Honours Project Report

E-Government

Hilja Angula

Supervised by: Sonia Berman

<table>
<thead>
<tr>
<th>Category</th>
<th>Min</th>
<th>Max</th>
<th>Chosen</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Software Engineering/System Analysis</td>
<td>0</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>2 Theoretical Analysis</td>
<td>0</td>
<td>25</td>
<td>?</td>
</tr>
<tr>
<td>3 Experiment Design and Execution</td>
<td>0</td>
<td>20</td>
<td>?</td>
</tr>
<tr>
<td>4 System Development and Implementation</td>
<td>0</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>5 Results, Findings and Conclusion</td>
<td>10</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>6 Aim Formulation and Background Work</td>
<td>10</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>7 Quality of Report Writing and Presentation</td>
<td>10</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>8 Adherence to Project Proposal and Quality of Deliverables</td>
<td>10</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>9 Overall General Project Evaluation</td>
<td>0</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Total marks</td>
<td>80</td>
<td>80</td>
<td></td>
</tr>
</tbody>
</table>
Electricity usage is very important because it is very expensive and it is one of the most used resources. Many people are not aware that consuming too much electricity has an effect on global warming; because in producing electricity, carbon emission is released. This project looks at making people aware of their electricity usage and giving them consumption information, with the hope that they will decrease their electricity consumption. The City of Cape Town wants to implement some kind of system that would let people know about their electricity usage and what impact that has on carbon emission and provide some option to the users so that they see what will happen if they were to make certain lifestyle changes. Less water consumption is just as important, the world experience water shortage, drought and low rainfall in recent years [20]. Carbon dioxide emission also occurs where water is heated with generators to purify or kill germs. This project therefore looks at building such a prototype that the City of Cape Town can use in order to decrease electricity and water consumption and carbon emission.

This report focuses on the monitoring system and the backend functionality of the project, the other two components not dealt with in this report are the Mxit and the web interfaces. The data from the City of Cape Town with water, sewerage and electricity bills of sample customers was loaded in a database and methods created to provide this in various forms to the Mxit and the web subsystem. The backend system provides functionality such as predicting electricity/water/sewerage consumption based on past history and on changed habits of a user, estimating what past electricity/water history would have been if certain lifestyle changes had been done, calculating Rand and carbon emission savings from changed usage in appliances, etc. I implemented a monitoring system to track usage of the system via Mxit and the web interface. Every call made to my subsystem via Mxit or the web is automatically tracked into the monitoring system, user’s individual usage via Mxit and the web is also tracked. The monitoring system is used internally by the City to gauge the success of the project; it displays graphs indicating usage of the system over time and consumption changes effected by its users.
# CONTENTS

1. Introduction ........................................................................................................................................4

2. Background chapter ..........................................................................................................................6
   2.1 What is e-government? ..................................................................................................................6
   2.2 Examples.......................................................................................................................................6
   2.3 Challenges.....................................................................................................................................7
   2.4 Research into appliance consumption .........................................................................................9

3. Analysis and design ..........................................................................................................................9
   3.1 Use case overview .......................................................................................................................10
   3.2 Design ..........................................................................................................................................12
   3.3 System requirement specification .................................................................................................14
   3.4 Design decisions .........................................................................................................................14
   3.5 Functional requirements ..............................................................................................................15
   3.6 The interface design ....................................................................................................................18

4. Implementation ..................................................................................................................................30
   4.1 Methods implemented ..................................................................................................................30
   4.2 Loading data into the database ....................................................................................................38
   4.3 Monitoring system usage via the web and Mxit interface ............................................................39

5. Testing ...............................................................................................................................................42

6. Conclusion .........................................................................................................................................44

References: ............................................................................................................................................45
1. INTRODUCTION

Electricity is quite a necessity for daily life, so almost everyone that has access to it uses it. However it is not free. Electricity users, pay monthly bills, or use prepaid electricity whereby they first buy the electricity before they use it, as oppose to monthly bill where the user only pays at the end of the month. Those that pay for electricity at the end of the month are in most cases not aware just how much they have consumed electricity until they see the bill at the end of the month. The monthly bill comes as a shock to people who have consumed too much electricity. For prepaid users they however have some indicator of their usage because they can use as much electricity as they can afford and see how quickly their electricity units that they bought runs out. However they don’t know how appliances or usage habits have affected consumption.

The aim of this project is therefore to offer online access to electricity bills, water bills and sewerage, and usage history. On top of that users can make queries about changes such as switching off the geyser or replacing non saver light bulbs with saver bulbs to see what their electricity history would have been if such changes were made. With water they can also play around with making some changes such as reducing the time they take to shower to see what effect that would have had in their history. While the aim of this project is to try and make people aware of their water, electricity, sewerage consumption, it is also in a way trying to reduce the impact of misusing those resources on earth. One of the impact or effect of using too much electricity is that is that it releases too much carbon dioxide. Scientists state that too much carbon dioxide in the atmosphere plays a major role in climate change, it causes the temperature to rise, and therefore it is very important to reduce Carbon dioxide emissions as it can have serious effect on the future [11]. It has also been stated that “on average, electricity sources emit 1.297 lbs CO2 per kWh (0.0005883 metric tons CO2 per kWh)” [12].

This project consists of three components, one deal with checking of electricity, water and sewerage bill via the web, this is dealt with by Ashiq and it is discussed in details in his report. The other part was providing similar services to people via Mxit, this is dealt with by Morithla and it is discussed in his report. The third part deals with the back end of the system, and this is the part that I dealt with. While the focus of the other two (Ashiq and Morithla) was more on the end users of the system, mine was more on the client which was the City of Cape Town. My focus was more on how the client wanted the system to work, handling and making the entire
database queries send from the Web or from Mxit. I was the one in control of the database, and I also had to come up with some functions that other two subsystems wanted. So I received data from the City of Cape Town and then created a database and came up with formulas for instance to check how much kilowatts a user would save if for instance they switched off the geyser. I would then send the result to the Mxit or the web interface components and they would then display this to the end user. So there was flow of information between me and each of the other two. I also implemented a monitoring system for the City of Cape Town to track the usage of the system once it was in place.
2. BACKGROUND CHAPTER

I will start off by describing what e-government is and how this project fits in with e-government. I will then look at some of the e-government aspect (the need for good response from the government, easy internet access) that many governments do not handle efficiently and discuss how our project tackles response from the government. I will also give an example of some e-government projects that have been done already, summarize challenges of an E-government and finish off with research into appliance consumption.

