M-BOARD IN AN AD-HOC NETWORK ENVIRONMENT

Sharon Panth[+]
Shri M. & N. Virani Science College, Yogidham Campus,
Rajkot, Gujarat 360005, India
sharon.panth20@gmail.com

Mahesh Jivani[*]
Department of Electronics, Saurashtra University, Saurashtra University Campus,
Rajkot, Gujarat 360005, India
mnjivani@gmail.com

Abstract

Notice Board is very essential part of any organization. This paper presents the design and implementation of M-Board (Mobile Notice Board) for Ad-hoc Network Environment that can be established and made available for an educational or industry environment. The cost-free communication among the mobile phone clients and server takes place with the help of Bluetooth wireless technology. M-Board is particularly developed as an informative application environment to provide the basic information like daily events or timetable to the users. The design is based on the amalgamation of Java ME with other technologies like Java SE, Java EE, PHP and MySQL. The system is designed to provide simple, easy-to-use, cost-free solution in a ubiquitous environment. The system design is easily implemented and extensible allowing the number of clients in Personal Area Network (PAN) for information exchange with the hotspot-server.

***Keywords*:** M-Board, Ad-hoc, Ubiquitous, Bluetooth, mobile, Java ME, PAN

1. Introduction

In the present era of Information age, there has been tremendous increase in demand for instant access to information. The easy access of mobile devices, their global presence and their adaptation to wireless ad-hoc networks have made revolutionary changes in the field of Mobile Ad-hoc Networks (MANETs) [Ramanathan and Redi (2002)]. They also facilitate cost free connectivity to an ad-hoc network using the popular wireless technology, Bluetooth [Frodigh *et al.* (2000)]. Mobile phone application development area awaits a great number of applications fulfilling diverse needs to facilitate the user. To program such ad-hoc environment for grabbing the opportunity of connectivity facilitated by the mobile devices to the ad-hoc network is a need of the time. Sun Microsystems, now Oracle, provides a great platform for the development of mobile and server applications to build an interactive ad-hoc network system [Zhang *et al.* (2006)]. The connectivity options provided includes the best available technology which is unlicensed and operates on 2.4 GHz globally operating frequency, Bluetooth. It provides the link to connect the nearby devices by a range up to 300 meter at the speed of 1 Mbps and more. Its availability in almost all kind of pervasive mobile devices provides simple, secured communication option where users utilize the fixed network services in a PAN (Personal Area Network) an ad-hoc network created using Bluetooth, while retaining the freedom of mobility. The strong hold of Bluetooth is that it provides data and voice transmission efficiently in ad-hoc environment which is very much advantageous in ad-hoc network to allow the mobile devices to participate in the communication process as and when required. This is very much useful for building a kiosk type application which can be accomplished by building networks of access ports. The ad-hoc network will then act as a fixed network extension, through which an interactive network service can be offered. Thus, it provides the information required by the user at any pace, any time within the network system. The proposed system included the Bluetooth access hotspot - server which is designed to be installed for any public or private area, and it will send out the content application to nearby mobile devices, free for the user over Bluetooth, once the device gets connected, device will be a part of ad-hoc network and the user will be able to interact with ad-hoc network using received content application. This hotspot will be managed centrally, for uploading content to be distributed by the hotspot, and to view statistics of the connected users on the proposed ad-hoc network. The M-Board Ad-hoc Environment was developed based on the following system architecture.

2. System Architecture

The system is mainly composed of the central data server, Bluetooth Modem and Bluetooth enabled mobile phone. The Bluetooth communication between the data server and mobile device is realized by Bluetooth Ad-hoc network. The system architecture is shown in the Fig. 1.

