



Design and Implement Privet Chatting Over Bluetooth for Wide Area

Aysar A. Abdulrahman¹ , Karzan H. shareef² , Ahmed F. Fadhil³

¹University of Sulaimani / College of Science / Computer Dept.

aysser2005@yahoo.com¹

²University of Human Development / Computer Dept.

karzan.cs@gmail.com²

³Kirkuk University / College of Engineering / Electrical Engineering Dept.

amet83@yahoo.com³

Received date : 7 / 1 / 2015

Accepted date : 22 / 10 / 2015

ABSTRACT

The mobile applications have become one of the important applications in human life, and it was widely used during the last ten years. Mobile devices can be used to do various tasks, such as browsing the internet, sending or receiving emails, and performing other tasks. Bluetooth technology is considered as one of the important specifications that were added to mobile phones during the current century where the user can use it for data and image transmission and others. One of the important applications of Bluetooth, which appeared in the last years, is the conversation via Bluetooth where users can hold short conversations using this technique. One of the most important benefits of Bluetooth technology is that it's free to send and receive messages. While the disadvantages of this application is that it is short-term, since users must be within the scope of the Bluetooth service which is (7 - 9 meters) to be able to send messages and use it.

In this paper, we will design and implement a private conversation via Bluetooth software which has the ability to work within biggest scopeby taking advantage of the properties of the (Scatternet). Each user of the application can be a crossroads for the other users. In other words, if there is, for example, users (a) and (b) 7 meters away from each other, and a third user (c) appeared within 12 meters of the first user (a). In this case, the third user (c) can be connected to the first user through the user (b). In other words, the second user (b) would be an intermediary between the first user (a) and the third user (c). So, we have been solved the distance limit problem for using the Bluetooth application. In addition, the users

are now able to chat as a group (not just peer-to-peer) in a private room with a wide range using the piconet and scatternet characteristics. Netbeans 5.5, Java 2 Micro Edition (J2ME), Bluetooth API (JSR-82), Nokia Pc Suites are the tools and technologies that have been used through building this application.

Keywords: Bluetooth, Piconet, Scatternet, Netbeans

تصميم وتنفيذ برنامج محادثة جماعية خاصة باستخدام البلوتوث ولمدى اوسع

أيسر عبد الخالق عبد الرحمن^١ ، كارزان حسين شريف^٢ ، أحمد فريدون فاضل^٣

^١كلية العلوم / قسم الحاسبات / جامعة السليمانية

^١aysser2005@yahoo.com

^٢جامعة التنمية البشرية / قسم الحاسبات

^٢karzan.cs@gmail.com

^٣كلية الهندسة / قسم الكهرباء / جامعة كركوك

^٣amet83@yahoo.com

تاريخ قبول البحث: ٢٢ / ١٠ / ٢٠١٥

تاريخ استلام البحث: ٧ / ١ / ٢٠١٥

المخلص

اصبحت تطبيقات الهواتف النقالة اليوم من التطبيقات المهمة في حياة الانسان، ولقد لاقت انتشارا كبيرا خلال السنوات العشرة الماضية. حيث اصبح الانسان باستطاعته القيام بمهام مختلفة باستخدام الهاتف النقال كتصفح الانترنت وارسال واستلام الرسائل الالكترونية وغيرها من التطبيقات المهمة الاخرى. تعتبر تقنية البلوتوث من احدى المواصفات المهمة التي اضيفت للهواتف النقالة خلال القرن الحالي. حيث يمكن للمستخدم استخدامه في نقل البيانات والصور وغيرها. من احدى التطبيقات المهمة للبلوتوث والتي ظهرت خلال السنوات الاخيرة هو المحادثة عبر البلوتوث. حيث يمكن للمستخدمين اجراء محادثات قصيرة باستخدام هذه التقنية. ان من اهم الفوائد المهمة لتقنية البلوتوث انه مجاني اي لا يكلف استخدامه في ارسال واستلام الرسائل. ولكن المشكلة لذلك التطبيق هو انه قصير المدى، حيث ان المستخدمين يجب ان يكونوا ضمن نطاق خدمة البلوتوث والتي هي (٧-٩) متر ليكونوا قادرين على ارسال واستخدام الرسائل. في هذا البحث تم تصميم وتنفيذ برنامج محادثة خاصة عبر البلوتوث قادر على العمل ضمن نطاق ومدى اكبر. حيث تم توسيع مدى استخدام التطبيق من خلال الاستفادة من خصائص ال (Scatternet). حيث ان كل مستخدم للتطبيق يمكن ان يكون كنقطة وصل بين مستخدمين اخرين. اي بمعنى اخر، لو كان هناك على سبيل المثال مستخدمين (أ) و (ب) اثنين على بعد ٧ متر، وظهر مستخدم ثالث (ج) على بعد ١٢ متر من المستخدم الاول (أ). ففي هذه الحالة، فان المستخدم