2.1 What is e-government?

Internet is overtaking many government sectors at a high speed now that the number of people with internet access is increasing. It is therefore possible to have online service delivery instead of a lot of paper work that people had to put up with. E-government refers to the use of Information technologies (ICTs) by the government [3]. Responsiveness from the government is however very important to boost public confidence in e-government services [2], so our project which is dealing with providing services of checking electricity bills, water and sewerage is also classified as E-government because it will be delivering this service to people on line instead of waiting to be mailed the bill at the end of the month. One of the disadvantages of mailing bills to people is that people travel a lot and it is possible that they might not be in the area when the mail is sent to them. The project also has tools to show effect of better habits which can be applied to personal data. Such a facility is not possible without ICT. The city hopes such tools will promote awareness and encourage wiser consumption. We therefore hope that our project would improve and strengthen the communication between the City of Cape Town and its residents to enable quick and easy delivery of the governmental information and services to people.

2.2 Examples

There are many e-government projects which use Information technologies to deliver information and services to the public; this transforms access of information to the public [3]. One is the Electronic Polling System for Remote Voting Operation (ePoll) which aims at implementing polling system remotely to simplify the voting process through the use of ICTs which will be more accessible to everyone importantly to disabled people and blind people [5]. Countries such as South Korea and South Africa have replaced their manual tax filing forms with electronic tax filing forms. Individual and corporations in South Korea are both capable of using the internet and this has eased the government in transforming its major civil affairs administration functions into an e-government system [8].
2.3 Challenges

An article “How responsive is E-Government? Evidence from Australia and New Zealand” by Gauld R, Gray A and McComb S reports governments’ response based on the basic capacity to answer a simple question posted by email [2]. An email was sent to list of relevant agencies in Australia and New Zealand, at the same time and out of office hours. Results showed that Australia did not respond much to their email compared to New Zealand [2]. Non response or slowness to respond or failure to answer the questions correctly even though it’s a simple scenario, it has the ability to lower public confidence in e-government [2]. Our project would tackle the aspect of responsiveness through self service on line, the user would not have to wait for any one from the City of Cape Town to confirm anything when they are online, all they would need to provide would be their account number and result would be shown to them. This is very important because it would save users time and increase the service of the City of Cape Town. E-government cannot be successful if the public does not benefit from it or make use of it, so it is very important to look at some of the aspects that will get the public involved when moving towards e-government [3]. Our project tackles the aspect of ensuring that most people can have access to the system by providing two user interfaces to the system via Mxit and the Web. Some people might have internet access on their computers and some people might not have computers but cell phones only. So people can still view the system via MXit on their phones, this has two advantages: MXit is one of the easily accessible forms of communication available on cell phones, and MXt is cheap [13] for everyone to use.

Some government tends to focus a lot on the technical aspects and little on the people [9]. Research done in china showed that very little is done to increase awareness of e-government to people [9]. Very few people know of the existence of e-government services and most do not have internet access [9]. Most of the people that have internet access in developing countries like China are the educated people: old people and poor people in rural area do not benefit from e-government services as they either do not have internet access because it is very expensive and sometimes due to lack of awareness and education [9]. Research suggests more awareness and training on how to use e-government services should be enforced. Computers and internet also need to be supplied freely or at a lower cost to enable poorer people to make use of e-government services [9]. Our project having two interfaces Mxit and the web, we hope to increase its accessibility, and for poor people who might not have computers, cell phones are widely used and many people have cell phones therefore they too can access the system via MXit. E-government can simplify life for people therefore their services needs to be made readily available. For an E-Government to be
successful, access to internet and ICT skills of the citizens are of importance as these have an impact on citizen's adoption of e-government services [2]. The success of e-government depends on how much the government involves citizens into using online public services [6]. Research indicates that e-government have failed to provoke the general public to accept the e-services offered by agencies in most countries, this implies that extensive efforts are necessary to increase citizen's awareness of on line service delivery. In order to prevent digital divide in terms of who has access to government service, it is necessary for all citizens from all segments of society to be equipped with basic ICT skills and high speed internet connections [6]. This needs to be resolved because they are critical challenges to citizens' adoption of the emerging e-government services [6]. E-Government services serve no purpose if the public has no access to using them. Digital divide refers to the gap between people with access to digital and information technology and those with limited or no access at all [7]. Another important option is to use m-government. M-government is a strategy to improve benefits to citizens, business and all government units involved in e-government, by implementation involving the utilization of mobile devices [23].

One of the countries that have been reported to show unequal access to e-government services is China. Research done in Chinese e-government shows that implementation focus more on the technical aspects, there is very little research on the users [9]. There have been reports on some of the challenges; these were lack of awareness of e-government, little or no education, and income gap [9]. Suggestions on the need for alternative strategies to enhance awareness of e-government services in China have been made [9]. Internet has a major role in the use of e-government by the public so more focus on internet access needs to be considered, internet is also expensive in developing countries which makes it difficult for poor people to make use of e-government services [9].

Governments around the world are moving towards online service delivery with high hopes that government services will be more responsive [2]. Some sort of training need to be incorporated into e-government services to allow for uneducated people to be able to use e-government services [9]. E-government can have lots of benefits if consideration into how people would use the services is considered. This would imply making e-government systems easy to use and to offer services that are needed by the citizens. One of the very important aspects to consider is the digital divide [6], while many governments around the world are moving towards E-Governments, very little is done to equip people with ICT skills and high speed fast internet connections [6]. It is very important to have a good response from the services offered, else it may further lower down public confidence in government services. Some countries have adopted e-government initiatives and have been very successful for corporations and individuals [8].
2.4 Research into appliance consumption

Researching how much each electrical appliance consumes electricity is useful; it quantifies total kilowatts of electricity consumed by appliances in a house or apartment [24]. This gives an indication of how much proportion of kilowatts can be reduced by decreasing usage of appliances that consume too much kilowatts. A study into different appliances of houses, can characterize different houses into categories according to number of appliances of the household. Houses that turn to have many electricity appliances and water appliances will tend to use high electricity and water compared to houses with fewer appliances. How much carbon emission will result from using an appliance can be deduced from how much electricity the appliances consumes, per kwh of electricity consumed by an appliance 1.297 lbs (0.0005883 metric tons) of Carbon dioxide is emitted [12].

