IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-5, May 2018 E-ISSN: 2347-2693

Access Control Android Application to Remotely Control PC

Shalabh Agarwal[1*], Sayantika Sengupta[2], Swarnali Sanyal[3], Naireeta De[4]

[1]Department of Computer Science, St. Xavier's College (Autonomous), Kolkata, India
[2] Department of Computer Science, St. Xavier's College (Autonomous), Kolkata, India
[3] Department of Computer Science, St. Xavier's College (Autonomous), Kolkata, India
[4]Department of Computer Science, St. Xavier's College (Autonomous), Kolkata, India

**Corresponding Author: Shalabh Agarwal. Email: shalabh@sxccal.edu, Tel.:+91-33-22551281*

Available online at: www.ijcseonline.org

Accepted: 19/May/2018, Published: 31/May/2018

Abstract- Access Control is an Android Application that works on the concepts of wireless socket programming. Our main aim for developing this application is to provide the user with a remote for his/her PC in the form of an android device. Using this application, the user may perform various actions on his PC such as controlling the mouse movement and operations, sliding through various PPT slides, managing media, entering text on any application on their PC, from a reasonable distance, all just with the help of their android device. In order for this application to work, the PC and the given android device need only be connected to a common network.

Keyword- Android, remote, wireless, socket program, client-server model, IP address, TCP/IP

I. INTRODUCTION

Access Control involves wireless control of a computer through an android device[1]. The android application turns the phone into several modes-

(i) Touchpad mode
(ii) PPT mode
(iii) Media mode
(iv) Keyboard mode

(i). **Touchpad mode:** The android device acts as a remote touchpad for the user's computer. Through this mode, the cursor movement and other mouse operations can be performed on the PC. It also allows maximization and minimization of applicaton windows on the PC.
(ii). **PPT mode:** This mode allows the user to start and end slideshows and navigate through the slides by the press of certain buttons.

(iii). **Media mode:** In this mode, the can control any media file on Windows Media Player on their PC, that is, the user can play, pause and navigate through the files.
(iv). **Keyboard mode:** This mode helps the user handle their computer's keyboard remotely. The user can enter/edit text in any application on the PC.

The connection between the computer and the android device is based on client-server communication through socket programming[2]. Server-client model is communication model for sharing the resource and provides the service to different machines. Server is the main system which provides the resources and different kind of services when client requests to use it. It uses the TCP/IP network.

Rest of the paper is organized as follows, Section I contains the introduction to the android application and client server communication through wireless socket programming, Section II contains the related work of remote control application, Section III contains the background of client server communication and socket programming, Section IV explains the methodology, Section V describes the result and discussion, Section VI concludes the research work with future directions.

II. RELATED WORK

Many developers are currently working on applications which can control the PC so that all kinds of operations can be performed with it. They are also trying to integrate more technologically advanced features into their applications.

III. BACKGROUND

The client–server model is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system. A server host runs one or more server programs which share their resources with clients. A client does not share any of its resources, but requests a server's content or service function.

TCP provides a reliable, point-to-point communication channel that client-server application on the Internet use to communicate with each other[3]. To communicate over TCP, a client program and a server program establish a connection to one another. Each program binds a socket to its end of the connection. To communicate, the client and the server each reads from and writes to the socket bound to the connection. A socket is one end-point of a two-way communication link between two programs running on the network.

Socket endpoints on TCP/IP networks have a unique address that is a combination of an IP address and a TCP/IP port number. Since the socket is bound to a specific port number, the TCP layer can identify the application that should receive the data sent to it.

Typically a server runs on one computer and has a socket that is bound to a specific port. The server waits for a different computer to make a connection request. The client computer knows the hostname of the server computer and the port number on which the server is listening. The client computer identifies itself and the server permits the client computer to connect.

A typical interaction between a client and a server is based on the wireless socket programming works as follow:

1. The server binds a socket to a specific port number and starts waiting for clients.
2. A client initializes a connection with the service specified by its host name and port number.
3. The server accepts the connection made by the client and creates a new socket for communicating with it.
4. From the point of view of the client it is (usually) the socket which was used to initialize the connection.