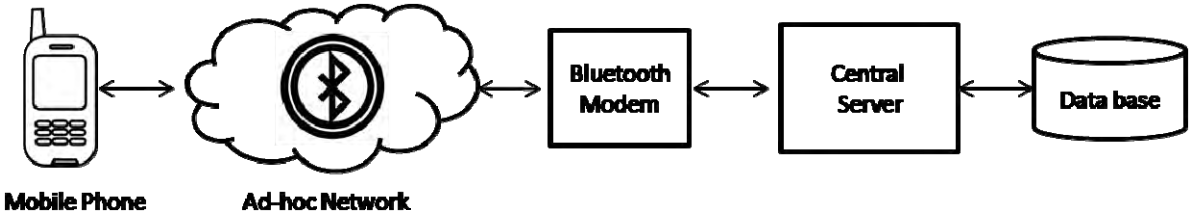

Fig. 1. System Architecture

The Central data server can be implemented by using ordinary computer server, workstations and so on. A mobile phone having the Bluetooth connectivity can connect to the server when it comes in its proximity. The communication channel is established over these Bluetooth enabled devices and communications can take place between these devices.

3. Technological Exploration

A number of technologies are used for the basis of the development scenario of M-Board in Ad-hoc Environment. The following subdivisions of this section give a clear idea of the utilization and role of the various technologies in the overall development of the application. The M-Board in Ad-hoc Environment is divided into two parts: Server-side Application and Client-side Application. This section describes the technologies responsible for the evolution of the server side and client side of the M-Board Application System. The Client-side application development presented in this paper includes the following technologies: Java ME [Muchow (2001)] Bluetooth [Haartsen (1998)], JABWT [Yang *et al*. (2010)], Networking implementation in Java ME [Harold (2004)] and XML [Tekli *et al.* (2009)] and kXML2 [Mnaouer (2004)]. The server-side application development includes technologies as mentioned in the client side environment. Along with the client side technologies, the server side development also includes Java SE (Java Platform, Standard Edition) and Java EE (Java Platform, Enterprise Edition), BlueCove [Zhang *et al.* (2006)], MySQL and PHP.

3.1. *Common technologies at Client and Server side of M-Board*

The commonly used technologies for the M-Board client and server applications are described in the following sections:

3.11 Java ME

Java ME (Java Platform, Micro Edition) is the edition of Java "Java for Mobile Devices". Java ME enables the developers to design and build mobile applications for the target consumer electronics and embedded devices. These applications suit the needs of a variety of information devices. It is a challenge for the developers to make compelling content for small and resource-constrained devices. This helps the service providers to provide content to their customers over those devices.

Java ME combines the functionality of the Java APIs and the reduced version of Java Virtual Machine. Java ME architecture consists of the Configuration, Profile and other optional packages. A Configuration includes the Java Programming Language features, a set of Java Virtual Machine features and some set of Java APIs. There are two types of Configurations: CDC (Connected Device Configuration) and CLDC (Connected Limited Device Configuration). A Profile can be termed as an extension to the Configuration. It consists of the libraries for the developers for the creation of mobile applications for the mobile devices. There are different types of Profiles like Mobile Information Device Profile (MIDP), Foundation Profile, PDA Profile, Personal profile and many more which work either with CDC or CLDC type of Configurations.

The MIDP (Mobile Information Device) Profile provides the Java runtime environment and a set of Java APIs when combined together with the CLDC Configuration for small computing devices. MIDP and CLDC combine