3. ANALYSIS AND DESIGN

This project aims to inform citizens about their water and electricity consumption and how lifestyle changes can reduce this. It provides two interfaces to data on water and electricity consumption- one via MXit and the other on the Web, as well as a monitoring tool whereby the city can track system usage.
3.1 Use case overview

![Diagram](image)

**Description of the use case above**

As shown above there will be external users to use the system either via Mxit or via the website, this is represented by external user. External users are the end users who will be using the system via Mxit or via the website. Every query that users make to Mxit/the website invoke my functions that connect to the database and do some calculation and return the required result. Each of my function has inner functions that, store, update every call made; this information is also stored on the database. Internal user is the person/people at the City of Cape Town who will be using the monitoring system to monitor the system usage. When an internal make a query to view tracked information, this invokes a call to the database that returns the required information, draw the requested graph and display it to the user.
This section gives an insight in how the whole project links together.

Figure 1. The whole system overview

Figure 1 above shows my section dealt with all database connections and tracking/recording all calls made via the Web/Mxit so that the City of Cape Town can monitor how the system is doing or how many people logged in via Mxit or via the web. Users would supply their account numbers via Mxit or the Web. Mxit or the web would then call functions from my subsystem and supply the account number of the user. Functions from my subsystem use the ConnectorJDBC driver in order to access Mysql database and return the requested information of the supplied account number. Mxit or the website then display returned information to the users via Mxit or via the website respectively.

Every call that Mxit or the website makes are automatically tracked in the monitoring system, users that make use of the website or the website their details are also automatically tracked in the monitoring system. The monitoring system would only be used internally to track the usage of the system; end users would not have access to the monitoring system. The monitoring system would display graphs indicating usage via Mxit and via the website. These graphs would indicate how many people have used Mxit or the website over time, by counting all the number of log on that were made via Mxit and the website.
The monitoring system also has options to check which aspects of the system are used and where as an example, the number of bulb and geyser queries that were made via Mxit or via the website are counted and displayed on a bar graph. The monitoring system displays the bulb or the geyser queries made in the last few days or monthly in the last few months. The monitoring system would also track different suburb’s electricity net consumption and display a bar graph with positive or negative increase in the electricity consumption in kilowatts per suburb over time.

3.2 Design

Software engineering Model

I had two different clients, providing the functions needed by Mxit and for the website interface. They needed me to create functions that can retrieve required information from the database and to also to come up with some formulas for calculating carbon emissions from consumed electricity consumption. One of the formulas that I had to come up with to apply to data in the database was to calculate average electricity/water/sewerage of a specific user. I also had to analyse past electricity/water/sewerage history for each user in the database by looking at the rate of change and predict the next 6 months usage for each user. The other formulas were for example decreasing hours of using the geyser or time of showering and predicting future usage or what the past history usage would have been if those changes had already been in place. This process of collecting requirements was started by gathering requirements from Morithla for Mxit and Ashiq for the website; I sketched the functions on a piece of paper and showed it to them, and repeatedly made changes to it and then programmed a prototype and tested it with them. I gathered their feedback and any requirements that were not fully met to their satisfaction and made changes then continued the process of testing it and collecting feedback as shown in figure 2 on the following page.
The other component that I implemented is the monitoring system that the City of Cape Town can use (Mr. Andre Stelzner, CIO for the City of Cape Town). I started by looking at some of the obvious indicators that could show the usage of Mxit and similarly for the website. This included incrementing some counter every time someone logs on to Mxit or on the website. I drafted rough graphs as my first prototype indicating what would display on the monitoring system and the different options and sent the draft prototype to Andre. I went to meet with Andre and gathered feedback to the prototype I had sent him. During my meeting with him he was happy with most of the things that were on the prototype, there was only one thing that he mentioned I change in one of the graphs.

There was an option to view suburb’s savings which was a calculation of how much money a suburb has saved due to their decreased electricity consumption. However he was not satisfied with this because he said electricity tariffs changes a lot therefore he is rather interested in suburb’s electricity net consumption with positive increase or negative decrease. The other thing that he said I should add was that he would like to be able to enter an account number of a customer on the monitoring system and be shown how many times that specific customer used Mxit and similarly how many times he used the website. Other than that he was happy with all the other things in the prototype, therefore I made changes to the prototype that were pointed out, the figure 3 on the following page summarizes this.
3.3 System requirement specification

We received spreadsheet data from the City of Cape Town which I then loaded into a database that I created. The data had electricity, water bill and sewerage bill of customers, each customer had a unique account number that can be used to retrieve their bills from the database. The data therefore had a column for consumption indicating how much electricity, water and sewerage a customer had consumed as well as the bill date indicating when the bill was taken and a column named rand value which indicates how much money a customer was being charged for their consumption. The data also contains suburb names of where each of the customers stays, along with many other items not relevant to the project.

3.4 Design decisions

The project used a database, it was my responsibility to create and load data received from the City of Cape Town in to the database and perform queries and calculations. I used Mysql because it is the world’s most popular open source database because of its reliability, ease of use and fast performance [15]. Other reasons for choosing Mysql is that it is open source and was familiar with some of its commands. Research into Mysql states that it is fast because it “has a query cache that can boost up the speed” and is supported by more existing programs [22].
My source code relied on retrieving data stored in a database, a database access was therefore of handy in order to interact with the stored data [21]. For this I learned about JDBC (Java DataBase Connectivity), which is a popular data access standard and which consists of a set of Java classes [14]. Developers (including myself) that develop applications that need to access Databases use ConnectorJDBC driver in order to access Mysql database. The JDBC was my preference since I had a data access application written in java that needed to connect to Mysql. By just learning the JDBC one can develop many data access application that can access different RDBMS (Relational Database Management System) without having to learn and use API for different RDMS vendors [14]. The other new thing I had to learn was how to draw graphs and for this I had to learn JFree chart which is a chart library for java platform. I chose JFree chart because it is free and because it is a java platform and because I was programming in java. Other reasons for choosing JFree chart is because it is easy to use and I exchanged ideas with the person in charge of the web interface as he was also using JFree chart.

3.5 Functional requirements

Below are the back end functionalities that I implemented to be used by Mxit and the website.

- Return latest month electricity, sewerage and water bill of a user from the database using the customer's account number.
- Return the history of past months electricity, water and sewerage of a given account number from the database.
- Calculate and return past months average electricity, water and sewerage consumption of a given account number.
- Return what the electricity and water history of a given account number would have been provided a user chose to make some changes e.g. decreasing geyser hour usage or reducing light bulb or shower usage to see how the history would have been.
- Calculate and return predicted future consumption of the next 6 months of what electricity, water and sewerage could be of a given account number by looking at the rate of change of the past months history (and also calculate the average of the predicted consumption).
• Calculate and return predicted future electricity consumption of a given account number as stated above but with an additional feature of what it would be provided a user choose to make some changes e.g. decreasing geyser hour usage or reducing light bulb or shower usage to see what the consumption would have been.

