Research Journal of Engineering Sciences ISSN 2278 – 9472
Vol. **2(8),** 12-17, August **(2013)** *Res. J. Engineering Sci.*

Blue-Soft: A Bluetooth based Wireless Secure download and upload station

Pal Arnab[1] and TahAvranil[2]
[1]Department of Computer Science, Chandernagore Govt. College, Hooghly, West Bengal, INDIA
[2]Department of Computer Science, University of Texas, EL PASO

Available online at: www.isca.in
Received 15th August 2013, revised 20th August 2013, accepted 26th August 2013

Abstract

The project discussed here is an interactive application system for the employees in a software development firm. The key concepts described in the project are, i) wireless connection between a server and a number of client nodes, from where a lot of users can do their jobs, and the server node, as a medium of storage, using Bluetooth; ii) an editor and a compiler, which supports multiple programming languages like, Python, Java, C, C++, HTML with an option to add other languages if the user want to; iii) an embedded system to check the necessary details regarding their payments and to claim the payment cheque, to check the available loan plans, to apply for a loan, to check the loan payment status etc.; iv) to communicate with the other users, who are currently available or not; v) to check their performances over a certain period; vi) and lastly to check for the available snacks in the office canteen and to order the products to get them at their desk, to reduce the appetite and increase the comfort in work. Furthermore, there is also an 'admin-panel', from where the administrator of the system can do several jobs like, i) to monitor the works of the employees; ii) to add new employee and his necessary details; iii) to send message to a specific employee or forward a notice to all of the employees; iv) to award points to the employees according to their performances and the attendance. The key factor of the system is, the system is not dependent on the identity of the machine, whereas, the integrity of the system lies on the unique identification of the Bluetooth adaptors, and thus, it increases the flexibility of the system. As an example, if an employee decides to do the job from their own PC or Laptop, or any device they are using, they can do their job just by connecting to the server using his 'user id' and 'password'.

Keywords- Bluetooth communication, file server, python programming, wireless cryptography.

Introduction

In the era of energy crisis, I was thinking of something that will save energy, as well as can be used to connect two or more terminals to form an efficient network to serve the purpose the of a small office or a business firm[1]. I had so many options to establish the connectivity, like; standard Cat-5 cable, Wi-Fi technology, Bluetooth and some other technologies. Now, I didn't use Cat-5 cable, because, it requires a lot of energy, and other devices like Router, Switch, Hub etc. In the next, the Wi-Fi technology was efficient but, again, it requires more energy, some other devices like; Wi-Fi Router, Wi-Fi Modem, and Wi-Fi hotspot. So, it is neither energy-efficient nor cost-efficient. So, I didn't use Wi-Fi technology. Last of all, Bluetooth is better option for serving my purpose. It requires low power which results to long battery life. Bluetooth technology can be used within the range of 30-feet, which is reasonable within a small building. Moreover, the data-transfer rate of the Bluetooth technology is quite acceptable, i.e. 3-4Mbps[2]. So, finally, due to the aforesaid benefits, I decided to use the Bluetooth technology for the connectivity in my Network applications. But, there is a major disadvantage using Bluetooth. The radio signals used to connect the Bluetooth devices considerably depends by the humidity of the environment; that affects the service of the Network in times. Since the Bluetooth technology was invented, a few decades ago, there was no such system was made to run the applications in a connected system of a few client terminals and a server terminal. So, I thought, if such an application can be created, which can be used in a software development firm, where the employees can work in some client terminals, and use the server as a purpose of storage and testing the results of the codes written using multiple languages, like; C, C++, JAVA, PYTHON, HTML etc.. And they can also check their other necessary details being an employee in a software developing firm. And, there are some other features of the application, like; there is an administrator terminal for the head of the firm, who will manage the employees, update the necessary details of the employees, and, the most interesting feature, the administrator will be able to monitor the works of the employees as well. The whole system and all the files that are created by any of the user will be stored with an encryption using symmetric cryptography, so this is an additional security measure. Apart from this, there is some other features used like, communication between the employees using message passing, and, ordering some food, from the canteen of the firm from the client if the user want.

Methodology

There are three basic terminals, in the application. One is for the user, other one is for the Canteen owner, and the most importantly, the server terminal. The server terminal has a fixed

MAC address, and the canteen part has a fixed MAC address. And the users of the application whether he is an administrator or a normal user, he can use any of the device available, or he can use his own device like, laptop, or tablet pc, running any version of windows operating system just by installing the application. As a result the MAC addresses of the users are not fixed. The user access the data saved in the server just by logging in to it using his 'user-id' and 'password'. The following figure gives us a top view of the system, how the different nodes are connected.