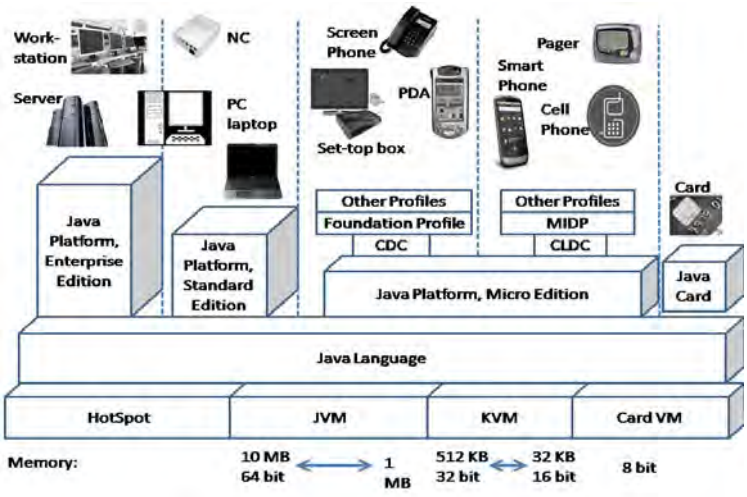

Fig. 2. The Java target market with its three editions and Java ME architecture

together for writing applications for the Mobile Information Devices (MIDs). MIDP, JSR-118 [Muchow (2001)] with CLDC is used for the current mobile application development. The "Write Once, Run Anywhere" term of Java still conforms for the quick deployment of these applications on a variety of mobile information devices that supports it. Java ME MIDP mobile devices were launched in April 2001, since then there has been rapid increase in the number of such devices leading to the growth of the mobile application development area.

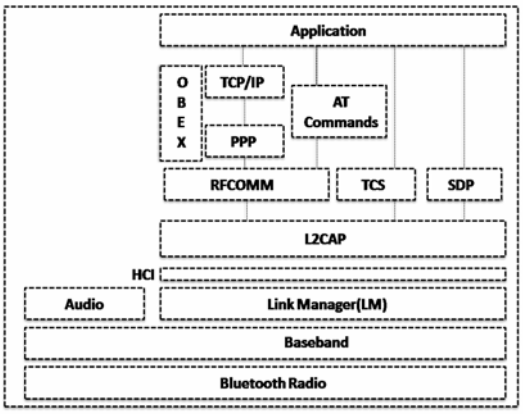

Fig.3. Bluetooth Protocol stack

3.1.2 Bluetooth

Bluetooth is a short-range wireless technology standard for radio transmissions between 2400–2480 MHz created by Telecom vendor Ericsson in 1994 [Haartsen (1998)]. This technology allows the creation of small, secured networks called the Personal Area Networks (PANs) and became an alternative to the RS-232 data cables. This standard is managed by the Bluetooth Special Interest Group, since 1998.

Almost all the mobile devices embed small, low-power and cheap integrated chips in different electronic devices that work as short-range radio transceivers for the radio communications over Bluetooth. The popularity of Bluetooth technology also depends upon its security mechanisms provided using device pairing, authentication, encryption and authorization techniques.

The MIDP support and the Bluetooth technology in the mobile devices gives a completely different alternative to create and deploy the mobile applications over the Mobile Information Devices (MIDs) with almost less or no cost.

Bluetooth works using a number of protocols as defined in its layered architecture. The Bluetooth protocol stack or the architecture [Singh P. *et al.* (2011)] as shown in the figure below defines the following protocols which can be used in the application.

- LMP: The Link Management Protocol (LMP) establishes and controls the links between the devices on the controller.
- L2CAP: The Logical Link Control and Adaptation Protocol (L2CAP) maintain multiple logical connections between devices and allow assembling and disassembling of packets during communication.
- SDP: SDP is the Service Discovery Protocol that provides service discovery for the device.
- RFCOMM: RFCOMM is the Radio Frequency Communication protocol, the replacement for the RS-232 cable to provide serial emulation.
- OBEX: OBEX (OBject EXchange) is an adopted session layer protocol for object exchange between the devices.

The Bluetooth profiles [Singh P. *et al.* (2011)] are the implementation of the Bluetooth protocols in full or partial manner defined and adopted by the Bluetooth SIG. The device manufacturers can maintain compatibility in their devices regarding the Bluetooth technology and to provide the services using the support of these profiles in there devices. Hence, a Bluetooth profile resides on the Bluetooth protocol stack and optionally requires the support of other protocols.

The Bluetooth profile used in M-Board client-side mobile phone application is the Bluetooth Serial Port Profile (btspp) [Singh P. *et al.* (2011)]. RFCOMM is a connection-oriented protocol. It provides streaming communication between the devices. The btspp profile and RFCOMM protocol are used in the application to access the serial port and communicate using streaming data.