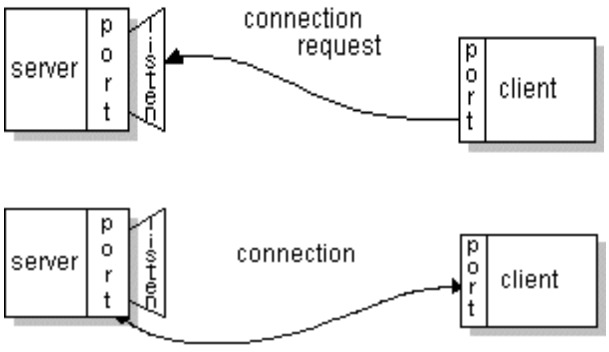

Figure 1: Client-server communication

IV. METHODOLOGY

1. Socket Establishment

The wireless connection between the server (computer) and the client (android device) is established by wireless socket programming.

Server Side:

The server side is implemented using Netbeans software, the programming language being Java.

The **java . net** package in the Java platform provides helps in the implementation of the server side of a client-server program[4].

Socket class- This class is used to create a socket.

ServerSocket class- This class is used to establish connection with the client.

ServerSocket server=new ServerSocket(SERVER_PORT);
Socket client=server.accept();

SERVER_PORT is an integer variable which stores the port number on which the server is listening. The object **server** open a socket on the port specified in **SERVER_PORT**.

When a connection is successfully established, the **accept** method returns a new Socket class object **client** which is bound to the same local port as specified by **SERVER_PORT**. The server communicates with the client over this new socket.

Client Side:

The client side is an Android application implemented in Android Studio. On the client side, a socket has to be created which will be bound to the specified port on the specified IP address[6].

Again, **java.net** package is used for the client establishment on the client side.

InetAddress class- This class represents an Internet Protocol (IP) address. An instance of an InetAddress consists of an IP address and possibly its corresponding host name.

InetAddress serverAddr = InetAddress.getByName("192.168.43.62");

This instruction verifies whether the specified IP address is authentic.

Socket class- Creates a stream socket and connects it to the specified port number at the specified IP address.

private Socket socket;
socket=new Socket(serverAddr,Constants.SERVER_PORT);

This instruction creates a socket on the host specified by serverAddr and port number SERVER_PORT.

Once the connection is established, data sent by the client will be processed by the server. From the client, instruction is sent to the server depending on the operation selected on the android application.

This involves data transmission from the client and the data handling on the server side[5].

2. Data Transmission

The data is sent from the android application using a global object of the **PrintWriter** class in java. **PrintWriter** is a class of the **IO** package.

public static PrintWriter out;

Here, out is the object of class PrintWriter which will store the data to be sent to the server.

out = new PrintWriter(new BufferedWriter(new OutputStreamWriter(socket .getOutputStream())), true);

The above instruction creates an output stream to send data to the server.

The simple **println** function is used to send the data to the server.

- void mouseRelease(int buttons): Releases one or more mouse buttons. . The parameter 'buttons' indicates which mouse button needs to be pressed.
- void keyPress(int keycode): Presses the given key specified by 'keycode'. The key should be released using the keyRelease(int). For example, if the keycode is KeyEvent.VK_A, it means that key A will pressed.

Example: *out.println("next");*

3. Data Handling

The data sent by the client needs to be processed, and the corresponding action need to be performed on the computer.

The **Robot** class of the java **AWT** package is used for the very purpose. This class constructs an object to control the mouse and the keyboard on the server side (in this case, the computer).

Some important functions of the class are:

- void mouseMove(int x,int y): Moves mouse pointer to given screen's x and y coordinates.
- Void mousePress(int buttons): Presses one or more mouse buttons. The parameter 'buttons' indicates which mouse button needs to be pressed. For example: If it is BUTTON1, it indicates that the left mouse button needs to be pressed.
- void keyRelease(int keycode): Releases the given key specified by 'keycode'.

This sums up how the socket is established, followed by the data transmission from the client side and simultaneous processing on the server side.

V. RESULTS AND DISCUSSIONS

The functioning of both the server (computer) and the client (android application) side can be explained by the following steps:

1. On the host computer which is the server side, we execute the JAR file for the server program. The following screen appears, along with a message box, asking the client to connect to the server.