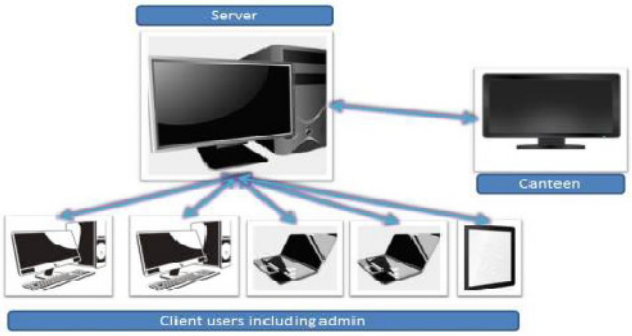

Figure-1
The top view of the system showing how the different nodes are connected

The nodes are connected by the wireless technology, Bluetooth. The maximum manipulation regarding the work of a user, that is, when a user try to compile or run one's source code, the process is done in the client machines, whereas, the calculations, regarding to the payment of the employees, or the security of the system, is done in the server machine. Each and every file in the system is encrypted using DES cryptography[2]. The administrator has two DES cryptographic keys. First one is common for all the admins, which is used to encrypt the files related to the user, like, the file containing the user name and password of the users, or the file containing the list of the files, a user has saved in the system etc. Where, the files related to the admins are also encrypted using a temporary key, which is automatically changed in a timely manner, and the files are also encrypted again correspondingly. Thus if anyone steals any data, he will definitely not be able to figure.ure out what does the encrypted data mean. On the other hand, when the data is sent from one node to another, the data is also encrypted using DES cryptography and sent[3]. Thus, if anyone taps in between two clients, again he will not be able to understand the meaning of the encrypted string. A diagrammatic representation of the entire process is given below in two consecutive figure and the step algorithm follows them.