3.1.3 JABWT

The Bluetooth standard is defined in Java ME with the help of the Java APIs for Bluetooth Wireless Technology (JABWT). It is built on top of the Bluetooth stack as shown in the figure below. In Java ME the Bluetooth API and OBEX API [Auletta *et al.* (2008)] are defined in the JSR-82 [Auletta *et al.* (2007)] specification. The Bluetooth API provides the standard Bluetooth functionality in Java ME for creating Bluetooth based applications. The OBEX API allows the MIDLETS to exchange different objects like files using the services provided by the devices.

In the M-Board mobile phone application and the server application the support of JABWT is taken so that the Bluetooth APIs can be used for the development. It also uses the OBEX API for exchanging different objects.

3.1.4 Networking in Java ME

Networking in Java ME applications can be implemented with the help of the standard Java package java.io [Roldán (2006)] with the Java ME package javax.microedition.io [Roldán (2006)]. The framework to support Networking for mobile devices on top of the basic Java Networking support is known as Generic Connection Framework (GCF).

In the M-Board Ad-hoc Environment we communicate the devices in a wireless Ad-hoc network known as a MANET created using Bluetooth technology which can be called as a Piconet. This requires the support of Networking in Java ME in the application. So, in M-Board environment both the client-server applications are built with the support of GCF which defines the Networking support in Java ME.

3.1.5 XML and kXML2

Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML (ISO 8879). It is designed to meet the challenges of large-scale electronic publishing. XML is also playing an important role in the exchange of a wide variety of data on the Web and elsewhere.

kXML2 is a BSD-licensed small XML pull parser specially designed for constrained environments, to access, parse, and display XML files for Java Platform, Micro Edition (Java ME)-enabled devices. This kXML version is based on the common XML pull API. The M-Board mobile application retrieves the XML data from the XML file received in the mobile phone from the M-Board server using the kXML2 and xmlpull APIs. The M-Board server application uses PHP to store data retrieved from MySQL database in an XML file. Finally this file is sent to the mobile client.

3.2. ***Other Server side technologies***

3.2.1 Java SE

Java SE (Java Platform, Standard Edition) is the core java technology that is used to build and deploy desktop and server applications as shown in the figure 2. It provides rich user interface, performance, versatility,

portability, and security for the modern application development environment. For building the M-Board server application, the core java language support is required.

3.2.2 Java EE

Java EE (Java Platform, Enterprise Edition) is the enterprise level standard for Java edition for enterprise applications as shown in the figure 2. It provides a runtime environment and APIs for developing and deploying such applications, including network and web services, and other large-scale, multi-tiered, scalable, reliable, and

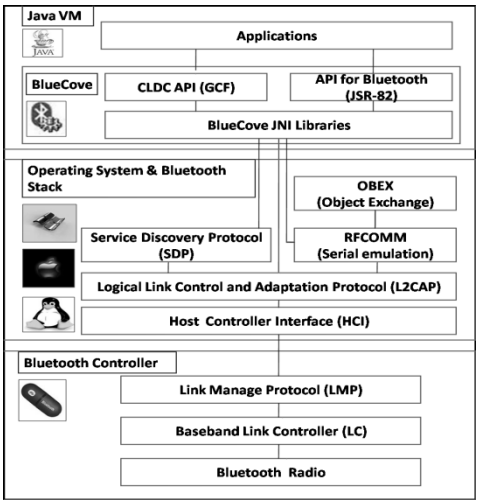

Fig. 4. BlueCove architecture

secure network applications.

The M-Board server application uses different Java EE facilities like to build the user interface, database connectivity with MySQL database, etc.

3.2.3 BlueCove

BlueCove is a Java ME library designed to implement Java APIs for Bluetooth JSR-82 J2SE implementation that to interface with the Mac OS X, WIDCOMM, BlueSoleil and Microsoft Bluetooth stack found in Windows XP SP2 and newer. Originally it is developed by Intel Research and currently maintained by volunteers.