الثالث (ج) يمكنه الوصول للمستخدم الاول من خلال المستخدم (ب). اي ان المستخدم الثاني (ب) سيكون وسيطا بين المستخدمين الاول (أ) والمستخدم الثالث (ج). وبذلك نكون قد تجاوزنا مشكلة المسافة المحددة لاستخدام هذا تطبيق ال (Bluetooth). ومن الفوائد الاخرى للتطبيق المصمم في هذا البحث هو ان المستخدمين قادرين على المحادثة الجماعية (اي انها لا تقتصر على استخدامها بين مستخدمين اثنين فقط) في غرفة خاصة مع مسافات واسعة باستخدام ال Piconet و scatternet characteristics . الادوات والتقنيات التالية (Java 2 Micro Edition J2ME), (Netbeans 5.5), (Bluetooth API JSR-82), (Nokia Pc Suites) هي التي استخدمت خلال بناء هذا التطبيق.

الكلمات الدالة: Bluetooth, Piconet, Scatternet, Netbeans.

1. INTRODUCTION

The mobile phone has become nowadays an inseparable part of our daily life. Considerably surpassing in number the amount of desktop computers, mobile phone technology has met great improvements in the last couple of years. It's use is not limited to communication, but can also serve for numerous other purposes whether for business , checking emails, paying for tickets, sending money transfers, taking pictures of your kids, watching soccer World Cup live, checking commodity pricing, and to emergency response to mobile Health. Mobile devices have become an essential tool to help us navigate our day [1]. The Mobile Operating System is responsible for managing physical resources, power consumption and communication interface. A mobile operating system is composed of the computer kernel, the power management facility, and the real-time kernel. The computer kernel is the entity that manages access to physical resources, such as CPU and disk space. The power management facility is responsible for reducing power consumption. The real-time kernel is responsible for managing the communication link [2]. The mobile application just like any other java application .It is a set of compiled class files. The basic outline of a mobile application is pretty much dedicated by the runtime system requirement and the API concepts (application programming interface, is a source code interface that an operating system , library or services provides to support request made by computer programs).As shown in Fig.(1) an application can be in the following states [3]:

1-Active: The application is running and may use system resources.

2-Paused: The application is temporarily paused because another application needs all resources, e.g. an incoming phone call.

3-Destroy: The application has been terminated and all resources except for persistent memory need to be cleared.

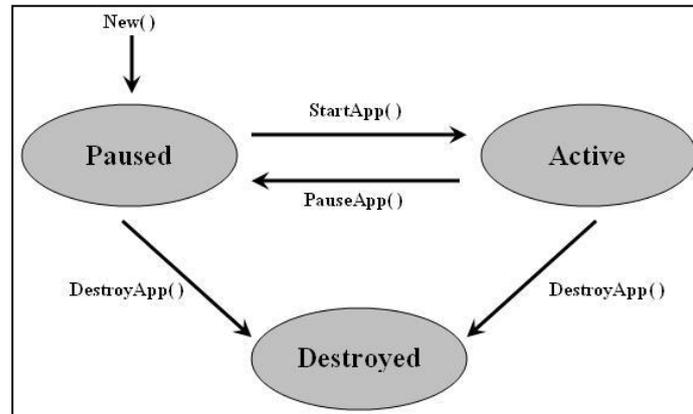


Fig. (1): An application States

2. MOBILE BLUETOOTH

Bluetooth is a short-range radio technology that enables wireless connectivity between mobile devices. It was initially developed by Swedish mobile phone maker Ericsson in 1994 as a way to let laptop computers make calls over a mobile phone. Since then, several companies have signed on to make Bluetooth the low-power short-range wireless standard for a wide range of devices.

The three main goals for Bluetooth were: small size, minimal power consumption, and low price [4]. The technology was designed to be simple, and the target was to have it become a standard in wireless connectivity. Bluetooth radio operates in the unlicensed Industrial Scientific and Medical (ISM) band at 2.4 GHz. In some countries, this band is reserved for military use, but these countries have now begun freeing that band for general use. The maximum gross data rate is 1 Mbps. The range of Bluetooth depends on the power class of the radio.

The devices which use a radio communications system, they do not have to be in line of sight of each other; they can even be far apart if the transmission has sufficient power as shown in Table (1) .

