ISSN 2320 - 2602
Volume 5 No.1, January 2016
International Journal of Advances in Computer Science and Technology
Available Online at http://www.warse.org/IJACST/static/pdf/file/ijacst03512016.pdf

A Bluetooth-Based Remote Control System for Manipulating Screen Images

Samuel King Opoku
Computer Science Department, Kumasi Polytechnic, Ghana, samuelk.opoku@kpoly.edu.gh

Abstract: Remote controllers employing sensors have been common devices for managing such electronic devices as television and Digital Versatile Disc (DVD) players. Current remote control systems are categorized as radio, ultrasonic, laser, mechanical and infrared. However, mobile phones have being prevalent in our modern society. They are not currently used for talking alone but they are also used for carrying out sophisticated activities like video streaming and playing electronic games. This paper proposes and implements the use of mobile phone as remote control system with the help of Bluetooth technology. The implemented mobile phone allows a user to control and select screen images remotely and securely. The system is implemented using Java Platform.

Key words: Bluetooth, Image, JavaSE, JavaME, Remote Control

1. INTRODUCTION

Remote controllers employing sensors have been common systems for controlling devices. There are many advantages to using remote control systems. They help in controlling devices more comfortably and conveniently without necessarily walking towards the device to be controlled. In almost every situation, remote control systems are great time and effort savers. When it comes to handling radioactive materials, remote control system are useful in protecting human health. Remote control cars and boats are used for hobbies and entertainment [1]. Mobile phones have being prevalent in our modern society. They are not currently used for talking alone but they are also used for carrying out sophisticated activities like video streaming and playing electronic games. This paper proposes and implements the use of mobile phone as remote control system with the help of Bluetooth technology. The implemented mobile phone allows a user to control and select screen images remotely and securely. The system is implemented using Java Platform.

Remote control system is a system that is used to control devices and its execution is parted by a relatively significant distance [1]. The system generally includes a command device where the control command is entered and an actuator that executes it. These are connected by a transmission medium that transmits the command, usually in a coded format [1], [2]. The most characteristic remote control system involves response that is provided by the user of the system. The user issuing the control command senses the effect of the control action and guides the system accordingly [1], [3].

There are many different types of remote control system. Some of the types include radio, ultrasonic, laser, mechanical and infrared control [3], [4]:

- Radio control is used in controlling remote control vehicles like planes, boats, or cars. Radio control is an advantage because there are military tests currently being conducted to use radio control aircraft for surveillance and air strikes which could save the lives of soldiers
- Ultrasonic control is used in telephone answering machines and in some television sets.
- Laser control is used mostly in guided weapons such as bombs and missiles.
- Mechanical control is used in handling radioactive materials.
- Infrared control uses a photo detector in most televisions, VCR's, stereos, and car audio systems.

2. BLUETOOTH TECHNOLOGY AND PROTOCOLS

Bluetooth technology is a short-range communication technology that is simple and secure. Bluetooth technology was designed primarily to support simple wireless connection of personal consumer devices and peripherals, including cell phones, Personal Digital Assistants (PDAs) and wireless headsets [5]. Bluetooth came about to ease the problems associated with infrared systems. Bluetooth does not require line of sight between communicating devices. Bluetooth can connect up to eight devices simultaneously. The Bluetooth Specification defines a uniform structure for a wide range of devices to connect and communicate with each other [5], [6]. The basic strength of Bluetooth wireless technology is the capability to simultaneously handle data and voice transmissions; which provides users with a variety of innovative solutions such as hands-free headsets for voice calls, printing and fax capabilities and synchronization for PCs and mobile phones [5]-[7]. Another advantage of Bluetooth technology is its Adaptive Frequency Hopping (AFH) capability which was designed to reduce interference between wireless technologies sharing the 2.4 GHz spectrum [5], [8]. For users of Bluetooth technology this hopping provides greater performance even when other technologies are being used along with Bluetooth technology [5]. Bluetooth is limited by slow transfer rate, distance limitation and is also prone to high power consumption [9], [10].