So, for the M-Board server application the BlueCove library is used for providing the JSR-82's J2SE support as shown in the figure 4.

3.2.4 MySQL

MySQL is an open source Relational Database Management System. It is very commonly used in conjunction with PHP scripts to create powerful and dynamic server-side applications. M-Board Environment also uses MySQL **database** to store and retrieve information for database related operations.

3.2.5 PHP

PHP (Hypertext Preprocessor) is an open-source server-side scripting language designed for Web development to produce dynamic Web pages. M-Board Environment uses PHP to build the M-Board server side application environment.

4. M-Board Ad-hoc Environment

The M-Board ad-hoc environment includes server, client and connection environment. The M-Board server application solution developed using the amalgamations of the three Java Editions Java SE, Java EE and Java ME as well as Bluetooth for establishing the wireless connection and MySQL with PHP for database and server side scripting. For server-side development the Bluetooth APIs are used by implementing the JSR-82 library for J2SE called BlueCove.

4.1. *Server*

The M-Board Server application launches the device discovery process first. This is one of the methods of discovering the devices in a Bluetooth based communication in an ad-hoc network. If the discovery process is successful, it lists number of mobile phones which are discovered in its immediacy and sends the M-Board

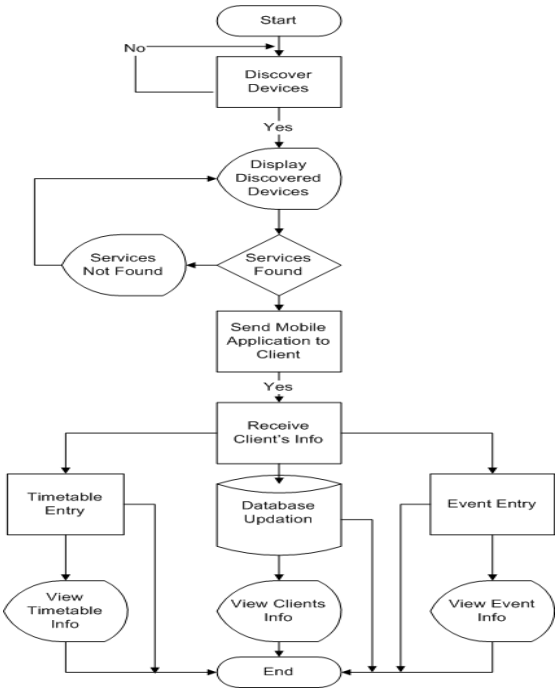

Fig. 5. The flow chart of the M-Board Server Environment

mobile application to the clients that are successfully discovered and having the OBEX service for receiving the mobile application packaged in a jar file. For unsuccessful attempts the server displays related messages. When the client receives the file, it is installed or opened directly according to the phone implementation. The server starts receiving from the client. The client then fills the details and sends the data to the server. The server stores the client information. It also allows viewing all the clients and their detailed information. The server application provides the facility to add, search and view Events and Timetable information which is to be shown to the client along with the updated Events and Timetable information. The server is notified when the client exits the application. The server can resume back the process or exit to shut down the network.

4.2. *Client*

A mobile phone having the Bluetooth connectivity can connect to the server when it comes in its proximity. The communication channel is established over these Bluetooth enabled devices and communications can take place between these devices. The RFCOMM protocol adapts only a part of the ETSI TS 07.10 standard. It is built on top of the L2CAP protocol, as shown in the Bluetooth protocol stack in the figure 3. RFCOMM allows the device manufacturers to easily extend the serial port functionality of their Bluetooth devices. It allows up to 60 concurrent connections between the devices; the implementation is device-specific generally 30 active connections. Any one of the services can be assigned to a single client connection at a time using this protocol. Finally, only one RFCOMM session can be handled between two devices. The implementation hence uses the combination of the Bluetooth serial port profile and the RFCOMM protocol by implementing JABWT with Networking in Java ME, respectively. Finally, the data represented to the user in the client side is done using the XML implementation in Java ME by kXML2.