• For all functions that return electricity values, calculate and return the respective carbon emissions from the kilowatts of electricity consumed or predicted.

• Calculate and return how much rand would be saved from decreasing e.g. geyser hours by some constant.

• Have different functions that calculate and return for several appliances given below how much money a customer would save or lose if they decrease or increase the hour of usage: a fan, non saver bulb, saver bulb, heater, washing machine, kettle, microwave and geyser.

• Have the same functions as above but calculate and return carbon dioxide emission.

• A function for calculating Rand saved if replace non saver bulb with saver bulbs.

• A function for calculating carbon dioxide emission saved if replace non saver bulb with saver bulbs

• A function for calculating how much water would be saved or lost if a user decides to reduce or increase numbers of times they pipe wash a car per month.

• A function for calculating how much water would be saved or lost if a user decides to reduce or increase number of times they bath.

• A function for calculating how much water would be saved or lost if a user decides to reduce or increase the minutes of taking a shower per day.

• A function for creating a password, and authenticating, remembering password.

As already mentioned, every call automatically invokes methods that increment counters, this information is also saved in the database and I use this information to display graphs that indicate the usage of the system via Mxit, the web and overall system usage.
Below are the functionalities that I implemented for the monitoring system

- View a graph indicating the total number of people who used the system via Mxit per suburb.
- View a graph indicating the total number of people who used the system via the website per suburb.
- View a graph indicating the overall number of people who used the system regardless via Mxit or via the web over time.
- View a graph indicating the number of bulb queries made via the website over time.
- View a graph indicating the number of bulb queries made via Mxit over time.
- View a graph indicating the number of geyser queries made via the website over time.
- View a graph indicating the number of geyser queries made via Mxit over time.
- View suburbs electricity net consumption a positive increase if the suburb increased its electricity consumption or negative if the suburb decreased its electricity consumption.
- Provide functionality that track how many times each customer has used the system via Mxit and the website. This would give the person that want to track a customer an option to enter the account number of the customer they want to track, and then display to the person how many times that customer used the system via Mxit and via the website.
- Every call made to the system automatically increments counters and invokes embedded functions that automatically store the information in the database, when the monitoring system is invoked it uses the stored information to draw graphs.
3.6 The interface design

For the back end functionalities to be used by Mxit and the website, no interface was needed. The monitoring tracking system has a user interface; the initial prototype was made and shown to the client where I took note of the feedback given. Unfortunately it was not possible to get hold of the client at the City of Cape Town to have a second iteration (due to his tight schedule) to show him a second version of the prototype. Therefore I gathered comments made on the first prototype to improve and implement the final product of the monitoring system.

Below I show screen shots of the initial iteration

Below is the high level menu that the monitoring system would consist of.

![Options Menu](image)

On the following pages are the screens shots 1-9 that you would see if you click on the menu.
Screen shot 1. Usage via Mxit

The bar graph below represents the total system usage by people in various suburbs via Mxit since the system went live. I checked with the client that the suburbs below are all that they would be interested in since they are only interested in suburbs in Cape Town so there are no suburbs missing.

<table>
<thead>
<tr>
<th>Pros</th>
</tr>
</thead>
<tbody>
<tr>
<td>The graph gives one picture of all the suburbs usage via Mxit at once.</td>
</tr>
<tr>
<td>A good comparison of all suburbs at once.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not so very clear to read the y-axis because there are many too suburbs distorting the view.</td>
</tr>
<tr>
<td>The suburbs are not grouped at all therefore not very clear to distinct which group of suburbs are using MXit the most and which are using it the least.</td>
</tr>
<tr>
<td>Due to many suburbs displayed on the y-axis the bars are too close to each other therefore not clear to read.</td>
</tr>
</tbody>
</table>
Screen shot 2. Usage of the system via the website

The graph below represents the total number of hits on the websites since the system went live.

Pros and cons as for previous screen.
Screen shot 3. Suburbs savings

The graph below shows the suburbs average saving; the savings are the average Rand saved from electricity by each suburb since it started using the system.

- The graph shows average electricity savings in rand but electricity tariffs changes a lot therefore it does not give a clear indication as to how much electricity a suburb has reduced by.
Screen shot 4. Bulb Queries on the web

The graph below represents daily number of queries made on the bulb function via the web over time.

Pros

- The graph clearly shows how the number of bulb queries change daily.

Cons

- Only the last month number of bulb queries displayed prior months number of bulb queries made not displayed.
- This type of chart is not very easy to read the values off, most users find bar graphs much more easier to read.
- Similar screens for usage of other functions via the web or Mxit are not shown here.
The following screen shots are the final implementation of the monitoring system interface design.

Final screen shot 1.

Below is the hierarchical menu of the monitoring system.

![Hierarchical Menu]

Changes were made to the menu so that it is now hierarchical; to give more detailed options to choose to display on the graph.

On the following pages I show what each nested menu from the main menu above displays.
Final screen shot 2.

Once the highlighted option above is clicked the graph below is what gets displayed.

Final screen shot 2.1. The graph below shows the top 12 suburbs that used Mxit the most, with the total number of people per suburb that used the system via Mxit.

Changes made

- Since the suburbs of interest were only 36, suburbs are now ranked into 3 ranks; each rank consists of 12 suburbs. The y axis is now clear to read as it only displays 12 suburbs at once.

- Within each rank the suburbs were displayed from highest to lowest usage. A similar approach is used for the other monitoring option Suburb Web Usage.
Final screen shot 8.

Once the highlighted option above is clicked the graph below is displayed.

Final screen shot 8.1. The graph below shows daily overall number of people that used the system regardless via Mxit/the website and regardless of which suburb they stay in. The days shown are the last few days of the last month in this case October 2009.

Changes made

- The graph is changed to a bar graph as most people find bar graphs easier to read and understand.
- There are two options to view the overall number of people that used the system, either daily thereby displaying only the last few days as above, and the other one to view monthly overall number of people that used the system.
Final screen shot 9.

Once the highlighted option above is clicked the graph below is displayed.

Final screen shot 9.1. The graph below shows monthly overall number of people that used the system regardless via Mxit/the website and regardless of which suburb they stay in. The months shown are the last few months of usage.

Changes made

- The graph is changed to a bar graph as most people find bar graphs easier to read and understand.
- There are two options to view the overall number of people that used the system, either daily thereby displaying only the last few days, and the other one to view monthly overall number of people that used the system as in the above bar graph.
Once the highlighted option above is clicked the graph below is displayed.