Bluetooth architecture is made up of a number of protocols which can be segmented into four categories. They are [11]: Bluetooth core protocols, telephony control protocols, cable replacement protocols and adopted protocols. Each of this protocol is responsible for a specific type of job and independent. The four categories of the architecture is depicted below in figure 1:

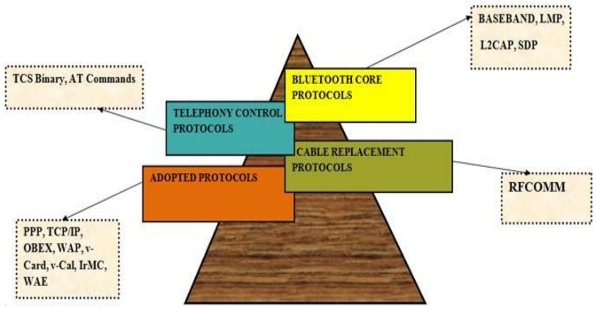


Figure 1 Protocol Categorization (Source [11])

The various protocols under Bluetooth Core Protocols are Baseband, Link Manager Protocol (LMP), Logical Link Control and Adaptation-Layer (L2CAP) and Service Discovery Protocol (SDP). Baseband enables radio frequency to be link between Bluetooth devices to form a Pico-net. The LMP controls the setting to link two Bluetooth devices dealing with security issues like authentication, encryption – exchange and monitoring the link and encryption keys [11]. The L2CAP supports higher level multiplexing, segmentation and reassembly of packets and quality of service communication and groups. The limitation with this layer is that, it is not responsible for reliability. SDP is basically for discovery of services on all Bluetooth devices. This part is important for all Bluetooth models because with SDP device information, characteristics of the services can be queried and connection between two or more Bluetooth devices may be established [5], [10]. RFCOMM protocol emulates the serial cable line settings. RFCOMM connects to the lower layers of the Bluetooth protocol stack through the L2CAP layer.

The Telephony Control Protocol (TCS) defines the call control signaling for establishing speech and data calls between two or more Bluetooth devices. It is bit oriented protocol instead of being packets. The Host Controller Interface (HCI) offers a command interface to the base band controller, link manager and access to the hardware status and control registers. This interface provides a uniform method of accessing the Bluetooth baseband capabilities. The Host control transport layer eliminates transport dependencies and provides a common driver interface. Three interfaces are defined in the core specification: Universal Serial Bus (USB), RS-232, and Universal Asynchronous Receiver/Transmitter (UART) [11].

Point-To-Point Protocol (PPP), Transmission Control Protocol and Internet Protocol (TCP/IP), User Datagram Protocol (UDP) and IP are standard Internet protocols defined by Internet Engineering Task Force (IETF). These are used as the lower layer protocols for transferring packets or datagrams on their specified IP addresses. Object Exchange (OBEX) is a session protocol defined by Infrared Data Association (IrDA). This protocol is also utilized by Bluetooth thus enabling the possibility for application to use either the Bluetooth radio or IrDA technologies.

In Wireless Application Protocol (WAP), Bluetooth may be used as a bearer technology for transporting data between a WAP client and a nearby WAP server. WAP operates on top of the Bluetooth stack using PPP and the TCP/IP protocol suite. Each of these protocols is arranged neatly as layers one above the other forming a stack of protocols. Non-specific protocols can be used with many other platforms like WAP, UDP and OBEX. These were used to speed up the development of Bluetooth protocol at higher layers, ensuring adaptation with Bluetooth devices and interoperability [11], [12].

3. JAVA PLATFORM TECHNOLOGY

Writing or coding Java applications needs development tools like the Java Development Toolkit (JDK). The JDK comprises of the Java Runtime Environment, the Java complier and the Java Application Programming Interface (APIs). The JDK is a very useful tool that helps users to manage applications using the right vision of Java [13], [14]. Java Platform, Standard Edition (Java SE) enables users develop and organize Java applications on desktops and servers, as well as in today's challenging embedded environments. Java offers the rich user interface, performance, versatility, portability and security that today's applications require [15], [16]. Java Platform, Standard Edition (Java SE) makes applications available across heterogeneous environments so that businesses can boost end-user productivity, communication and collaboration which drastically reduce the costs of enterprise and consumer applications [17], [18].