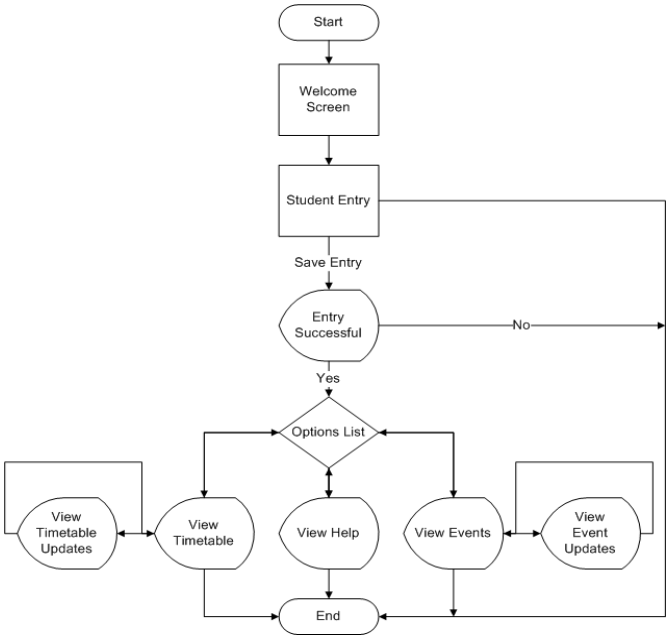

Fig. 6. The flow chart of the M-Board Client Environment

Hence, the mobile phone application combines the aforementioned technologies as per the requirement for its implementation. The mobile phone application works on the Java ME enabled phone with MIDP specification. It connects to the M-Board server system which is built using the mentioned technologies.

4.3. *Connection*

Generally, there are two methods to establish connection between the communicating devices using Bluetooth. First is the **Discovery method** where the device discovers other Bluetooth enabled devices. In second case, there is the **Known method** that allows direct connection to be established between already known remote devices.

M-Board Ad-hoc environment works using both the methods. The M-Board server system uses the Discovery method to discover the mobile phones that comes under its scrutiny. At client side, there is no need to use the first method as the target device is the known device which is the server, so the server's device address can be used directly to avoid unnecessary delay in communication in an ad-hoc environment.

The advantage of using the second method at client side is that the connection is established directly by specifying the known device's address within very less time as compared to the first method where it searches for Bluetooth enabled devices first in its proximity and then establishes connection, if wanted.