Final screen shot 10.1. The graph below shows daily number of bulb queries made via the website. The days shown are the last few days.

Changes made

- The graph is changed to a bar graph as most people find bar graphs easier to read and understand.
- There are two options to view the number of bulb queries made via the website, either daily thereby displaying only the last few days as above, and the other one to view monthly number of bulb queries made via the website. The other bulb and geyser options are done in the same way
Once the highlighted option above is clicked the graph below is displayed.

Final screen shot 18.1. The graph below shows suburb’s increase or decrease in electricity consumption in kilowatts.

Changes made

- The graph is changed to display actual kilowatts instead of rand savings each suburb made from electricity as the client suggested.
Final screen shot 19

Once the highlighted option above is clicked Final screen shot 20 below is shown.

Final screen shot 21 below shows what is displayed after clicking the “view result” button.

What is displayed in the in the above is “Account number 142082046 used Mxit 0 times and the website 8 times” or an error message if an account number is not found in the database.
4. Implementation

This section discusses the implementations of the back end and the monitoring system. Some of the code in the methods shown under this section have been cut out because of space, all methods do have try and catch however this is not shown in all of them under this section, the focus here is more on trying to explain how the methods were implemented rather than how errors were handled, to see the full methods please see the source code.

Data in excel files were sent by the City of Cape Town, first thing done was to create a database and load the data. Mysql database was created and the data loaded, and the ConnectorJDBC (Java DataBase Connectivity), driver was used to access Mysql database from java. Functions were implemented to retrieve values from the database; an account number was used to search the database to retrieve information of a particular customer.

4.1 Methods implemented

Here is an example of the methods of the back end showing how all access is affected.

- Return customer name of a given account number.

```java
public static String getcustomername(int account) {
    String name=null;
    Statement statement ;
    statement= databaseconnection.createStatement ();
    statement.executeQuery("   SELECT DISTINCT businesspartner from customerdetail Where contractaccount = "+account");
    ResultSet result1 = statement.getResultSet ();
    while (result1.next ()) {
        name = result1.getString ("businesspartner");
    }
    return name;
}
```
• Return current/latest month electricity bill (rand value, kilowatts, carbon emission) of an account number.

An account number is used to retrieve the latest month bill of that account number in the database. Kilowatts returned were multiplied by 0.5883 to get the corresponding kilograms of carbon emission emitted. This is following the statement that per kilowatts hour 0.0005883 metric tons Carbon dioxide is emitted [12], the metric tons was converted to kilograms from the conversion that 1 metric tons = 1000 kilograms [16]. The rand value for the consumed kilowatts was calculated using that 44.384 kilowatt hour is free and next is charged at 0.4946, per kilowatt hour, this was from a sample receipt from City of Cape Town.

• Return past electricity history of an account number

An account number is used to retrieve all the months’ electricity bills from the database corresponding to the given account number.

One problem encountered was that the months were not ordered by bill date correctly, because of the bill date excel file string format ("28.01.2009"), so this had to be corrected.

• Return what past electricity history of an account number would have been if certain changes had been done.

This method takes in parameters, either increase/decrease hours of usage per day of various electricity appliances. Each is multiplied by its corresponding appliance electricity consumption rate and by 30 days to get the total per month, this is summed together to get the overall changed consumption.

• Different functions to calculate average rand value, carbon emission, and consumption from what past electricity of an account number would have been if certain changes had been done.
The used estimated electricity consumption by appliances per hour from [17] is given below.

Estimated fan rate=0.066 kWh per hour, estimated non saver 40 Watt bulb rate=0.04 kWh per hour, estimated 16 Watt saver bulb rate=0.016 kWh per hour, estimated heater rate=2.5 kWh per hour, estimated washing machine rate=1.25 kWh per hour, estimated kettle rate=1.5 kWh per hour, microwave oven rate=1.35 kWh per hour, geyser rate=5.81 kWh per hour;

- Predict future electricity consumption based on past months electricity history of the user, compute its average rand value, carbon emission and kilowatts.

- Different functions to calculate average rand value, carbon emission, and consumption from predicted future electricity consumption above.

As an example below is the Past electricity history of account number 208402026 retrieved from the database where the data received from the City of Cape Town were loaded.

<table>
<thead>
<tr>
<th>Bill date</th>
<th>kilowatts</th>
<th>Rand value</th>
<th>Carbon emission</th>
</tr>
</thead>
<tbody>
<tr>
<td>01.06.2009</td>
<td>784.0</td>
<td>371.46</td>
<td>461.23</td>
</tr>
<tr>
<td>04.05.2009</td>
<td>943.0</td>
<td>446.79</td>
<td>554.77</td>
</tr>
<tr>
<td>30.03.2009</td>
<td>701.0</td>
<td>332.13</td>
<td>412.4</td>
</tr>
<tr>
<td>03.03.2009</td>
<td>720.0</td>
<td>341.14</td>
<td>423.58</td>
</tr>
<tr>
<td>04.02.2009</td>
<td>630.0</td>
<td>298.49</td>
<td>370.63</td>
</tr>
<tr>
<td>08.01.2009</td>
<td>983.0</td>
<td>465.75</td>
<td>578.3</td>
</tr>
</tbody>
</table>

Below is the computed predicted electricity history of account number 208402026 based on past history given above:

<table>
<thead>
<tr>
<th>Bill date</th>
<th>kilowatts</th>
<th>Rand value</th>
<th>Carbon emission</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.12.2009</td>
<td>943.2</td>
<td>444.55</td>
<td>554.88</td>
</tr>
<tr>
<td>1.11.2009</td>
<td>590.2</td>
<td>269.96</td>
<td>347.21</td>
</tr>
<tr>
<td>1.10.2009</td>
<td>680.2</td>
<td>314.47</td>
<td>400.16</td>
</tr>
<tr>
<td>1.9.2009</td>
<td>661.2</td>
<td>305.08</td>
<td>388.98</td>
</tr>
<tr>
<td>1.8.2009</td>
<td>903.2</td>
<td>424.77</td>
<td>531.35</td>
</tr>
<tr>
<td>1.7.2009</td>
<td>744.2</td>
<td>346.13</td>
<td>437.81</td>
</tr>
</tbody>
</table>
I used the last 6 months consumption history, noting down the rate of change to predict next 6 months consumption.