Table (1): The Bluetooth Classes [5]

Class	Maximum Permitted Power mW (dBm)	Range (approximate)
Class 1	100 mW (20 dBm)	~100 meters
Class 2	2.5 mW (4 dBm)	~10 meters
Class 3	1 mW (0 dBm)	~1 meter

Most devices are expected to use the class 2 radio that provides 4 dBm nominal output power, resulting in a range of up to 10 meters in an obstacle-free environment. This range is sufficient for cable-replacement applications. When a longer range is needed (e.g., in access points), a more powerful radio (class 1) can be used. Larger power consumption is not a problem if the device is a piece of fixed equipment. With mobile devices such as mobile phones, power-consumption issues are crucial and therefore class 2 is the only feasible option. The Bluetooth system consists of a radio unit, a link control unit, and a support unit for link management and host terminal interface functions see Fig.(2) . The Host Controller Interface (HCI) provides the means for a host device to access Bluetooth hardware capabilities. For example, a laptop computer could be the host device and a PC card inserted in the PC is the Bluetooth device. All commands from the host to the Bluetooth module and events from the module to the host go through the HCI interface. The protocol stack is above the radio and baseband hardware, partly residing in the Bluetooth unit and partly in the host device.

A Bluetooth solution can also be implemented as a one-processor architecture (embedded solution) where the application resides together with the Bluetooth protocols in the same hardware. In that case, the HCI is not needed. This is a feasible implementation for simple devices such as accessories or micro servers [6].

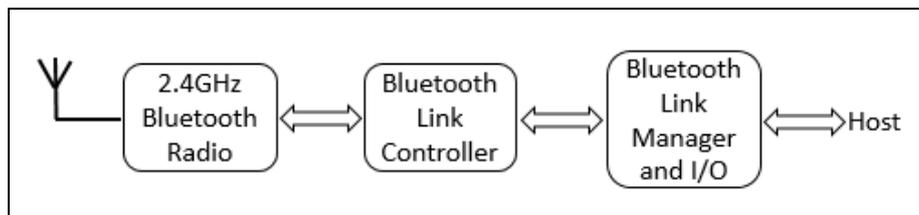


Fig. (2): Bluetooth System

3. PROBLEM DESCRIPTION

The Bluetooth specification aims to allow Bluetooth devices from different manufacturers to work with one another, so it is not sufficient to specify just a radio system. Because of this, the Bluetooth specification does not only outline a radio system but a complete protocol stack to ensure that Bluetooth devices can discover each other, explore each other's services, and make use of these services. The Bluetooth stack is made up of many layers, as shown in Fig.(3) .

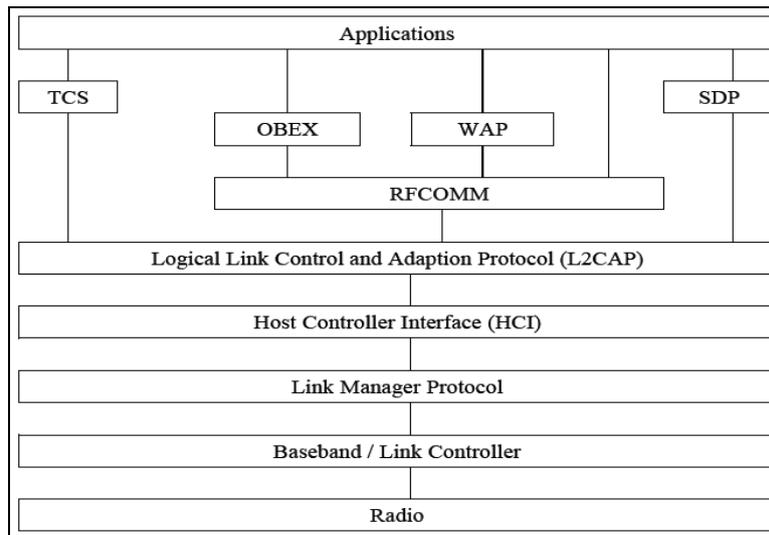


Fig. (3): The Bluetooth protocol stack [7]

The Host Controller Interface (HCI) is usually the layer separating hardware from software and is implemented partially in software and hardware/ firmware. The layers below the HCI are usually implemented in hardware and the layers above the HCI are usually implemented in software. Note that resource constrained devices such as Bluetooth headsets may have all functionality implemented in hardware/firmware. Table (2) gives a short description of each layer shown in the Fig. (3) :

Table (2): Descriptions of Bluetooth protocol layers [8]

Layer	Description
Applications	Bluetooth profiles guide developers on how applications should use the protocol stack
Telephony Control System (TCS)	Provides telephony services
Service Discovery Protocol (SDP)	Used for service discovery on remote Bluetooth devices
WAP and OBEX	Provide interfaces to higher layer parts of other communications protocols.
RPCOMM	Provides an RS-232 like serial interface.
L2CAP	Multiplexes data from higher layers and converts between different packet sizes.
HCI	Handles communication between the host and the Bluetooth module
Link Manager Protocol	Control and configures links to other devices
Baseband and Link Controller	Controls physical links, frequency hopping and assembling packets.
Radio	Modulates and demodulates data for transmission and reception on air.