Java Platform Micro Edition (Java ME formerly J2ME) refers to a collection of technologies and specifications of a Java runtime environment that specifically fit the requirements of a particular device such as mobile phones, Blu-ray Disc players, digital media devices and printers [19]. The Java ME Platform is solely used for building mobile applications. It comprises a number of specified components, which have been defined by industry through the Java Community Process (JCP). Java ME is designed to provide portability of applications between platforms. It is a stripped-down version of Java targeted at devices which have limited processing power, storage capabilities and intermittent or fairly low-bandwidth network connections [19], [20]. The Java ME technology is based on three fundamentals [20]-[22]:

- A configuration which provides the most basic set of libraries and virtual machine capabilities for a broad range of devices,
- A profile which defines a set of APIs that support a narrower range of devices, and
- An optional package known to be a set of technology-specific APIs.

The Java ME platform has been segmented into two base configurations, one to fit small mobile devices and one to be targeted towards more capable devices like smart-phones and set top boxes. The configuration for small devices is called the Connected Limited Device Configuration (CLDC) and the more capable configuration is called the Connected Device Configuration (CDC) [22], [23]. Java ME Profiles define additional class libraries and APIs needed to enable domain-specific applications on a particular configuration. For instance, the Mobile Information Device Profile requires at least the Connected, Limited Device Configuration. It enables development of applications to provide wireless access to information. The most significant profile for CDC is the Foundation Profile which explicitly supports devices with no user interface but it does not support networking [24].

Profiles lie on top of a configuration and will not function as required without the underlying configuration. Profiles target devices in a specific market (for example MIDP profile, which uses of the CLDC configuration targets low end cell phones, while the Personal Profile is targeted towards higher-end mobile devices). Profiles can include the java classes that focus on specific implementations such as user interface components and record management [23]. Figure 2 depicts an overview of the components of Java ME technology.

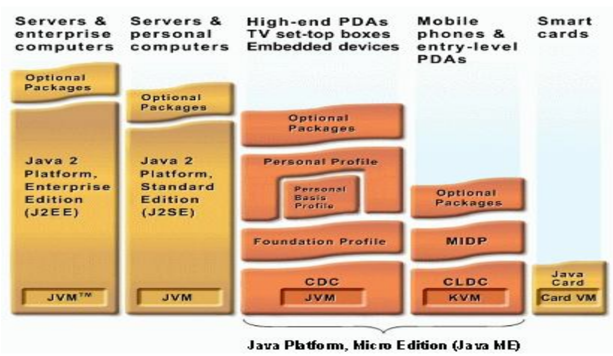


Figure 2 Components of Java ME Technology (Source [22])

A typical Java ME application consists of a Java source file and optionally you can add PNG files (for graphic files), MIDI and WAV files for audio files. The build and compile stage creates two files, Java Archive (JAR) file which contains all the executable files and a JAVA descriptor (JAD) file. A Java ME application is made up of MIDlet class which has four different states: loaded, active, paused and destroyed [25], [26]. The figure below illustrates how the states are interconnected

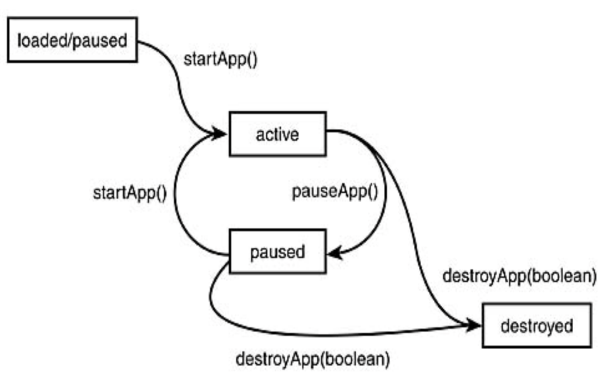


Figure 3 MIDlet States (Source [25]).

