ISSN: 2379-3686 International Journal of Science Research and Technology
Volume 1 Issue 2, p p 44-50, 15th December 2015

SMART ATTENDANCE SYSTEM THROUGH BLUETOOTH ENABLED DEVICES

Dr.Shantakumar B. Patil
Professor and HOD, Department of Computer Science & Engineering Nagarjuna College of Engineering & Technology, Bengaluru

Dr. Premjyoti Patil
Professor, Department of Computer Science & Engineering Nagarjuna College of Engineering & Technology, Bengaluru

Harshavardhana Doddamani
Research Scholar Nagarjuna College of Engineering & Technology, Bengaluru,
Visvesvaraya Technological University, Belagavi

Abstract: Wireless communication is one of the fastest-growing technologies. The demand for connecting devices without the use of cables is increasing everywhere. Wireless LANs can be found on college campuses, in office buildings, and in many public areas. In our paper we are making use of the existing technology to mark the attendance for the staffs of the institutes. It saves lot of paper and time. Already many people have worked on the same concept. The disadvantage of the existing systems is, throughout the working hours of the institute the staff has to enable the Bluetooth. This may lead to the possibility of copying of personal information from one mobile to other Bluetooth enabled mobile. Sometimes the information maybe eavesdropped and hence confidentiality of the data is lost. To overcome this drawback we are proposing the system called "Smart Attendance System through Bluetooth Enabled Devices". In this work the devices are enabled for the specified time period (twenty minutes). Failing which, the staff of the particular device is marked as absent. Before marking absent it checks the status of the staff through SLMS software. This system also has the facility of sending an absent message to those faculties, who are absent and not applied any type of leaves. At the end of month this system generates report like number of days faculty worked for the particular month including Sunday and holiday, number of leaves and number of leaves without payment (LOP).

I. BLUETOOTH

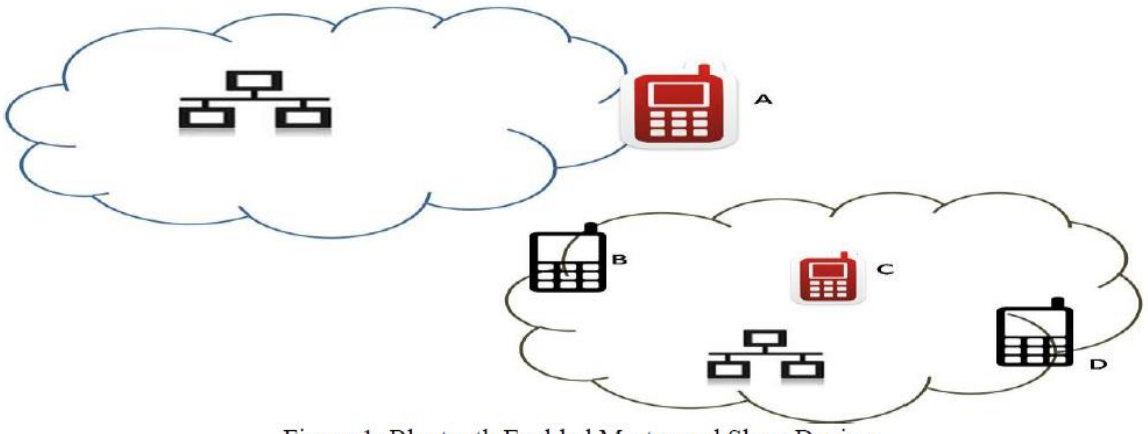

Figure.1: Bluetooth Enabled Master and Slave Devices

Bluetooth is a wireless LAN technology designed to connect devices of different functions such as telephones, notebooks, computers (desktop and laptop), cameras, printers, coffee makers, and so on. A Bluetooth LAN is an ad hoc network, which means that the network is formed spontaneously; the devices, sometimes called gadgets, find each other and make a network called a piconet. A Bluetooth LAN can even be connected to the Internet if one of the gadgets has this capability. A Bluetooth LAN, by nature, cannot be large. If there are many gadgets that try to connect, there is chaos. Bluetooth technology has several applications. Peripheral devices such as a wireless mouse or keyboard can communicate with the computer through this technology. Monitoring devices can communicate with sensor devices in a small health care center. Home security devices can use this technology to connect different sensors to the main security controller. Conference attendees can synchronize their laptop computers at a conference. Today, Bluetooth technology is the implementation of a protocol defined by the IEEE 802.15 standard.