Application developers do not need to know all the details about the layers in the Bluetooth stack. However, an understanding of how the Bluetooth radio works is of importance. The Bluetooth radio is the lowest layer of Bluetooth communication. The Industrial, Scientific and Medical (ISM) band at 2.4 GHz is used for radio communication. Note that several other technologies use this band as well. Wireless fidelity (Wi-Fi) technologies like IEEE 802.11b/g and kitchen technologies like microwave ovens may cause interference in this band. The Bluetooth radio utilizes a signaling technique called Frequency Hopping Spread Spectrum (FHSS). The radio band is divided into 79 sub-channels. The Bluetooth radio uses one of these frequency channels at a given time. The radio jumps from channel to channel spending 625 microseconds on each channel. Hence, there are 1600 frequency hops per second. Frequency hopping is used to reduce interference caused by nearby Bluetooth devices and other devices using the same frequency band. Adaptive

Frequency Hopping (AFH) is introduced in the Bluetooth 1.2 specification and is useful if your device communicates through both Bluetooth and Wi-Fi simultaneously (e.g. a laptop computer with both Bluetooth and Wi-Fi equipment). The frequency hopping algorithm can then avoid using Bluetooth channels overlapping the Wi-Fi channel in use, hence avoiding interference between your own radio communications. Every Bluetooth device is assigned a unique Bluetooth address, being a 48-bit hardware address equivalent to hardware addresses assigned to regular Network Interface Cards (NICs). The Bluetooth address is used not only for identification, but also for synchronizing the frequency hopping between devices and generation of keys in the Bluetooth security procedures [9]. Last seven years, many applications have been developed by using Bluetooth technology. One of those applications was the chatting over Bluetooth peer-to-peer. The limitation of the application was the short range of it, which means, the users should be within the range of the Bluetooth, about 7-9 meter, to be able to send and receive messages [10].

4. Proposed Approaches by Using Piconet and Scatternet Characteristics

A piconet is the usual form of a Bluetooth network and is made up of one master and one or more slaves. The device initiating a Bluetooth connection automatically becomes the master. A piconet can consist of one master and up to seven active slaves. The master device is literally the master of the piconet. Slaves may only transmit data when transmission-time is granted by the master device, also slaves may not communicate directly with each other and all communication must be directed through the master. Slaves synchronize their frequency hopping with the master using the master's clock and Bluetooth address.

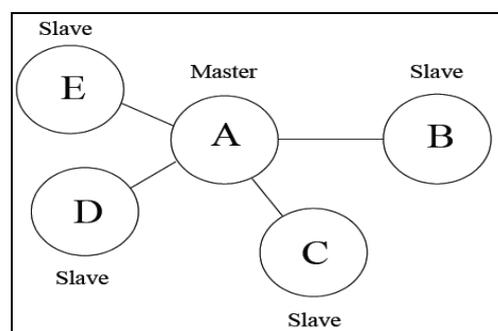


Fig. (4): A typical piconet

Piconets take the form of a star network, with the master as the center node as shown in Fig. (4) . Two piconets may exist within radio range of each other. Frequency hopping is not synchronized between piconets, hence different piconets will randomly collide on the same

frequency. When connecting two piconets the result will be a *scatternet*. Fig. (5) shows an example, with one intermediate node connecting the piconets. The intermediate node must time-share, meaning it must follow the frequency hopping in one piconet at the time. This reduces the amount of time slots available for data transfer between the intermediate node and a master; it will at least cut the transfer rate in half. It is also important to note that neither version 1.1 nor version 1.2 of the Bluetooth specification defines how packets should be routed between piconets. Hence, communication between piconets cannot be expected to be reliable.

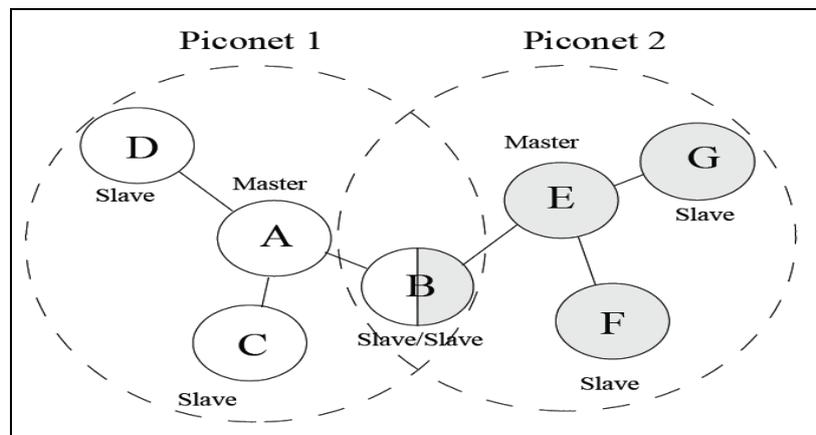


Fig. (5): Scatternet

Role-switching enables two devices to switch roles in a piconet. Consider the following example: You have two devices A and B. Device A connects to device B, hence, device A becomes the master of the piconet consisting of devices A and B as shown in the Fig. (6) :