4. JAVA AND BLUETOOTH INTEGRATION

In Java, the basic concept of any Bluetooth application contains elements like Stack Initialization, Device Discovery, Device Management, Service Discovery and Communication [12]. Bluetooth can be integrated into Java SE by integrating the Java SE platform with the BlueCove API. BlueCove API provides Java API for Bluetooth Java Specification Request (JSR 82). BlueCove is a Java library for Bluetooth (JSR-82 implementation) that currently interfaces with the Mac OS X, WIDCOMM, BlueSoleil and Microsoft Bluetooth stack found in Windows XP SP2 or Windows Vista, WIDCOMM and Windows Mobile [27].

BlueCove provides JSR-82 Java interface for the following Bluetooth Profiles: Service Discovery Application Profile (SDAP) which describes how an application should use SDP to discover services on a remote device, RFCOMM emulates the serial cable line settings and status of an RS-232 serial port and is used for providing serial data transfer, Logical Link Control and Adaptation Protocol (L2CAP) which supports higher-level protocol multiplexing, packet segmentation and reassembly. It also conveys Quality of Service (QoS) information and Object Exchange (OBEX) [26], [27].

Currently, JSR-82 cannot alone be implemented on a Java SE platform. A Generic Connection Framework must also be implemented by the Java SE. The generic connection framework is normally implemented by JDK. The JSR-82 actually consists of two independent packages [28]:

- Javax.bluetooth: classes and interfaces needed to perform wireless communication with the Bluetooth protocol
- Javax.obex: classes needed to send objects between devices, independent of the transmission protocol between them.

Figure 4 below depicts a stack diagram of a Bluetooth API:

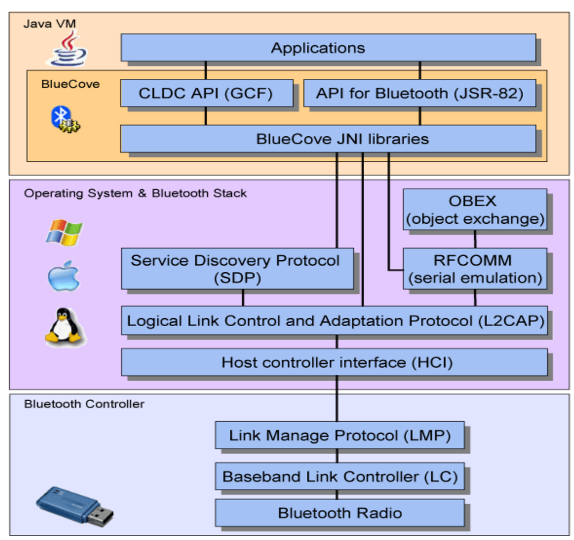


Figure 4 A Stack Diagram of a Bluetooth API (Source [27]).

Every Bluetooth device has a unique Bluetooth address, which is 48 bit long (normally expressed in 12 hex digits) and a Bluetooth "friendly name", a string of printing ASCII characters [29], [30]. A client must engage a connection by "searching" the server device. To speed up this process, an operating system utility called Bluetooth Control Center (BCC) can perform the device (and service) search and store the devices and their services in an internal database. Stored devices are called "preknown". For security reasons the "pairing" process of two Bluetooth devices requests a security code that must be confirmed on the remote device [30].

5. REMOTE CONTROL SYSTEM DESIGN

The PC acts as the server that contains images which are manipulated by the phone. A single phone can be used at a time. The PC is the fundamental server to the mobile application. The PC receives commands from the mobile phone that is manipulated by the user. The PC in turn responds to commands by displaying the appropriate images. If something goes wrong, it will notify the user by sending an error message back to the mobile phone. Once the user gains access (authorize) to use the application on the server, the user then uses the mobile phone to manipulate images on the PC. Figure 5 depicts a use case diagram for the interface design of the PC:

