = d.getElementsByTagName("body")[0].appendChild(d.createElement("div"));
    mObj.id = "modalContainer";
mObj.style.height = d.documentElement.scrollHeight + "px";

alertObj = mObj.appendChild(d.createElement("div"));

```

```

    alertObj.id = "alertBox";
if(d.all&& !window.opera) alertObj.style.top = document.documentElement.scrollTop + "px";
alertObj.style.left = (d.documentElement.scrollWidth - alertObj.offsetWidth)/2 + "px";
alertObj.style.visibility="visible";

    h1 = alertObj.appendChild(d.createElement("h1"));
h1.appendChild(d.createTextNode(ALERT_TITLE));

msg = alertObj.appendChild(d.createElement("p"));
    //msg.appendChild(d.createTextNode(txt));
msg.innerHTML = txt;
btn = alertObj.appendChild(d.createElement("a"));
    btn.id = "closeBtn";
btn.appendChild(d.createTextNode(ALERT_BUTTON_TEXT));
btn.href = "#";
btn.focus();
btn.onclick = function() { removeCustomAlert();return false; }

alertObj.style.display = "block";
}
functionremoveCustomAlert() {
document.getElementsByTagName("body")[0].removeChild(document.getElementById("modalCon
tainer"));
}
});// End of alert template

```

## Tarrifs

```
$(document).ready(function(){

console.log(localStorage.getItem("username"));

varServiceURL = ' http://41.43.81.185:8080/WattSee/api/tiers/get-all/';

// URL of the get-all web service

$.ajax({

url: ServiceURL,

type: 'GET',

data: "",

contentType: "application/json",

success: function (response, textStatus, xhr) {

varobj = jQuery.parseJSON(response);

if (obj.id == 0) {

alert(obj.message);

}else{

varHTMLFooter;

var Temp = 7;

for (var i = 0; i < Temp; i++) {

console.log(i);// Design of prices table

HTMLFooter += ' <tr role="row" class="odd">' +

'<td class="sorting_1">'+obj[i].name+'</td>' +

'<td>'+obj[i].from+'</td>' +

'<td>'+obj[i].to+'</td>'+

'<td>'+obj[i].price+'</td><td>'+obj[i].taxes+'</td><td>'+obj[i].taxes+'</td></tr>';
```

```

    }

    $("#tr_data").append(HTMLFooter);

var HTMLFooter1;
var Temp1 = 12;
for (var i = 8; i < Temp1; i++) {

    HTMLFooter1 += ' <tr role="row" class="odd">' +
        '<td class="sorting_1">'+obj[i].name+'</td>' +
        '<td>'+obj[i].from+'</td>' +
        '<td>'+obj[i].to+'</td>'+
        '<td>'+obj[i].price+'</td><td>'+obj[i].taxes+'</td><td>'+obj[i].taxes+'</td></tr>';
    }

    $("#com_data").append(HTMLFooter1);

var HTMLFooter2;// Second table
var Temp2 = 16;
for (var i = 12; i < Temp2; i++) {

    HTMLFooter2 += ' <tr role="row" class="odd">' +
        '<td class="sorting_1">'+obj[i].name+'</td>' +
        '<td>'+obj[i].from+'</td>' +
        '<td>'+obj[i].to+'</td>'+
        '<td>'+obj[i].price+'</td><td>'+obj[i].taxes+'</td><td>'+obj[i].taxes+'</td></tr>';
    }

    $("#ind_data").append(HTMLFooter2);
}

```

```

        },// Alert of error template when we fail to call
error: function (xhr, status, error) {    the web service
console.log(error, "error");
    }
})

//-----//

var ALERT_TITLE = "Oops!";
var ALERT_BUTTON_TEXT = "Ok";

if(document.getElementById) {
window.alert = function(txt) {
createCustomAlert(txt);
    }
}

functioncreateCustomAlert(txt) {
    d = document;

if(d.getElementById("modalContainer")) return;

mObj = d.getElementsByTagName("body")[0].appendChild(d.createElement("div"));
    mObj.id = "modalContainer";
mObj.style.height = d.documentElement.scrollHeight + "px";

```

```

alertObj = mObj.appendChild(d.createElement("div"));
    alertObj.id = "alertBox";
if(d.all&& !window.opera) alertObj.style.top = document.documentElement.scrollTop + "px";
alertObj.style.left = (d.documentElement.scrollWidth - alertObj.offsetWidth)/2 + "px";
alertObj.style.visibility="visible";

    h1 = alertObj.appendChild(d.createElement("h1"));
h1.appendChild(d.createTextNode(ALERT_TITLE));

msg = alertObj.appendChild(d.createElement("p"));
    //msg.appendChild(d.createTextNode(txt));
msg.innerHTML = txt;

btn = alertObj.appendChild(d.createElement("a"));
    btn.id = "closeBtn";
btn.appendChild(d.createTextNode(ALERT_BUTTON_TEXT));
btn.href = "#";
btn.focus();
btn.onclick = function() { removeCustomAlert();return false; }

alertObj.style.display = "block";

}

functionremoveCustomAlert() {

```

```
document.getElementsByTagName("body")[0].removeChild(document.getElementById("modalCon  
tainer"));  
  
}  
  
});// End of template
```



## REFERENCES

### 1- Motivation

- I- [www.misr5.com](http://www.misr5.com)
- II- [www.moee.gov.eg](http://www.moee.gov.eg)
- III- [www.masralarabia.com](http://www.masralarabia.com)

### 2- Used Technologies Background

- I- [www.w3schools.com](http://www.w3schools.com)
- II- [recombu.com/mobile/article/what-is-android-and-what-is-an-android-phone\\_M12615.html](http://recombu.com/mobile/article/what-is-android-and-what-is-an-android-phone_M12615.html)
- III- [ipad.about.com/od/iPad-Glossary/g/What-Is-iOS.htm](http://ipad.about.com/od/iPad-Glossary/g/What-Is-iOS.htm)
- IV- [www.pcmag.com/encyclopedia/term/61785/windows-phone](http://www.pcmag.com/encyclopedia/term/61785/windows-phone)
- V- [Apache Cordova 3 Programming - John M. Wargo, Addison-Wesley Professional, 2012](#)
- VI- [PhoneGap 3 Beginner's Guide - Giorgio Natili, Packt Publishing, 2013](#)
- VII- [PhoneGap Essentials: Building Cross-Platform Mobile Apps - John M. Wargo, Addison-Wesley Professional, 2012](#)
- VIII- [Beginning PhoneGap - Thomas Myer, Wrox, 2011](#)

### 3- Responsive web design

- I- [smallbiztrends.com/2013/05/what-is-responsive-web-design.html](http://smallbiztrends.com/2013/05/what-is-responsive-web-design.html)

### 4- Bootstrap

- I- [Bootstrap 3.3: A Guide Book for Beginners to Learn Web Framework Fast! - Kevin Lyn, Amazon Digital Services, Inc., 2015](#)
- II- [Bootstrap: The ultimate beginners guide to Bootstrap 3.0 - Mike Stevenson, Amazon Digital Services, Inc., 2014](#)
- III- [Jump Start Bootstrap - Syed Fazle Rahman, SitePoint, 2014](#)

## 5- Related works and previous attempts

- I- [www.ibm.com/developerworks/downloads/ws/worklight/](http://www.ibm.com/developerworks/downloads/ws/worklight/)
- II- [www.appcelerator.com/](http://www.appcelerator.com/)
- III- [www.qt.com.io/](http://www.qt.com.io/)
- IV- [www.xamarin.com/](http://www.xamarin.com/)