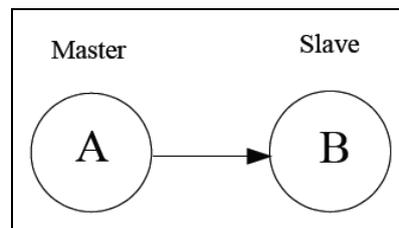


Fig. (6): Piconet with two nodes

Then a device C wants to join the piconet. Device C connects to the master device, A. Since device C initiated the connection it will automatically become the master of the connection between device C and device A. We now have two masters; hence, we have two piconets. Device A is the intermediate node between these piconets, being the master for device B and the slave for device C, as seen in Fig. (7) . Fig. (8) shows that a role-switch

between device A and device C will give us one piconet where A is the master and both B and C are slaves. We see that when a new device wants to be part of a piconet we actually need a role-switch to make this happen, else we get a scatternet [9].

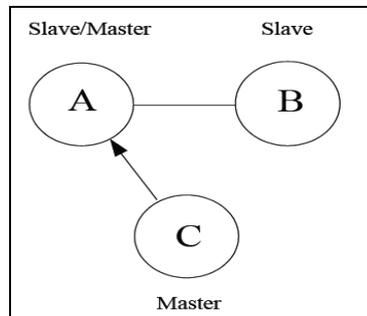


Fig. (7): Scatternet with 3 nodes

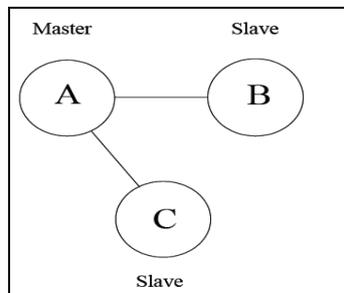


Fig. (8): Piconet with 3 nodes

5. Bluetooth Links

Two types of physical links are defined in version 1.1 of the Bluetooth specification, Synchronous Connection Oriented (SCO) links and Asynchronous Connectionless (ACL) links [11]. The SCO and ACL links are part of the base band specification. SCO links are intended for audio transmission. When setting up a SCO link time slots are reserved for transmission of data, thus providing a Quality of Service (QoS) guarantee. Lost or erroneous packages are not re-transmitted which makes sense for voice transmissions. All SCO links operate at 64 kbps. A master device can have up to three simultaneous SCO links at a time, all to the same slave or to different slaves. Slave devices can have up to three SCO links to the Master device. ACL links are intended for data communication. An ACL link provides error-free transmission of data which means that lost or erroneous packets are re-transmitted. No QoS guarantee is provided. The maximum data rate at the application level is around 650 kbps for an ACL link. A master device can have a number of ACL links to a number of different devices, but only one ACL link can exist between two devices.

User data is usually transferred to and from the Logical Link Control and Adaption Protocol (L2CAP) layer of the Bluetooth stack. Application developers usually refer to L2CAP and RFCOMM links when talking about Bluetooth links. To be precise, L2CAP and RFCOMM are separate layers in the Bluetooth stack which rely on an ACL physical link for data transmission. L2CAP provides multiplexing between different higher layer protocols over a single physical ACL link, enabling several logical data links to be set up between two Bluetooth devices. L2CAP also provides segmentation and reassembly of packets from higher layers. Different protocols use different packet sizes, some of these may need to be segmented in order to be sent over an ACL link due to package size constraints. An ACL packet can have a maximum of 339 bytes of payload data, while an L2CAP packet can have a maximum of 65,535 bytes of payload data. The RFCOMM layer emulates RS-232 serial ports and serial data streams. RFCOMM relies on L2CAP for multiplexing multiple concurrent data streams and handling connections to multiple devices. The majority of Bluetooth profiles make use of the RFCOMM protocol because of its ease of use compared to direct interaction with the L2CAP layer [12].

6. The Service Advertisement And Discover

A Bluetooth device may offer services for other devices to connect to other devices themselves and make use of their services. A Bluetooth device that offers services to others is known as a server, and the user is known as a client.