The standard defines a wireless personal-area network (PAN) operable in an area the size of a room or a hall.

II. ARCHITECTURE

Bluetooth defines two types of networks: piconet and scatternet.

A .Piconets

A Bluetooth network is called a piconet**,** or a small net. A piconet can have up to eight stations, one of which is called the primary; the rest are called secondary. All the secondary stations synchronize their clocks and hopping sequence with the primary. Note that a piconet can have only one primary station. The communication between the primary and the secondary can be one-to-one or one-to-many. Figure shows a piconet.

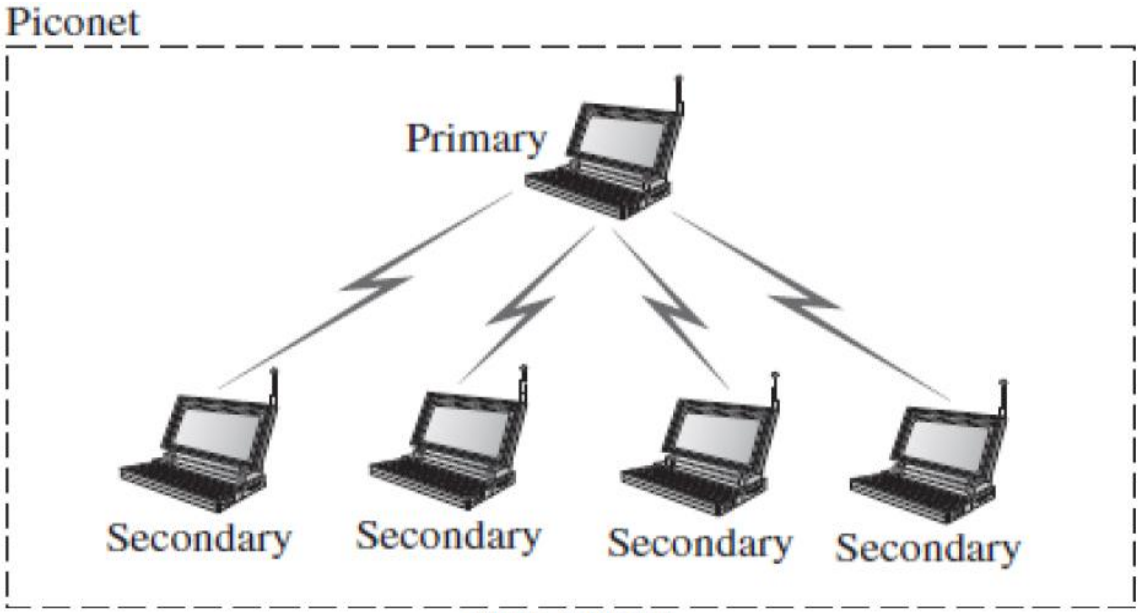

Figure.2: Piconet

Although a piconet can have a maximum of seven secondary, an additional eight secondary can be in the parked state. A secondary in a parked state is synchronized with the primary, but cannot take part in communication until it is moved from the parked state. Because only eight stations can be

active in a piconet, activating a station from the parked state means that an active station must go to the parked state.