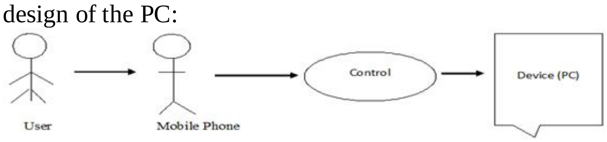


Figure 5 Interaction Design of the System

The user will need to logout after successfully using the system so that others can have access. Below is the sequence diagram of the design:

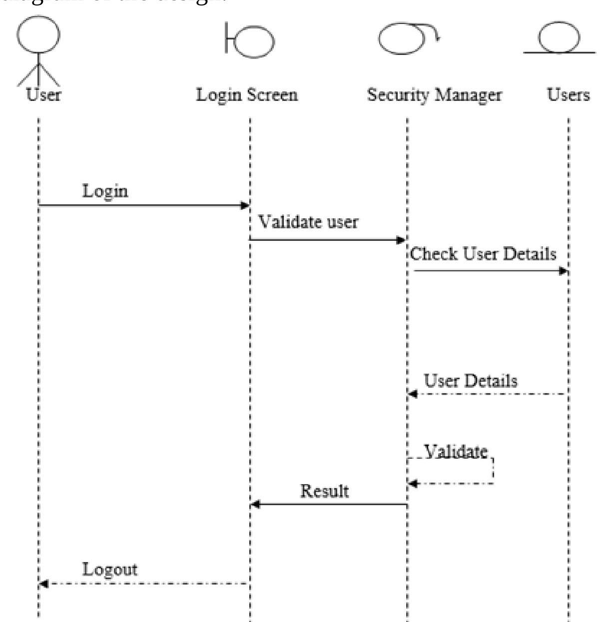


Figure 6 Interface Design of the System

After the mobile phone has been able to communicate with the PC, the user is now able to select images from the PC using the mobile phone and the computer then displays the selected images on its screen.

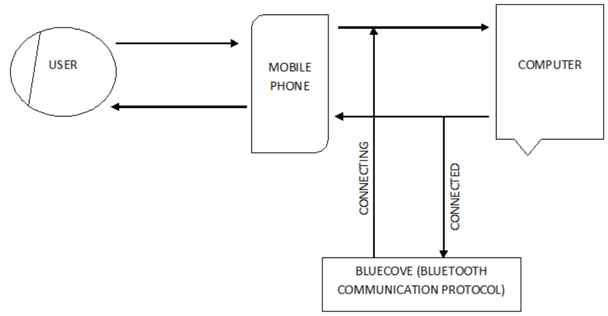


Figure 7 System Architecture

With the user-system interaction, users basically interact with the system through the mobile phone (client side). There is no direct interaction between the user and the computer (PC). The various parts making up the mobile phones interface are responsible for handling user actions and selections.

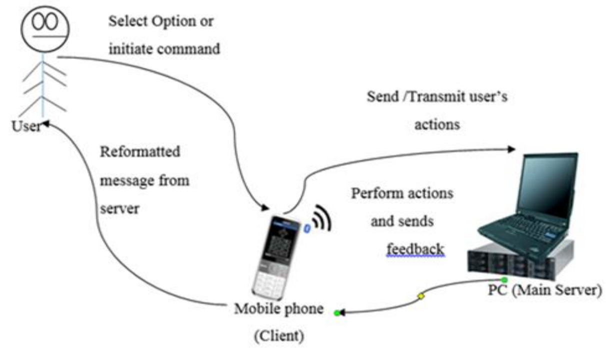
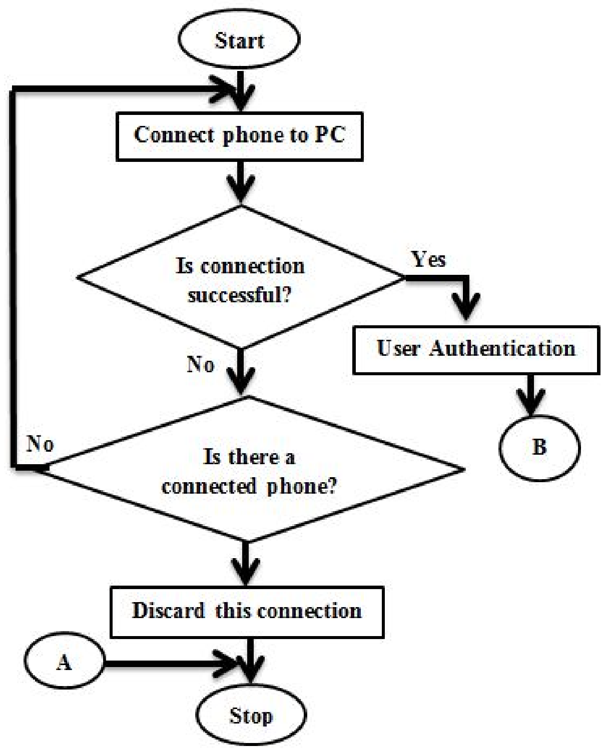