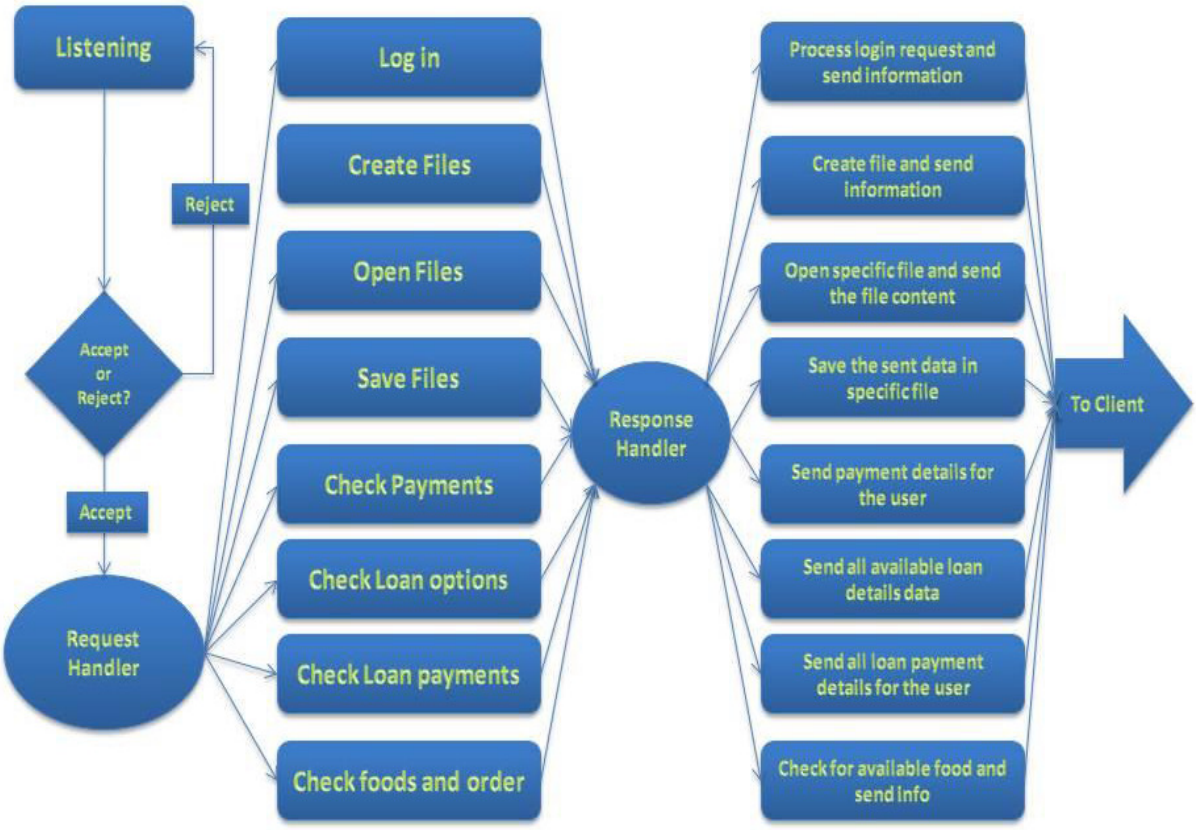

Figure-2
The main process in the server node

Step Algorithm for the server: i. The server node is always in listening mode. ii. Request from the client node is received. iii. If the request is accepted and correct in the specified protocol, then continue to step 4, else step back to 1. iv. The received request is passed to the request handler. v. Then the particular request is then passed to the response handler. vi. The response handler finishes the specified task, and then the processed result is sent back to the client, from where the request has been received. vii. The control is again sent back to the step 1.

Step algorithm for the Response Handler: i. The request is received from the request handler. ii. If the request is authenticated by the user, then forward the control to step 4 (for example, the user is trying to open a file created by other user). iii. Send proper error message to the user. iv. Process the request properly, and send the information to the user node.