B. Asynchronous socket programming

There is usually some sort of waiting involved. During this time, the controlling thread blocks and can't do anything else, such as respond to user input or display progress information. To avoid these pitfalls of synchronous programming, it is possible to use multiple threads of control, with one thread dedicated to each task that requires some waiting. That can get quite hairy and complicated, though, so instead we'll turn to using asynchronous techniques as a solution. The first step in asynchronous programming is to switch the sockets to non-blocking mode, so that all the operations that would block (wait) beforehand return immediately instead. The idea is "Don't wait for something to happen. Just get it started and we'll figure it out later". To switch a socket into non-blocking mode, use the setblocking method and pass it False. Conversely, to switch back into blocking mode, pass it True. For example:

```
from bluetooth import *
sock = BluetoothSocket( RFCOMM )
sock.setblocking( False )
s.bind(("", get_available_port( RFCOMM ))).
```

C. Scanning for nearby devices

After choosing the local Bluetooth adapter to use and allocating system resources, the program is ready to scan for nearby Bluetooth devices. In the example, hci_inquiry performs a Bluetooth device discovery and returns a list of detected devices and some basic information about them in the variable ii. Int hci_inquiry(int dev_id, int len, int max_rsp, const uint8_t *lap, inquiry_info **ii, long flags);
Here, the function doesn't actually use the socket opened in the previous step. Instead, hci_inquiry takes the resource number returned by hci_get_route (or hci_devid) as its first parameter. Most other functions we'll see will use the socket opened by hci_open_dev, but this one creates its own internal socket. The inquiry lasts for at most 1.28 * len seconds, and at most max_rsp devices will be returned in the output parameter ii, which must be large enough to accommodate max_rsp results. We suggest using a max_rsp of 255 for a standard 10.24 second inquiry. If a flag is set to IREQ_CACHE_FLUSH, then the cache of previously detected devices is flushed before performing the current inquiry. Otherwise, if a flag is set to 0, then the results of previous inquiries may be returned, even if the devices aren't in range anymore.
The inquiry_info structure is defined as

```
typedef struct {
bdaddr_t bdaddr;
uint8_t pscan_rep_mode;
uint8_t pscan_period_mode;
uint8_t pscan_mode;
uint8_t dev_class[3];
uint16_t clock_offset;
} __attribute__ ((packed)) inquiry_info;
```

For the most part, only the first entry - the bdaddr field, which gives the address of the detected device – is of any use. Occasionally, there may be a use for the dev_class field, which gives information about the type of device detected (i.e. if it's a printer, phone, desktop computer, etc.) and is described in the Bluetooth Assigned Numbers 2. The rest of the fields are used for low level communication, and are not useful for most purposes. If you're interested, the Bluetooth specification has all the gory details.

D. Determining the user-friendly name of a nearby device:

Once a list of nearby Bluetooth devices and their addresses has been found, the program determines the user-friendly names associated with those addresses and presents them to the user. The hci_read_remote_name function is used for this purpose. int hci_read_remote_name(int hci_sock,

const bdaddr_t *addr, int len, char *name, int timeout) hci_read_remote_name tries for at most timeout milliseconds to use the socket hci_sock to query the user-friendly name of the device with Bluetooth address addr. On success, hci_read_remote_name returns 0 and copies at most the first len bytes of the device's user-friendly name into name. hci_read_remote_name only tries to resolve a single name, so a program will typically invoke it many times to get a list of all the use-rfriendly names of nearby Bluetooth devices.

III. ALGORITHM

Switch on Master and the Slave Bluetooth devices.
2. One time registration of slave device with the Master as shown in figure 4.
3. Enable both the master and the slave Bluetooth devices.
4. Master captures the incoming and outgoing times of slave when it comes in the range of master and the same is marked and maintained in front of the particular device ABC as shown in the figure 3.
5. Master is automatically enabled and disabled twice a day.
6. If slave does not come in the range of master, during the enable period, it checks the status of Bluetooth device XYZ for the particular day.
7. It sends the message of "not present" to that slave device which has not applied any type of leaves through SLMS such as CL, EL, SCL, OOD etc.
8 At the end it generate the monthly report.