Figure 8 User-System Interaction

Algorithm for system interaction is illustrated below:



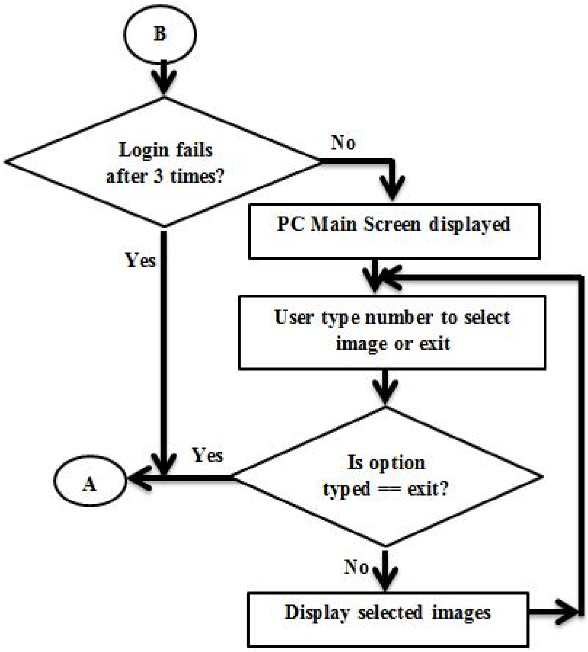


Figure 9 Overview of System Interaction

6. SYSTEM IMPLEMENTATION AND TESTING

After a successful connection between the client and the server, the main page of the PC screen is the categories of the various sets of images to work with. The figure below illustrates the main screen of the PC.

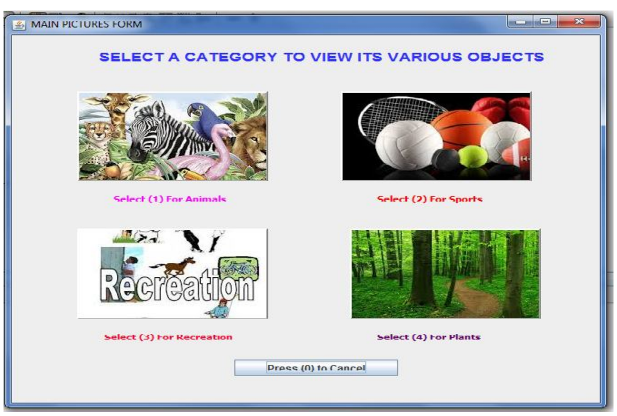


Figure 10 Main Interface of the PC System

These are animals, sports, recreation and plants. The option is selected by typing the appropriate number as:

- 1 for animals
- 2 for sports
- 3 for recreation
- 4 for plants
- A user will need to type 0 to exit the program

If Option 1 is selected, the following interface is obtained.