1. First sum up the difference between consecutive past months.

Sum the difference = \((\text{pastmonth}_1 - \text{pastmonth}_2) + (\text{pastmonth}_2 - \text{pastmonth}_3) + (\text{pastmonth}_3 - \text{pastmonth}_4) + (\text{pastmonth}_4 - \text{pastmonth}_5) + (\text{pastmonth}_5 - \text{pastmonth}_6)\)

\[= (784-943) + (943-701) + (701-720) + (720-630) + (630-983) = -199\]

2. Get the average net difference = sum the difference/5

\[= -199 / 5 = -39.8\]

3. Add the net difference to each past month to get the consequent predicted month consumption.

\[
\begin{align*}
\text{predictedmonth}_6 &= \text{pastmonth}_6 + \text{net difference} = 983 + (-39.8) = 943.2 \\
\text{predictedmonth}_5 &= \text{pastmonth}_5 + \text{net difference} = 630 + (-39.8) = 590.2 \\
\text{predictedmonth}_4 &= \text{pastmonth}_4 + \text{net difference} = 720 + (-39.8) = 680.2 \\
\text{predictedmonth}_3 &= \text{pastmonth}_3 + \text{net difference} = 701 + (-39.8) = 661.2 \\
\text{predictedmonth}_2 &= \text{pastmonth}_2 + \text{net difference} = 943 + (-39.8) = 903.2 \\
\text{predictedmonth}_1 &= \text{pastmonth}_1 + \text{net difference} = 784 + (-39.8) = 744.2
\end{align*}
\]

Predicted carbon emission = predicted consumption \(\times 0.5883;\)

While it is clear that prediction is very hard and that this formula could be easily regarded as imperfect, the client required only that any reasonable formula be used in order to illustrate how the system would work if available to Cape Town citizens. If necessary the formula would be given to experts in the field to alter subsequently, before the system was deployed to the public.
- Predict future electricity consumption based on electricity consumption and on changes the user intent to make, compute its average rand value, carbon emission and kilowatts.

This method combines the actions of the last two methods. The former calculates what the history would have been, and then the other feeds those returned values into the prediction method.

- Different functions to calculate average rand value, carbon emission, and consumption from predicted future electricity consumption above.

- Return saved Rand per month if a given number of saver bulbs are replaced with saver light bulbs.

- Return carbon emission per month if a given number of saver bulbs are replaced with saver light bulbs.

These two methods take in a parameter which is the number of non saver light bulbs to be replaced as shown below:

```
double getRandsPerMonthSavedFromReplacingNonSaverBulbWithSaverBulbs(int numberOfBulbsToReplace)

double getCarbonEmissionPerMonthSavedFromReplacingNonSaverBulbWithSaverBulbs(int numberOfBulbsToReplace)
```

The two methods above assumes that the number non saver light bulbs will be replaced by the same number of saver light bulbs and that the bulbs are used for 6 hours (19h00 - 00h00) a day. The methods use the estimated 40 Watt non saver bulb of 0.04 kwh and the estimated 16 Watt saver light bulb of 0.016[17]. The kilowatts saved per month as shown below:

\[
\text{KilowatsFromUsingSaverBulbs} = \text{estimatedsaverbulbrate} \times \text{numberOfSaverBulbToReplaceby} \times \text{numberofHoursTheBulbsIsUsedPerDay} \times \text{numberofdaysInAMonth};
\]

\[
\text{KilowatsFromUsingNonSaverBulbs} = \text{estimatednonsaverbulbrate} \times \text{numberOfNonSaverBulbToBeReplaced} \times \text{numberofHoursTheBulbsIsUsedPerDay} \times \text{numberofdaysInAMonth};
\]

\[
\text{KilowatsSaved} = \text{KilowatsFromUsingNonSaverBulbs} - \text{KilowatsFromUsingSaverBulbs};
\]

Rand saved per month is calculated by multiplying the kilowatt saved by 0.4946 (R 0.4946 per kilowatt from a sample receipt from City of Cape Town). The saved carbon emission kilograms per month is calculated by multiplying the kilowatts saved by 0.5883 [12].
• Return current/latest month Water bill (rand value, kiloliters) of an account number.

An account number is used to retrieve the latest month bill of that account in the database.

The rand value for the consumed kiloliter was calculated using the following extract from a sample receipt from City of Cape Town.

<table>
<thead>
<tr>
<th>Water consumption charge</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) 6.3120 kl free</td>
</tr>
<tr>
<td>(2) 6.3130 kl @ R 3.33</td>
</tr>
<tr>
<td>(3) 8.4160 kl @ R7.100</td>
</tr>
<tr>
<td>(4) 21.0410 kl @ R10.5200</td>
</tr>
<tr>
<td>(5) 3.9180 kl @ R12.9900</td>
</tr>
</tbody>
</table>

Below is how the above water consumption charge was implemented.

```java
public static double getpredictedWaterRandValue(double consump )
{
    double consumption=consump;
    double rand=0;
    double left=consumption-6.312;//6.312 free water
    double left1;
    double left2;
    if(left>0)
        while(left>0)
        {
            left1=left;
            if( (left - 6.313)>=0 )
            {
                rand = (rand) + (6.3133*3.33);
                left= (left - 6.313);
            } //if
            if( (left - 8.416)>=0 )
            {
                rand = ( rand ) + ( 8.416*7.1 );
                left= (left - 8.416);
            } //if
            if( (left - 21.041)>=0 )
            {
                rand = ( rand ) + ( 21.041*10.52 );
                left= (left - 21.041);
            } //if
            if( (left - 3.918)>=0 )
            {
                rand = ( rand) + (3.918*12.99 );
                left= (left - 3.918);
            } //if
            left2=left;
            if(left1==left2)
                break; //if there is no change then it should break to avoid an infinite while loop
        } //while
    return (rand);
}
```
This extract from receipt from City of Cape Town shows the formula used to calculate sewerage disposal costs.

Sewerage (disposal) charge
(2) 4.4190 kl free (2) 4.4180 kl @ R 4.0100 (3) 5.8920 kl @ R 8.5200 (4) 14.7290 kl @ R 9.3200 (5) 2.7420 kl @ R 9.7800 Below is how the above sewerage (disposal) charge was implemented.