IV. STAFF LEAVE MANAGEMENT SYSTEM (SLMS)

Staff Leave Management System is software designed and developed by the team of Nagarjuna College of Engineering and technology, Bengaluru and implemented same in the college as well as in its sister institutes. SLMS is very simple, user friendly and it saves lot of paper and time. The faculty of these institutes uses this software to apply any type of leaves through on line. When a faculty applies any type of leave he/she need to make alternate arrangement of duty for a particular day with date, time and reason for the leave along with the name of the alternate staff. After completion of all the formalities, the complete information is passed to the corresponding alternate staff. After the approval of the alternate staff the same is passed to the HOD and the Principal for final approval. The information which the principal receives is in the consolidated form of different faculty of different departments.

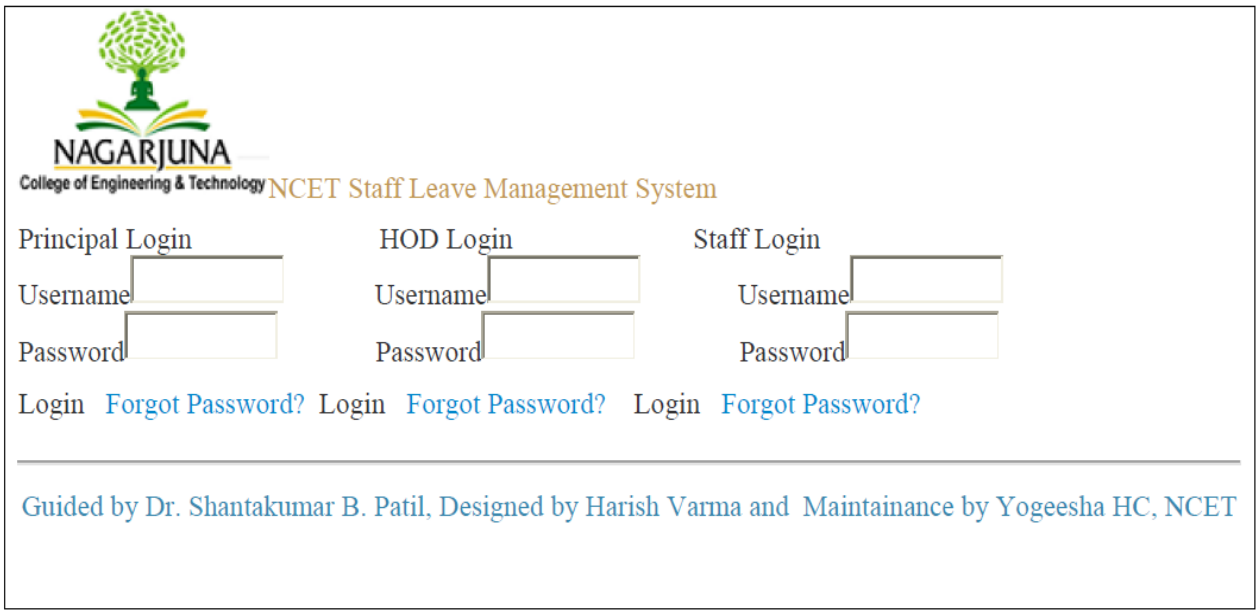

Figure. 3: Staff Leave Management System

V. RESULT

E. Snap Shot

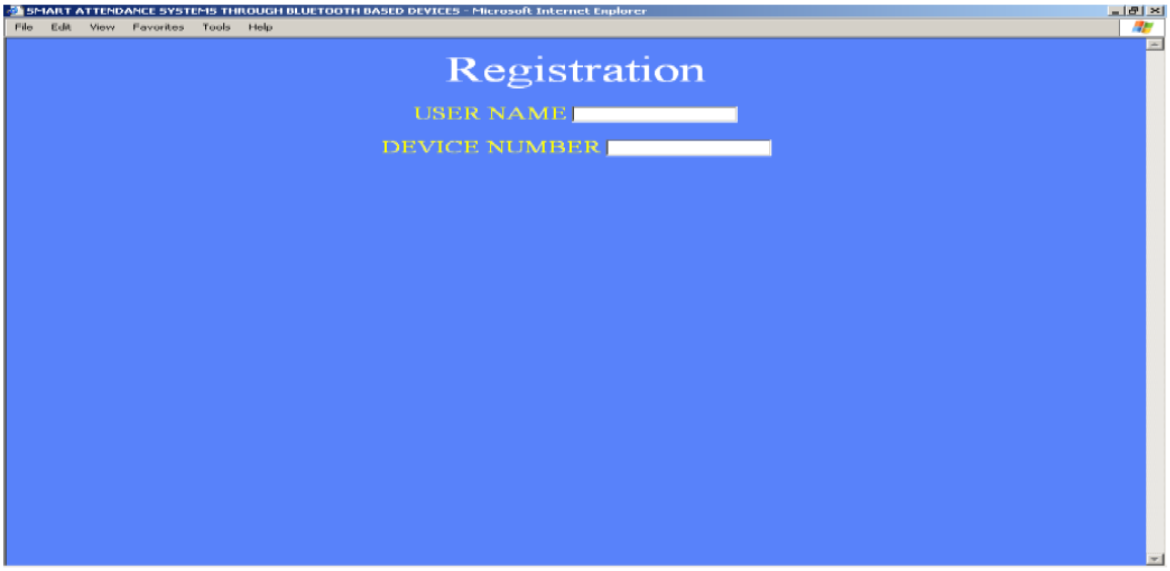

Figure 4: Device Registration

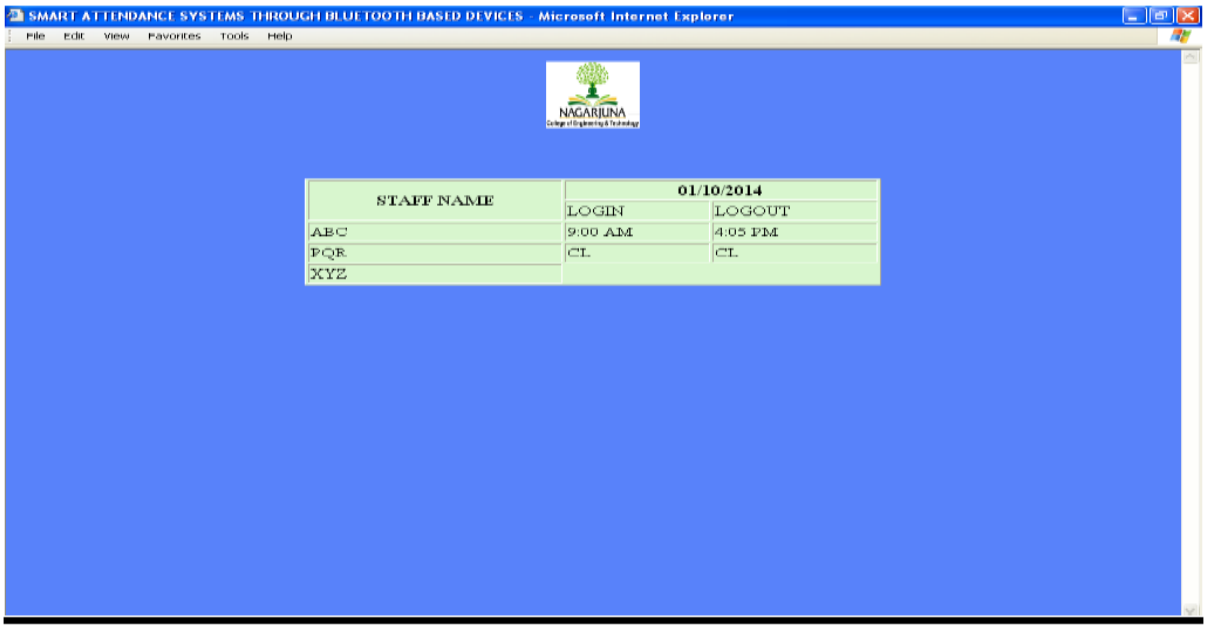

Figure 5: Smart Attendance System

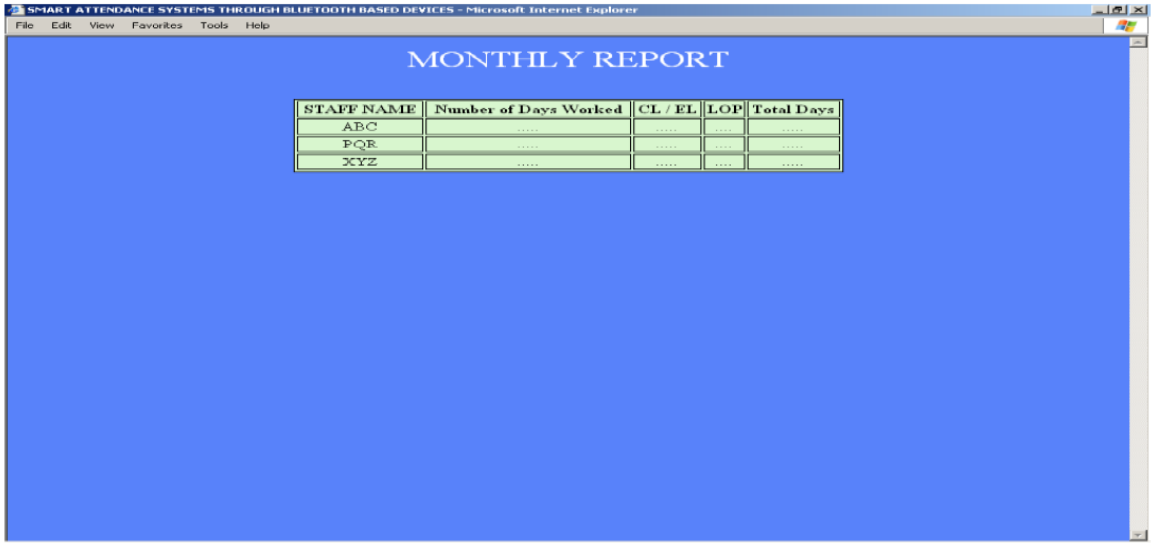

Figure 6: Monthly Report

REFERENCES

[1]. Forouzan: Data Communications and Networking, Fourth Edition III. Data Link Layer 14. Wireless LANs The McGraw−Hill Companies, 2007

[2]. William Stallings Wireless Communications And Networks Second Edition Pearson Prentice Hall Pearson Education, Inc. 2005,2002

[3]. An Automated Biometric Attendance Management System with Dual Authentication

Mechanism Based on Bluetooth and NFC Technologies Samuel King Opoku Computer Science Department, Kumasi Polytechnic, Ghana

[4]. Bluetooth Based Attendance Management System Vishal Bhalla, Tapodhan Singla,Ankit Gahlot Vijay Gupta Department of Computer Science And Engineering Dronacharya College of Engineering, Gurgaon, Haryana, India

[5] Mobile Based Attendance Marking System Using Android And Biometrics Mr. Gautam Shanbhag, Mr. Hussain Jivani , Mr. Sushil Shahi3, Bachelors of Engineering Department of Computers Rajiv Gandhi Institute of Technology, Mumbai

[6] Being There with Bluetooth Vesa A. Korhonen Department of Mathematical Information Technology University of Jyväskylä, Jyväskylä, Finland

[7]. Electronic Student Attendance Recording System**,**Dr. AkramAbedelKarim Ahmad AbdelQader1 , Dr. Mohammed Abbas Fadhil Al-Husainy2, Department of Multimedia Systems, Faculty of Sciences and Information Technology,Al-Zaytoonah University of Jordan. Amman-Jordan