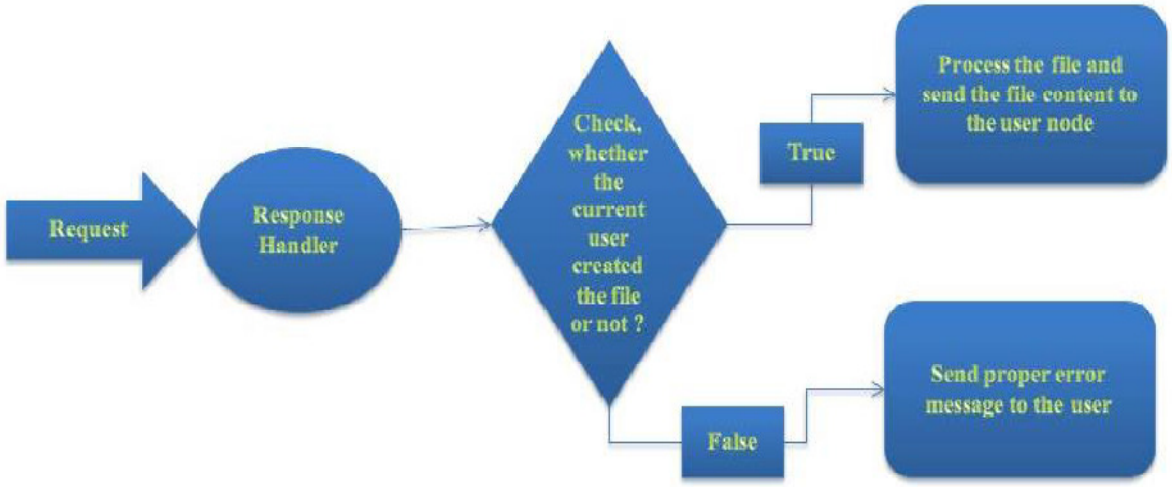

Figure-3
Flowchart of the process in the response handler

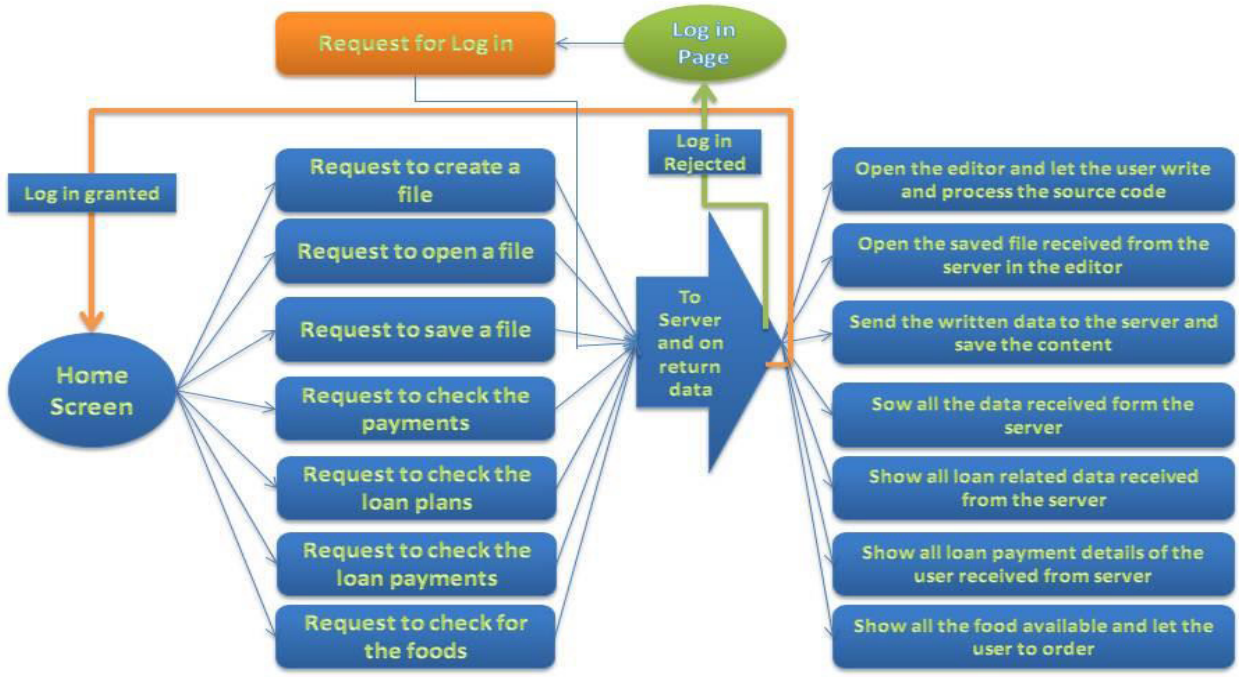

Figure-4
The main process running in the client nodes

Step algorithm for client: i. The client has to log in to the system for using the entire system and facilities, thus, the system prompts the user to login first. ii. The login request is generated and sent to the server. iii. Server responses to the login request. iv. If the server grants the login for the user, control is passed to step 5, else, it is sent back to the step 1. v. The user in the client node gets a home screen, containing several options to start the work.

Data Structure: The entire building block of the permanent data storage system is data structures like, dictionary, list, tuples etc. We have chosen to use simple text files to store the data permanently and to use basic data structures extensively as mentioned earlier. We have used dictionary to store several data which is responsible for the user to grant access to number of services. A dictionary is a special kind of data-structure of python, which support the following features. Keys must be immutable, and this key can be number, string, tuple or anything, but, it cannot be changed after creation, because of hashing. And moreover, the keys must be unique again because of hashing. There are no restrictions of values in a dictionary, and the keys will be listed in arbitrary order[1]. For instance, the file responsible for the users to login to the system looks like this:

Here, the username field is a string, containing the username, and there is a tuple corresponding to the key i.e. the username, and the first field of the tuple contains the password, and the second field contains the string for the generation of the cryptographic key which we use to encrypt the data stored by the user. There are several other similar kinds of files containing other related data. For example, the next screenshot shows the text file containing the list of the files, a user saved in the system. In this case, we have used a dictionary, where the username has been used as the key, and there is a list corresponding to each key, containing the name of the files saved by the user.

As I have mentioned in the motivation, we are trying to build a system, which will be low power consuming as well as cost efficient, we have purposely chosen to use data-structures stored in simple text files, instead of database. As a result the whole system will not be bulky and will be very suitable for the smaller systems as well as for the larger systems.