Water and sewerage consumption methods were created in a similar way to those for electricity, in other words these return

- Latest bill
- Past history
- What historical values would have been if savings or changes were made
- Predicted monthly consumption in the future
- Predicted monthly consumption if changes effected

The changes/savings catered for were reduced car washes, baths, and shower usage. The estimated water consumption used is given below:

Estimated bath water consumption=0.16 kiloliter; [18]
Estimated shower water consumption per minute= 0.012 (0.06/5) kiloliters;//"a five-minute shower uses 60 liters" [18]
Estimated pipe washing a car water consumption =0.3 kiloliters; [18]
- Create password of a user

This method takes in the account number, password, phone number, an integer indicating whether the user has chosen to remember or not to remember password. The method of creating the password this way was necessary for use on the Mxit interface of the project. The password must be more than 6 characters long. Below is the pseudo code for this method.

```java
createPassword ( int accountnumber, String passwordTobe,int phone, int automaticallyogin)
{
    if(passwordTobe.length()>=6)
    {
        statement= databaseconnection.createStatement ();

        statement.executeUpdate("update customerdetail set password= "+passwordTobe+"
Where contractaccount = "+accountnumber);

        statement.executeUpdate("update customerdetail set phonenumber= "+phone+"
Where contractaccount = "+accountnumber );

        statement.executeUpdate("update customerdetail set automaticlogin= "+automaticallyogin+" Where contractaccount = "+accountnumber );
    }
}
```

- Automatically log in with cell phone number, this method check if user has set remember password as default to log in only with phone number and returns a Boolean. It takes in a phone number as a parameter and returns true if user has set to remember password else false. This method was necessary for use on the Mxit interface of the project.

- Authentication of account number and password, for a user that has not set to remember password.
4.2 Loading data into the database

The first thing to do was to create a database and to load and store the data received from the City of Cape Town. The sample data received from the city was a small subset of the city’s database from which data relevant needed to be generated and cleaned. The received data had some information which was irrelevant for the project, this was resolved as follows, instead of loading the one big spreadsheet (with irrelevant data) in the database; the received big spreadsheet file was separated into two separate smaller spreadsheets named customer.csv and payment.csv that contained only relevant information. Three tables were created “customer”, “customerdetail”, and “paymentdetail”, the customer.csv and payment.csv were then loaded into the customer and the payment detail table respectively. The “customer” table was an intermediate relation loaded with data from the spreadsheet that had lots of repetition about customers’ detail. Another table (“customerdetail”) was created and loaded with unique details selected from the customer table, after this the customer table was deleted. The customer detail table consists of static personal information about customers and the payment detail table contained payment detail. Below is how the three tables were created and loaded with data.

CREATE TABLE customer (contractaccount INT, businesspartner VARCHAR (50), streetaddress VARCHAR (50), suburb VARCHAR (50), postalcode INT, password VARCHAR (50));

LOAD DATA INFILE 'customer.csv' replace INTO TABLE customer FIELDS TERMINATED BY ',' LINES TERMINATED BY '
' (contractaccount, businesspartner, streetaddress, suburb, postalcode, password);

CREATE TABLE customerdetail(contractaccount INT, businesspartner VARCHAR (50), streetaddress VARCHAR (50), suburb VARCHAR (50), postalcode INT, password VARCHAR (50));

insert into customerdetail(contractaccount, businesspartner, streetaddress, suburb, postalcode, password) SELECT DISTINCT contractaccount, businesspartner, streetaddress, suburb, postalcode, password from customer;

CREATE TABLE paymentdetail (contractaccount INT, device INT, billdate VARCHAR(50), expense VARCHAR(50), consumption DECIMAL (20,4), differenceindays INT, dailyaverage DECIMAL (20,4), realmetersresult INT, randvalue DECIMAL(20,2));

LOAD DATA INFILE 'payment.csv' replace INTO TABLE paymentdetail FIELDS TERMINATED BY ',', LINES TERMINATED BY '\n' (contractaccount, device, billdate, expense, consumption, differenceindays, dailyaverage, realmetersresult, randvalue);
4.3 Monitoring system usage via the web and Mxit interface

This system, which aims to provide a tool for City of Cape Town employees, required (1) extending system functionality to record all method calls in the database, (2) generating calls artificially so as to populate this with enough data (3) generating artificial consumption values (showing some savings) and finally (4) creating the tool itself.

The first thing to do was to add the JFreechart jar files to the project, in order to draw the required graphs displaying system usage. To distinguish queries made from Mxit and the Website, the constructor to my Data class was changed as below:

Data mxitObject=new Data(accountnumber,1);//1 indicate mxit

Data websiteObject=new Data(accountnumber,2);//2 indicate the website

It is necessary to distinguish which interface and account number is being used for when the table of usage of the system is updated. Every method in the Data class updates the relevant tables for monitoring purposes every time it is invoked.

On following pages I show the tables that are used and updated whenever a function in the Data class is invoked via Mxit or the web interface.
The table created below is necessary for updating or storing number of times Mxit is used in different suburbs. First the table is created and all the account numbers and their respective suburbs are inserted into this table. Every time Mxit sends a request to my Data class, the account number's (that made the query) number of logon via to Mxit interface is incremented, which in turn increases number of Mxit usage of the suburb where the owner of the account number stays.

CREATE TABLE usageOfSystemviamxit (contractaccount INT, logoncount INT DEFAULT 0, suburb VARCHAR (50));

INSERT INTO usageOfSystemviamxit(contractaccount, suburb) SELECT DISTINCT contractaccount, suburb FROM customer;

To view all suburbs and their respective number of usage of Mxit the following select stament is what gets executed. JFree Chart is used to draw a bar graph that indicates different suburb's usage of the system via Mxit.

SELECT suburb, sum(logoncount) as total from usageofsystemviamxit group by suburb order by total

The Same approach is used updating or storing the number of times the website interface is used in different suburbs.

CREATE TABLE usageOfSystemviaweb (contractaccount INT, logoncount INT DEFAULT 0, suburb VARCHAR (50));

INSERT INTO usageOfSystemviaweb(contractaccount, suburb) SELECT DISTINCT contractaccount, suburb FROM customer;

To view all suburbs and their respective usage of the website the following select statement is executed. JFree Chart then is used to draw a bar graph that indicates different suburb's usage of the system via the website.

SELECT suburb, sum(logoncount) as total from usageOfSystemviaweb group by suburb order by total
The table created below is necessary for updating or storing overall number of logins made into the system regardless via Mxit or via the website over time. First the table is created and every time a query is made from the web or Mxit interface, system time (day, month and year) is generated. The date is stored in the table created, if a query is the first one made on the generated day the number of login corresponding to that day is set to 1 else if it is not the first query made that day the number of logins corresponding to that day is incremented by 1.

Create table systemoveralllogins (numberoflogins INT DEFAULT 0,day INT,month INT, year INT);

It was not possible to get users to use the system via Mxit or website so that their usage is tracked in this table so random information was inserted into this table. This is because Mxit and the website interface which were being handled by my colleagues were not yet complete. When the project is put together the table will be updated automatically according to day of usage.