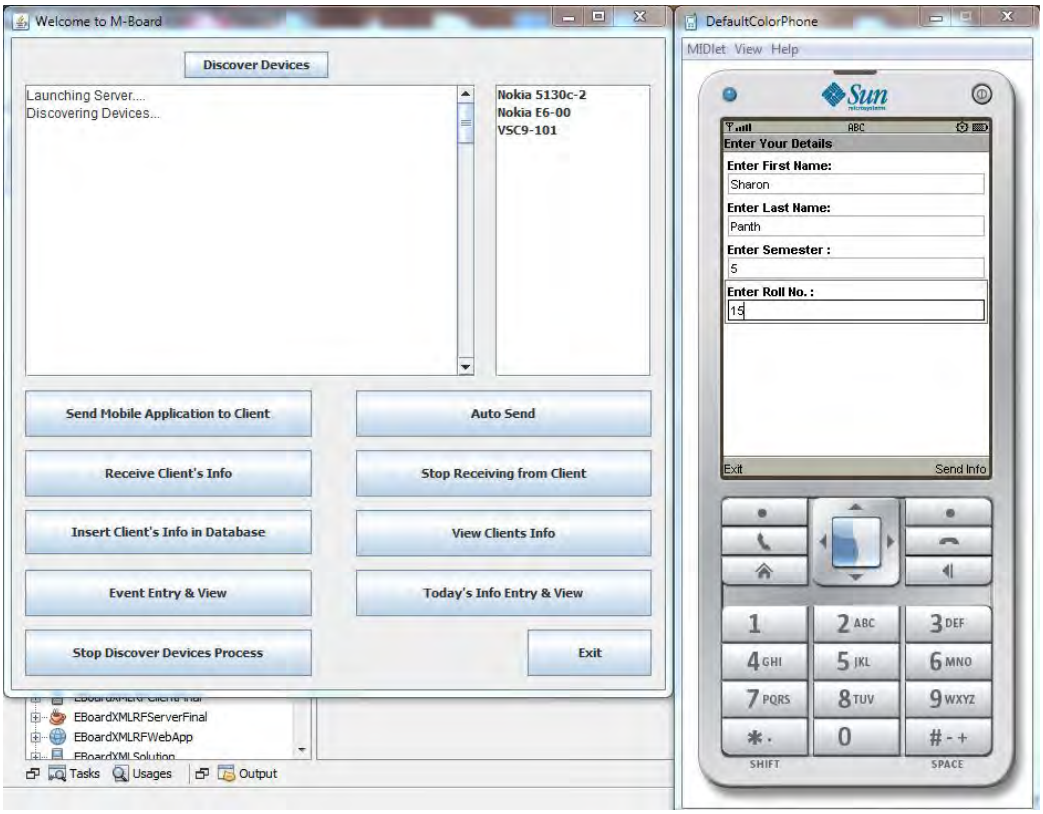

Fig. 7. Server and Client Screen Shot

The JABWT implementation includes the support for btspp protocol over RFCOMM service. For the communication between the mobile phone and the server system the connection string contains the btspp protocol in the URL. This is implemented by using the Connector class in the GCF Hierarchy. The Connector class creates new connections dynamically using the protocol implementation class formed from platform name (from system property) and protocol name (from the parameter string provided by the developer).

The parameter string that describes the target takes the general form: {scheme} :[{target}][{params}] . It conforms to the URL format. This form describes the {scheme} as the name of a protocol such as http, the {target} as some kind of network address and {params} formed as a series of equates of the form "; x=y".

The Connector class open method has a parameter of connection string. It is casted as per the requirement of the connection in the application. The connection string used in the mobile phone application is in the format of the following: **btspp://address:port.**

Where, btspp:// is the Bluetooth Serial Port Profile,

address is the Bluetooth known remote device address of 12 digit hexadecimal format,

port is the port number or the communication channel on which the remote device receives data.

For example: **btspp:// 00265EE4E865:11**

So, in the application the server side address is manually entered in the client program in the connection string as shown in the above example. The client connects to the server using this server address and a service port. The port number is assigned dynamically by JSR-82, which in this case is '11'.

5. Implementation

The user must have a mobile phone with MIDP support. The application is automatically transferred to the phone by the server packaged in a jar file and gets installed as per the phone's default location mentioned by the implementation which is used to store the jar file. When user launches the application it displays the splash screen and moves to the first screen. This screen has an option to exit the application or to enter the details of the

client like first name, last name, and semester and roll number and connect to the already enabled server system using the serial port for communication. When the user attempts to connect to the server system and if the

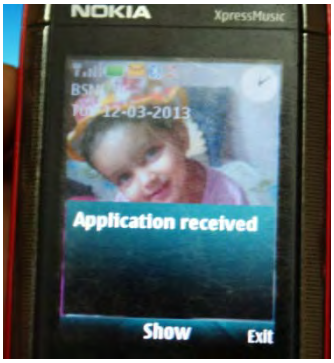

Fig. 8. Screen Shot of Client Mobile

connection is successful then the client application gives appropriate message and redirects to a list of options. The list has choices to view Event and Timetable information along with Help option. The user can choose to exit the application from this step or move further to choose Event or Timetable option. If any of the Event or Timetable option is chosen by the user than it moves to their respective screens. The Event screen contains a menu that has options to view Event information, to view updated event information, if any, available during the current session and finally to move back to the previous screen containing the list. In the same way, the Timetable screen contains a menu that has options to view Timetable information, to view updated timetable information, if any, available during the current session and move back to the previous screen containing the list. The user can move back anytime and can finally exit the application from the exit option in the option list screen.