A Bluetooth server makes services available by means of process known as Service Advertisement – information about the new service is provided to any other Bluetooth devices that wish to query it. It does not necessarily guarantee that an attempted connection will be successful. When an application advertises its ability to accept connections, an entry is placed in the device's Bluetooth Service Discovery Database (SDDB). Regardless of the type of service provided, a Bluetooth service advertisement will consist of the following information:

- 1. Port:** The port number an incoming connection will be accepted on. Many applications can advertise their services on a device, with each accepting a connection on a specified port.
- 2. Class ID:** A class ID that will provide information on the type of service provided.
- 3. Availability:** The availability of this service. When an application accepts an incoming connection, it will mark its entry in the Service.



Discovery Database is not available. This will inform any other devices that may wish to connect that it is currently occupied. Most entries will also contain user-readable text to explain the type of service they offer, and it is even possible to provide a graphical icon for a service. On the client side, an attempt to make a Bluetooth connection is typically preceded by a process of searching for local Bluetooth-enabled devices and querying the services they provide. This process is known as Service Discovery [13].

7. The Proposed System Design

The main aim of the proposed system is to design and implement a mobile chatting over Bluetooth, as a group (not peer-to-peer), by using the Piconet and Scattering characteristics to extend the range of the application.

Fig. (9) shows the address book flowchart. In this part of the project, we created a MIDP project for the mobile client application using Java 2 Micro Edition program, Java platform, Micro Edition is a common language used for developing the mobile applications providing a flexible environment for the application running on mobiles and other embedded phones devices.

In this designed application, the user will have six choices to use.