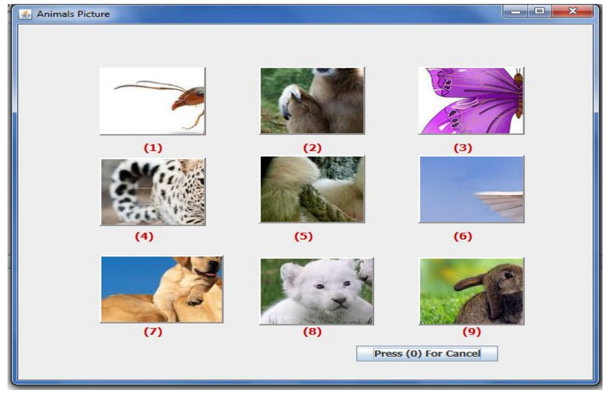


Figure 11 Animals Pictures Interface Design

The animal's picture interface design depicts the various images that can be displayed under the animal's category. The pictures are limited to only nine (9). A user would need to select 1 through 9 to view any of the pictures. Selecting or pressing 0 will exit this form and return to the categories interface.

After entering the digit, the user presses the Options button and then selects OK. The user can press the Next button to view the next image of the selected category, if available. The user can press the Back button to view the previous image of the selected category. The buttons on the PC screen determines the buttons on the mobile device. Figure 12 illustrates the mobile phone interface that allows the user to enter a number to control the PC

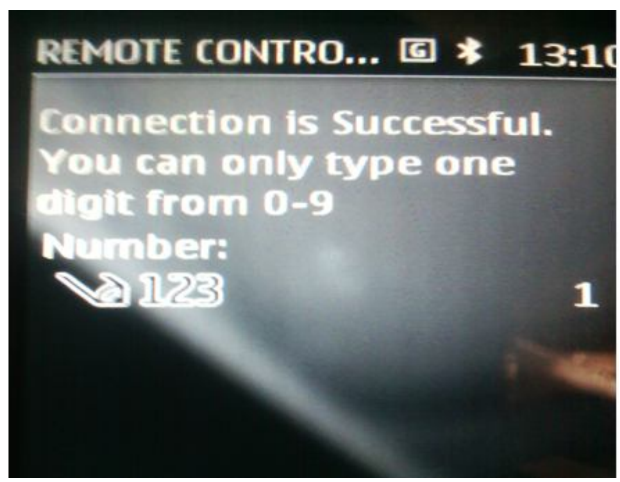


Figure 12 Mobile Phone Interface

When a user selects any of the images using one of the selection key (1- 9), the full image size is displayed on another form. From the above mobile interface, option 1 was selected and the full image size is displayed as shown in the figure below:

Figure 13 Full Image Size in the Animals Category

A user can view the previous or next pictures in succession. However, when the first picture is selected, the Previous button on the PC interface is disabled which also disables the Back key on the mobile phone interface. When the last picture is selected; the Next button on the PC interface is disabled to identify the end of pictures that can be viewed. This also disables the Next key on the mobile phone interface. The various buttons available is displayed in Figure 14 below:

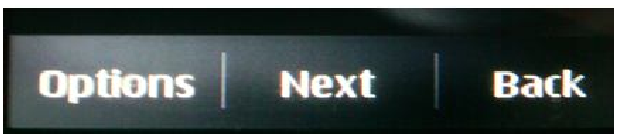


Figure 14 Available Buttons for the Mobile Application

7. CONCLUSION

A mechanism for implementing a remote control system using a mobile phone has been designed. The system ensures maximum security by providing a login form so that clients can use the system after a successful authentication. It also underlines an alternative mode to the traditional option of selecting images on a PC screen by the use of a mouse pointer or the keyboard. In almost every situation, remote control systems are great time and effort savers.