6. Conclusion

Finally, the M-Board Ad-hoc environment presents a priceless solution to establish Bluetooth connections using the btspp protocol over Bluetooth channels for Java ME enabled mobile phones. The resultant solution is lightweight with no extra cost for the mobile users and application developers, but it only requires the presence of a Bluetooth connection between the mobile devices i.e. the client and the server i.e. workstation.

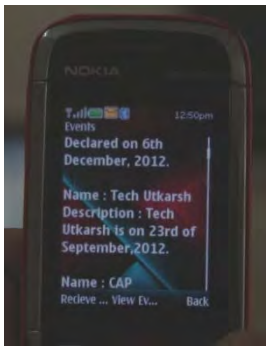

Fig. 9. Screen shot of client mobile: Event is in progress

The client-server implementation is flexible allowing programmers for the enhancement of the features of the client and the server application with basically no effort for inexpensive mobile phones provided with Bluetooth interface, which is free to use. The test scenario as shown in the figure 8 and 9 proves the real time application of our solution. This verification shows the usefulness and efficiency of the application in a live environment suitable enough for the development of a similar number of applications.

REFERENCES

[1] Auletta V.; Blundo C.; De Cristofaro E., (2007). *A J2ME transparent middleware to support HTTP connections over Bluetooth*, ICSNC, ISBN: 0-7695-2938-0.

[2] Auletta V.; Blundo C.; De Cristofaro E., (2008). *HTTP over Bluetooth: a J2ME experience*, International Journal on Advances in Telecommunications, **1**(1), pp. 57-66.

[3] Frodigh M.; Johansson P.; Larsson P., (2000). *Wireless ad hoc networking—The art of networking without a networ*k, Ericsson Review.

[4] Haartsen, J., (1998). *Bluetooth: The universal radio interface for ad hoc, wireless connectivity*, Ericsson review.

[5] Harold E. R., (2004). *Java Network Programming third edition*, O'Reilly Media.

[6] Mnaouer, A.B.; Shekhar, Anand; Liang, Zhao Yi., (2004), *A generic framework for rapid application development of mobile Web services with dynamic workflow management*, IEEE International Conference on Services Computing.
[7] Muchow J.W., (2001). *Core J2ME Technology and MIDP*, Prentice Hall, ISBN: 0130669113.
[8] Ramanathan, R.; Redi, J., (2002). *A brief overview of ad hoc networks: challenges and directions*, IEEE communications Magazine.
[9] Roldán E.; Montón E.; Guillén S., (2006). *Programming mobile devices with J2ME***,** Workshop on technologies for health, ITACA.
[10] Singh P.; Sharma S.; Agrawal S., (2011). *Study of Bluetooth wireless technology using java*, Indian Journal of computer Science and Engineering (IJCSE), **2**(3), pp. 295-307.
[11] Tekli J.; Chbeir R.; Yetongnon K., (2009). *An overview on XML similarity: background, current trends and future directions*, Elsevier publication, Computer Science Review, **3**(3), pp. 151-173.
[12] Yang, SY; Hsu CL; Lee DL, (2010). *An ontology-supported ubiquitous interface agent for cloud computing – example on Bluetooth wireless technique with java programming*, Proceedings of the 9th International Conference on Machine Learning and Cybernetics, Qingdao.
[13] Young B. K.**;** Vaidya N. H., (2000). *Location-Aided Routing (LAR) in mobile ad hoc network*, Wireless Networks, Springer, ISSN: 1022-0038, **6**(4), pp. 307-321.
[14] Zhang Y.; Zhang S.; Vuong S.; Malik K., (2006). *Mobile learning with Bluetooth-based E-learning system*, Proceedings of the International conference on Wireless communications and mobile computing, pp. 951-956.