1. Enter the Name.
2. Chat.
3. Clear.
4. About the program.
5. Help.
6. Exit.

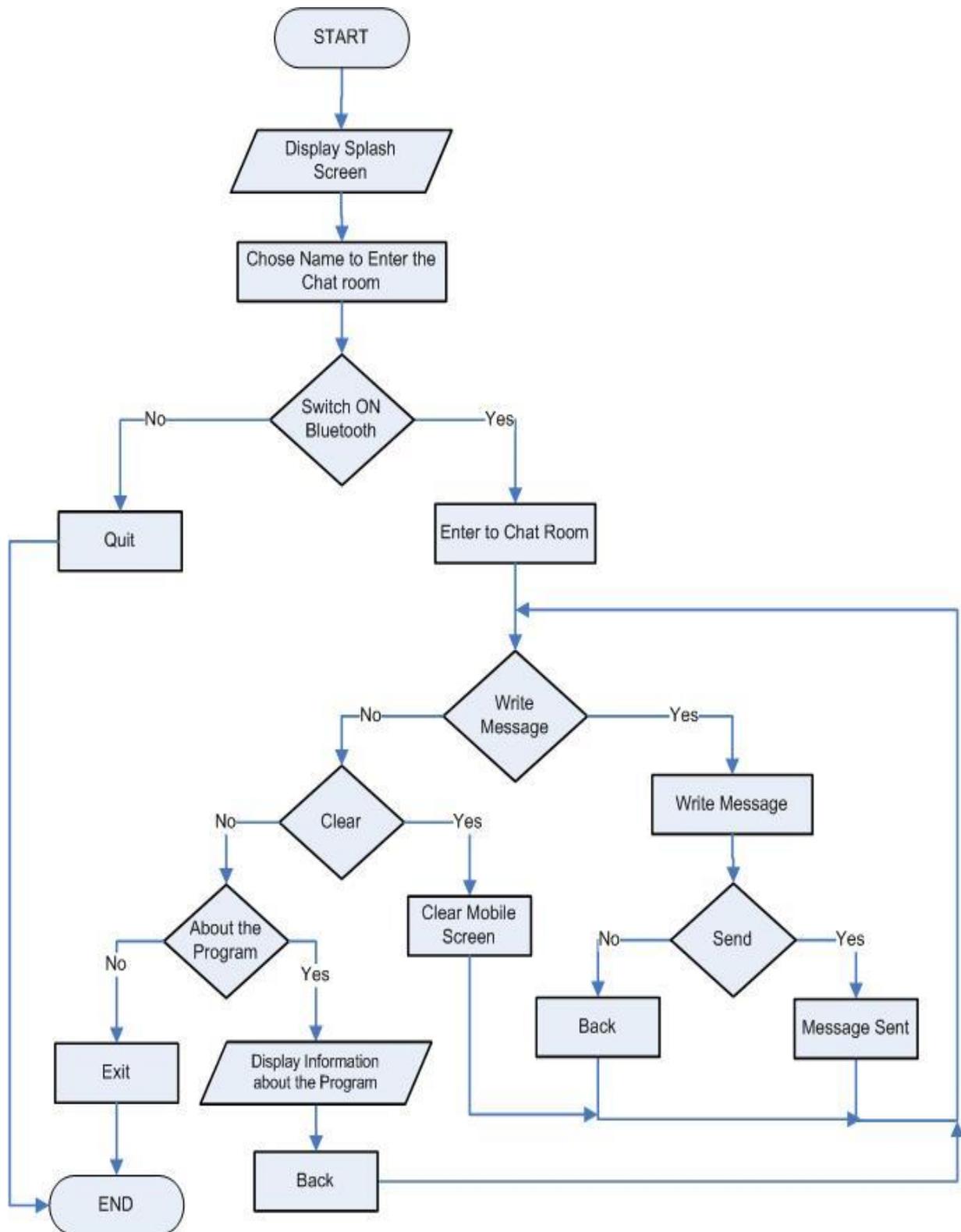


Fig. (9): The Proposed Main Menu Client Application Flowchart

8. The System Implementation

Users of this application have to download and install the application on their devices. After installing the application, the icon of the new application can be seen in the application menu of each device.



Fig. (10): The Intro Screen for the Application

After installing the application on the mobile client, a user can see that the icon of the application is created on the mobile device. By clicking on the application's icon, the starter screen will appear. Fig. (10) contains the start or intro screen for the application which just appears for few seconds. Then, automatically will disappear to show the next screen.

Fig. (11) is showing the option of choosing and entering the nick name of the chatting over Bluetooth. In this part, the user will enter the name, and the name will be the same name of the chatting for the current user.



Fig. (11): Add new part of the Application

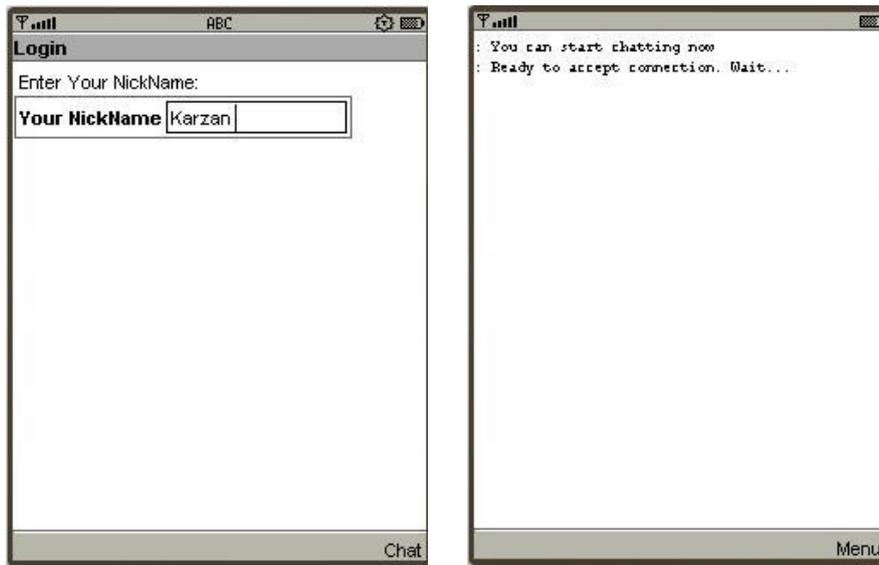


Fig. (12): Enter the Nick Name

As shown in Fig. (12) , when the user select and add his\her name for chatting, this notification will appear to show that the user is ready to accept and connect with any other device.



Fig. (13): About the Application

Fig. (13) is showing some details about the program. It includes short information about the application and the year of developing it.

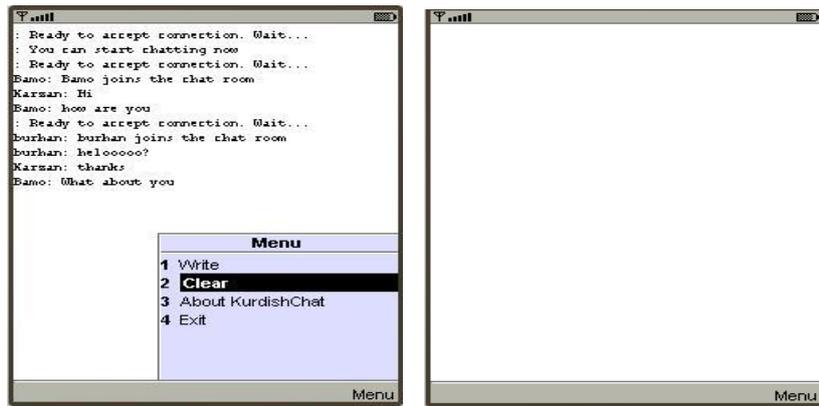


Fig (14): clear of the mobile screen

As shown in Fig. (14) , when the mobile screen is going to be full with previous messages, the user can clear and delete all of them by selecting the clear option.