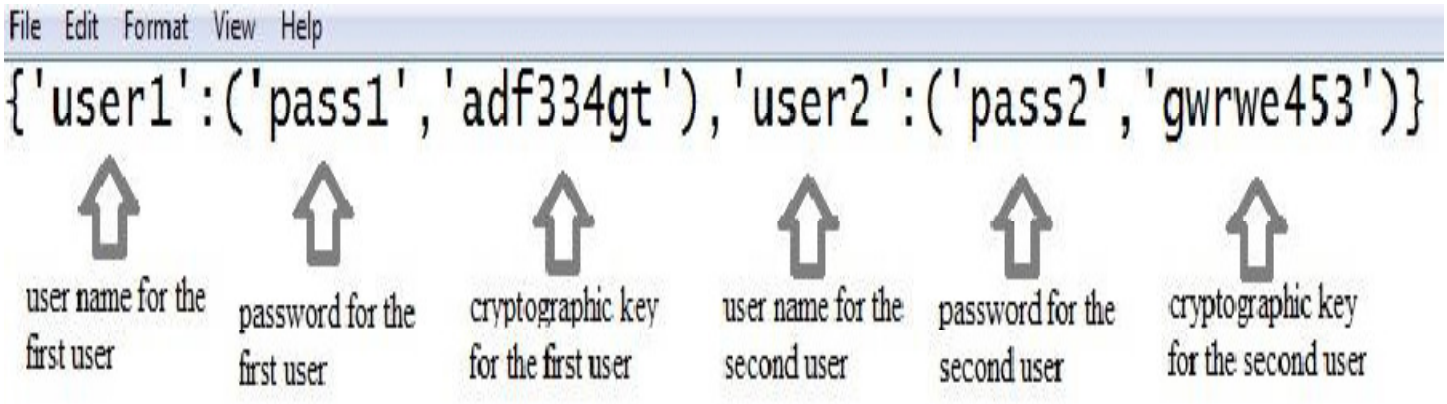

Figure-5
The screen-shot of the file used to store the user-name and password for the user

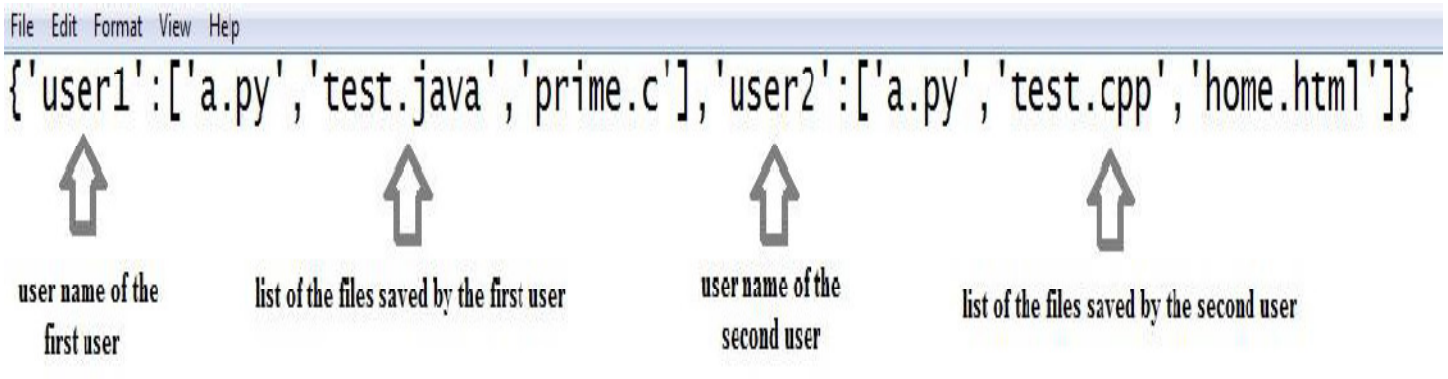

Figure-6
The screen-shot of the file used to store the list of the files saved by the users

Results and Discussion

As mentioned earlier, the user has to login first to use the facilities available for the user. The window for the users to log in to the system looks like the screen-shot given in figure-7.

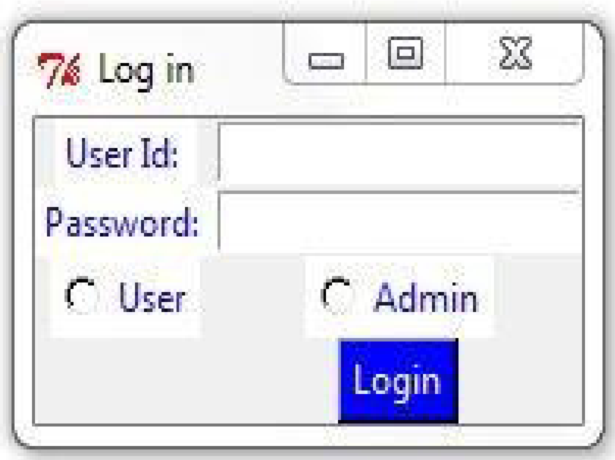

Figure-7
Screenshot for the login window of the users

Again the screen-shot, of the home screen of a user is given below in the figure-8.