The following is a list of constraints of the system:

- The receiver must reside in a location where a signal with sufficient strength can be received.
- The only person who can communicate with the control module is the person who will be successfully authenticated.
- The images to be selected and controlled on the screen is only limited to nine (9)

REFERENCES

[1] http://www.answers.com/topic/times-topics-remote-control-systems#ixzz2jBsTg6a3 (Last accessed 7th November, 2013)

[2] H. Campbell, "An Enhanced Remote Controller System for Manipulating Screen Images Using a Mobile Phone," HND dissertation, Computer Science Department, Kumasi Polytechnic, 2014

[3] http://www.techopedia.com/definition/17269/remote-control-rc (Last access 07/11/13)

[4] http://library.thinkquest.org/25304 / (Last access 07/11/13)

[5] http://www.bluetooth.com/Pages/Basics.aspx (Last accessed 9th November, 2013)

[6] http://electronics.howstuffworks.com/bluetooth1.htm (Last accessed 9th November, 2013)

[7] S. Rathi, "Infrastructure: Bluetooth Protocol Architecture", Microware Architect, Microware System Corporation, p. 1 – 6, 2000

[8] B. Hopkins and R. Antony, "Bluetooth for Java", ISBN 1-59059-78-3, pp 33-35, 2003

[9] http://www.ask.com/question/what-are-the-limitations-of-bluetooth (Last accessed 9th November, 2013)

[10] M. M. Organero, S. K. Opoku, "Using NFC Technology for Fast-Tracking Large-Size Multi-Touch Screens", Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT), Vol. 2, No. 4 April Edition, 2011

[11] http://www.engineersgarage.com/articles/bluetooth-protocol-types-security (Last accessed 14th November, 2013)

[12] ber, 2013)

[13] http://www.java.com/en/download/faq/develop.xml (Last accessed 7th November, 2013)

[14] http://www.oracle.com/technetwork/java/javase/overview/index.html (Last accessed 7th November, 2013)

[15] S. K. Opoku, "A Framework for Designing Micro-browser for MIDP Applications," presented at 1st International Conference on Applied Science and Technology, Kumasi, November 19-21, 2014

[16] http://www.oracle.com/us/technologies/java/standard-edition/overview/index.html (Last accessed 7th November, 2013)

[17] http://www.oracle.com/technetwork/java/javase/overview/faqs-jsp-136696.html (Last accessed 7th November, 2013)

[18] http://developer.nokia.com/Community/Wiki/Java_ME (Last accessed 7th November, 2013)

[19] http://java.com/en/download/faq/whatis_j2me.xml (Last accessed 7th November, 2013)

[20] http://uberthings.com/mobile/ (Last accessed 7th November, 2013)

[21] http://www.oracle.com/technetwork/java/javame/about-java-me-395899.html (Last accessed 7th November, 2013)

[22] http://www.blueboard.com/javame/intros.htm (Last accessed 7th November, 2013)

[23] http://www.wirelessdevnet.com/channels/java/features/j2me.html (Last accessed 13th November, 2013)

[24] http://www.java-forums.org/java-tutorial/7734-midlet-programming.html (Last accessed 18th December, 2013)

[25] http://compnetworking.about.com/cs/bluetooth/g/bldef_bluetooth.htm(Last accessed 9th November, 2013)

[26] http://electronics.howstuffworks.com/bluetooth1.htm (Last accessed 9th November, 2013)

[27] https://developer.bluetooth.org/TechnologyOverview/Pages/RFCOMM.aspx (Last accessed 15th November, 2013)

[28] http://bluecove.org/ (Last accessed 15th November, 2013)

[29] http://www.google.com.gh/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&cad=rja&ved=0CFAQFjAG&url=http%3A%2F%2Fwww.springer.com%2Fcda%2Fcontent%2Fdocument%2Fcda_downloaddocument%2F9781590590782-c1.pdf%3FSGWID%3D0-0-45-89804-p2293780&ei=ZraEUry9Fce70QWhyoHIDQ&usg=AFQjCNEObVkdnA_J1VXkLnn-PCk82AifVw&bvm=bv.56343320,d.d2k (Last accessed 15th November, 2013)

[30] http://www.google.com.gh/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&cad=rja&ved=0CDkQFjAD&url=http%3A%2F%2Fwww.oracle.com%2Fus%2Ftechnologies%2Fjava%2Fojwc-datasheet-167822.pdf&ei=M1mGUoHHGImohAe_7YCoBg&usg=AFQjCNE6QRVTz0TLehNaq-nrjz5rq6DfEw&bvm=bv.56643336,d.ZGU (Last accessed 15th November, 2013)