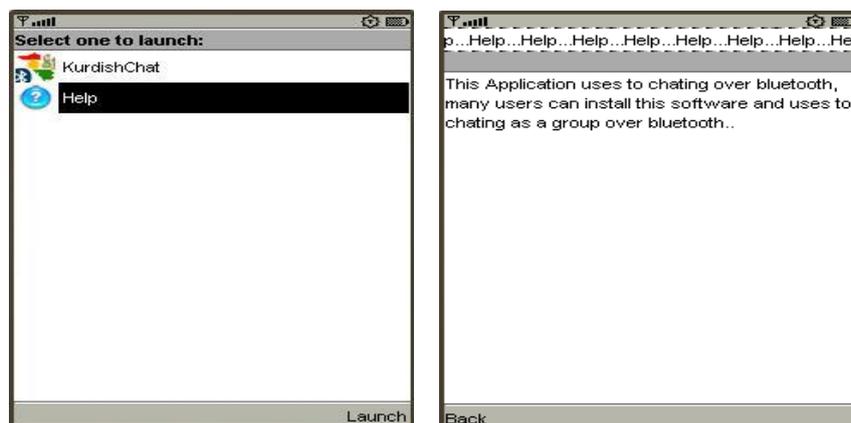


Fig. (15): The help of the application

As shown in Fig. (15) , any user can use this branch of the application to read the help file that will assist the users to use this application.



Fig. (16): The End of the Application

Fig. (16) shows the end screen of using our application. Also, it will appear for few seconds and automatically disappear.

9. Conclusions and Future Work

The most important points which is concluded throughout designing and implementation of this software are:-

1. By using this application, a user can improve a security. By using a private chatting room, a user will not be able to use this application without installing the application on his or her mobile phone, and user client cannot copy the application from a mobile device to another.
2. The distance problem of using the Bluetooth between devices has been solved by using intermediate users for connection.
3. The ability of chatting as a group over Bluetooth, not peer-to-peer.
4. It is completely free. The users of this application can chat as a group with no charge or fee.

The future works can be progressed to include many new parts using updated technologies and tools. The main future work can be developing the proposed software application to support files and pictures. In order to transfer not only messages, but also some other types of files. The same application can be developed to include image, audio, video clips also.

References

- [1] M. Barakat, et al., "*Mobile programming using PyS60: three designs, one Implementation*". ITF32508 (2008). [Online]. Available: <http://www.ia.hiof.no/~gunnarmi/bachelors/barakat2008mpu.pdf>. [Accessed: Jan. 10, 2015].
- [2] P. Layne, "*Software Architecture for Mobile Computing*", Master Thesis in Computer Science and information System department, University of Jyvaskyla, Oct. 1999.
- [3] A. Abdul Khaliq ,"*Design and Implementing a Private SMS using GPRS*", Thesis submitted to the Council of the College of Science, Sulaimani University in partial Fulfillment of the Requirements for the degree of Master of Science in Computer Science, September 2008
- [4] O. Kasten , and M. Langheinrich , "*First experiences with bluetooth in the smart-its distributed sensor network*" Workshop on Ubiquitous Computing and Communications, In Proc. PACT 2001, Oct. 2001.

- [5] Kleinschmidt, John et al, “*RFID Security*” Publisher: Syngress, 2006.
- [6] “*Bluetooth Technology Overview*”, Forum NOKIA, Version 1.0; April 4, 2003.
- [7] B. Miller, and C. Bisdikian, “*Bluetooth revealed. Upper Saddle River*”, NJ: Prentice Hall PTR, 2001.
- [8] A. Esteban “*Bluetooth Architecture Overview*” Applied in Windows Embedded Compact 7, Microsoft Copyright 2011
- [9] A. N. kilingsheim “*j2me Bluetooth programming*” a master thesis in Department of informatics , University of Bergen, 2004.
- [10] C. Kumar, and P. Kline, “*Bluetooth application programming with the Java APIs*”. San Francisco, CA: Morgan Kaufmann, 2004.
- [11] R. Zurawski , “*The Industrial Information Technology Handbook*” CRC Press, 2004.
- [12] A. N. Amir, “*Development Wireless Mobile Application for Series60 and Symbian OS Using J2ME*”, a Thesis submitted to College of Science, Baghdad University in partial Fulfillment of the Requirements for the degree of Master of Science in Computers, 2005.
- [13] R. Morrow, “*Bluetooth operation and use*”. New York: McGraw-Hill, 2002.

AUTHOR



Aysar A. Abdulrahman received his B.Sc. in Information and Communication Engineering from University of Baghdad, in 2003. In 2008, he received the M.Sc. degrees in Computer Science from University of Sulaimani. During 2008-2010, he worked as an assist lecturer at University of Sulaimani \ College of Science \ Computer Department. In Dec 2013, he got his Ph.D. in Computer Engineering from Southern Illinois University. Now, he is working as a lecturer in the University of Sulaimani \ College of Science \ Computer Department.