To view last 12 days of the last month usage of the system the following select statement is executed. JFree Chart is then used to draw a bar graph that indicates total number of logins into the system via Mxit/Web in the last 12 days of the last month of usage.

select day,month,year ,numberoflogins from systemoveralllogins where month=( select max(month) from systemoveralllogins where year =(select  max(year) from systemoveralllogins) ) order by day limit 12;

To view last 12 Months usage of the system the following select statement is what gets executed. JFree Chart is then used to draw a bar graph that indicates total number of logins into the system via Mxit/Web in the last 12 months of the last month of usage.

SELECT month,year,sum(numberoflogins) as numberoflogins from systemoveralllogins group by month order by year,month  limit 12;

The remaining parts of the monitoring system (Mxit and Web usage, and by function such as geyser or bulb queries) were handled in the same way.
To get suburb’s change in electricity after using the system two tables were used one to store suburb’s electricity user’s consumption before using the system, the other one to store suburb’s electricity user's consumption after using the system. Electricity data received from City of Cape Town in the table called “paymentdetail” was treated as suburb’s electricity consumption before using the system. To get suburb’s electricity consumption after using the system, since the system was not at the stage used by real users, estimates of next 6 months of electricity users was made based on their past electricity consumption and with some 30 or 40% reduction and inserted into a table called “assumedactualcurrentelectricityconsumptiondata”.

Suburb’s change in electricity after using the system is calculated as suburb's total consumption after using the system minus suburb’s total consumption after using the system. The select statements used are below:

SELECT suburb, sum(consumption) as totalThen from paymentdetail where expense = 'ELECTRICITY' group by suburb;

SELECT suburb, sum(consumption) as totalNow from assumedactualcurrentelectricityconsumptiondata group by suburb;

5. Testing

Querying the database:

In order to test that the right information was retrieved from the database data returned after making a database query were compared with the information in the excel files that were loaded in the database.

Monitoring system usage

The relevant methods were invoked for a couple of times and the corresponding tables that stores how many times the system is used were viewed to see that that they were automatically updated as the method calls were made. The Datagui class that uses these tables (of system usage) to draw graphs displaying the system usage was also run to see that the graphs drawn were displaying the updated information of system usage.

Formulas to predict future usage

This was done manually on a piece of paper and compared to the returned future formulas to predict savings from changed appliance usage to see that they are the same.
Testing system integration

My Data class was needed at the very start of the project by Mxit and the web subsystem, I set up Mysql on my colleagues' computer and gave them the Data class that has functions that connects and retrieve information from mysql database such as user’s electricity/water/sewerage consumption. At the stage the Data class missed methods for tracking system usage as the monitoring system was not yet implemented. However there was already integration between my subsystem and the other two subsystems because throughout their implementation of their subsystems they were each connecting to my Data class stored on their computer by creating an object to it and invoking its methods, the integration worked well for both systems.

When the methods for the monitoring system were complete, the Mxit and the web subsystem were still not complete, I therefore tested the monitoring system by running a dummy class that creates two constructors the same way Mxit and the web system does. To distinguish queries made from Mxit and the Website, the constructor to my Data class was changed as below:

Data mxitObject=new Data(accountnumber,1);//1 indicate mxit
Data websiteObject=new Data(accountnumber,2);//2 indicate the website

I invoked methods in the Data class from the dummy class and ran the monitoring system to check that the graphs drawn were displaying incremented Mxit and Web usage graphs as methods from the Data class were invoked with Mxit object and with web object respectively. This worked right and a demo of the monitoring system was also done with the client who showed appreciation as the graphs of the monitoring system were picking up changes of usage of the system via Mxit and the Web.
6. Conclusion

This project can promote wiser usage of electricity and water; people are likely to react to changes that will save them money. The other aspect of the project is on making people become environmentally friendly by making them aware of how much carbon dioxide is emitted per kwh of electricity consumed. Although water is known not to emit carbon dioxide, heating water with electrical appliances such as ovens or geysers still produce carbon emission. We hope that the project would enlighten many so that a lot of people start switching off the geyser when necessary and generally change their habits to reduce water and electricity usage.

An extension to the project could be online payment of bills, because this project only looks at making people aware of their electricity, water consumption and carbon emission. The one thing not tackled in the project is that people tend to be forgetful about switching off their geysers. Therefore an extension to the project could be researching into automated system where users can just program the switch as to when to switch the geyser on and off. I am not sure if such a device exists already but I think it will be very handy in reducing electricity consumption and thereby decreasing carbon emission because people are very forgetful. Before it can be deployed the system needs to be tested with a large number of users and over an extended period of time in such a way that significant amounts of real usage feeds into the monitoring system. The system also needs to be tested on the full extent of data in the City's database in order to see what problems are introduced by large volumes of data.

The project has potential to make a difference on earth resources, and the underlying concept can be applied to a lot of things. The data sample used for the project was small but sufficient to build our project. The monitoring system is a good indicator of how many people are using the system, this can have advantages to the City of Cape Town for example if many people are using the system then more features could be added to the system because it implies the system serves a good communication medium between people and the city. It could also be used as an opportunity to reach out to citizens e.g. by posting notices of public meetings etc. on the site. If a lot of people are not using the system then more user design might be needed to ensure that the system is built in the best interest of the people. The project was worth doing, it is a try to minimize electricity and water and this can affect a lot of people.
References:

[1] Shim DC, Eom TH, Anticorruption effects of information communication and technology (ICT) and social capital, International review of administrative sciences, 75 Issue: 1 Pages: 99-116 Published: MAR 2009


[4] Reidl R, Document-Based Inter-Organizational Information Exchange, Department for Information Technology, University of Zurich, Winterthurerstasse 1900, CH-8057 Switzerland


[8]

[9]

[10]
Brahim Medjahed, and Jerry Cameron, Managing Government Databases, IEEE Computer Society, Volume 34, Published: February 2001

[11]

[12]

[13]

[14]

[15]

[16]
www.metric-conversions.org/weight/metric-tons-to-kilograms.htm Accessed 13 September 2009
[17]

[18]

[19]

[20]

[21]

[22]

[23]
Ljupco Antovski and Marjan Gusev, M-Government frame work

[24]
Jessica Webster; Ray Tomalty; Brett Korteling, Residential Energy Use Characterization and Mapping: Research and development of information for community energy and greenhouse gas planning in Canada, 45th ISOCARP Congress 2009

Acknowledgement

1.

Aziz Hendricks, Used his report (A system for automated geo-referencing) as a guide in laying out this report