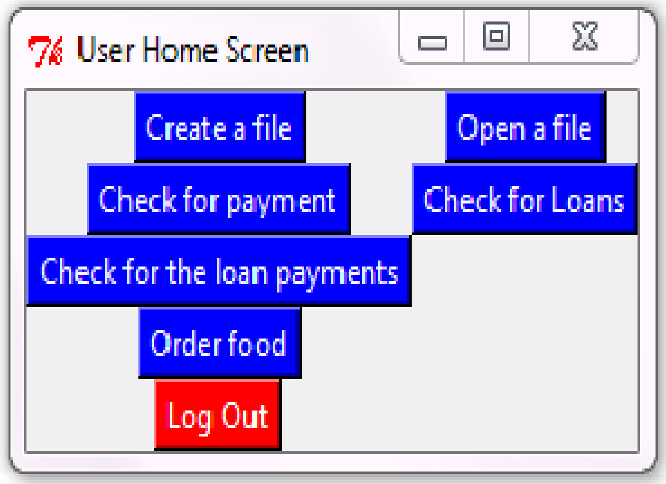

Figure-8
Screen-shot of the Home-screen of the user

In the next screen shot, it shows how the files are stored in the server, i.e. with an encryption. If anyone steals the data, he will never be able to the original text.

Again, the next screen shot in figure-10 shows the, decrypted document, which is readable by the human being.

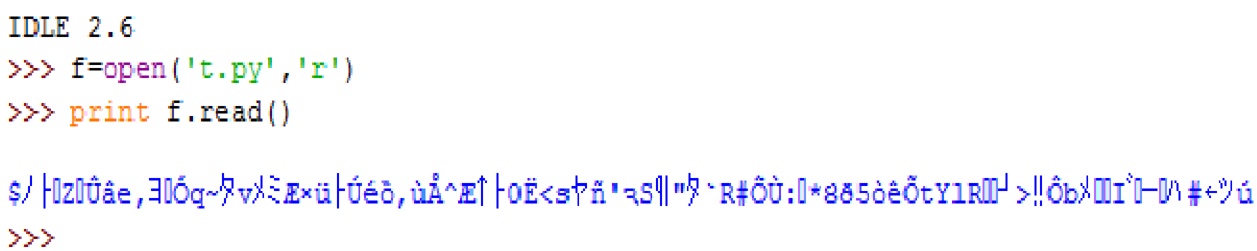

Figure-9
The above screenshot shows the encrypted text stored in the file 't.py'

```
>>> f=open('key.txt','r')
>>> from Crypto.Cipher import DES
>>> key=DES.new(f.read())
>>> f.close()
>>> f=open('t.py','r')
>>> print key.decrypt(f.read())
from Tkinter import *
root=Tk()
root.title('This is my Test file')
root.mainloop()
~~~~~~
>>>
```

Figure-10
The screen-shot of the decrypted document, which is stored in the file 't.py'

Conclusion

The project gave me a lot of first-hand experience of using technologies like Bluetooth. It gave me an experience of GUI programming, and network programming using python. I used data-structures, stored in files. That gave crystal clear ideas of different data structures. In future, this project can be modified, by using database for storing huge data, and that will definitely increase the reliability of the application. And, depending on the need of the number of users, Wi-Fi can be used as the connecting technology, but we will have to consider the energy efficiency and cost efficiency. Furthermore, we can use public key cryptography, to better the security of this application. I will definitely work on this project later and will try to modify it, to improve the user experience as well as reliability in terms of security.

References

1. Tah A., A deadline-driven epidemic data collection protocol suitable for tracking interpersonnel rendezvous (January 1, 2010), ETD Collection for University of Texas, El Paso. Paper AAI1483985. http://digitalcommons.utep.edu/ dissertations/AAI1483985 **(2013)**

2. https://code.google.com/p/pybluez/wiki/Documentation **(2013)**

3. http://en.wikipedia.org/wiki/Data_Encryption_Standard **(2013)**

4. http://www.python.org/ **(2013)**

5. Prakash Vaibhav V. and Kutnikar Ajay L., Green Intelli Campus Using Radio Frequency Technology, *ISCA J. Engineering Sci.,* **1(1),** 8-13 **(2012)**