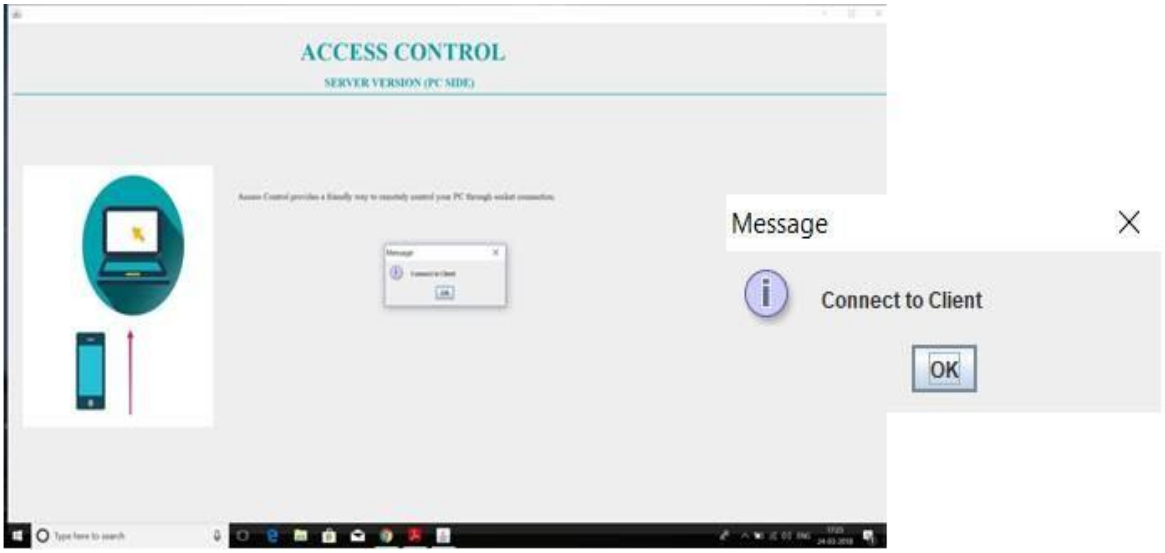

Figure 2: Server side (message box 1)

2. The IP Address that is used to establish the connection is displayed in the message box on the server side computer screen.

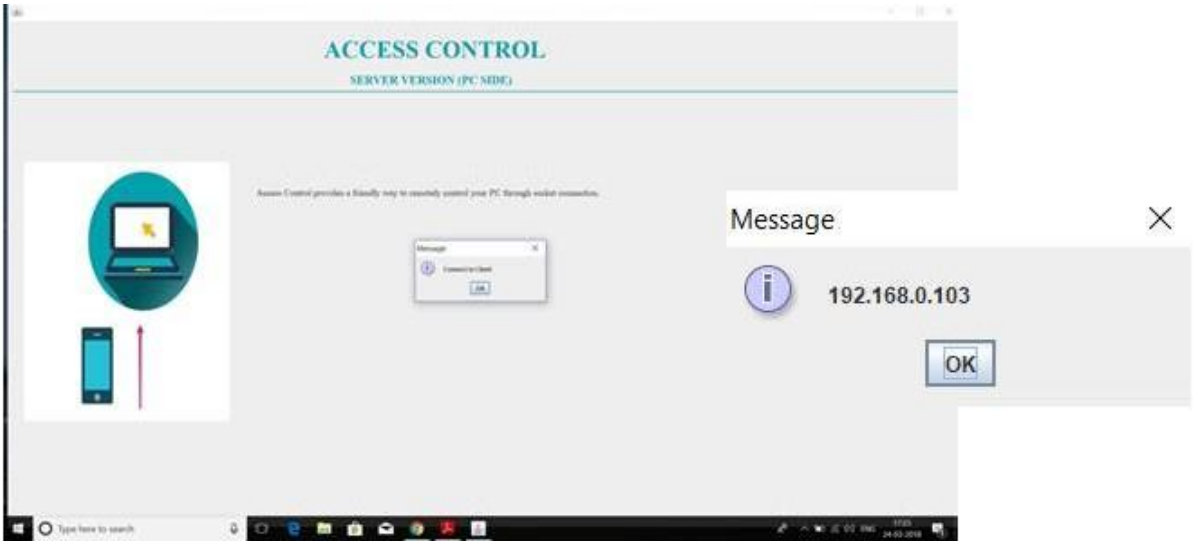

Figure 3: Server side (message box 2)

3. Next, Android application 'ACCESS CONTROL' which is the client side is opened. A splash screen appears on the android device which remains for a specific time and once the timer is out, it automatically moves on to the connection screen.

Figure 4: Splash Screen

4. On the connection screen, the 'CONNECT' button is used to send connection request to the server. The 'CONTINUE' button is used to move to the next page of the application once the connection has been established.

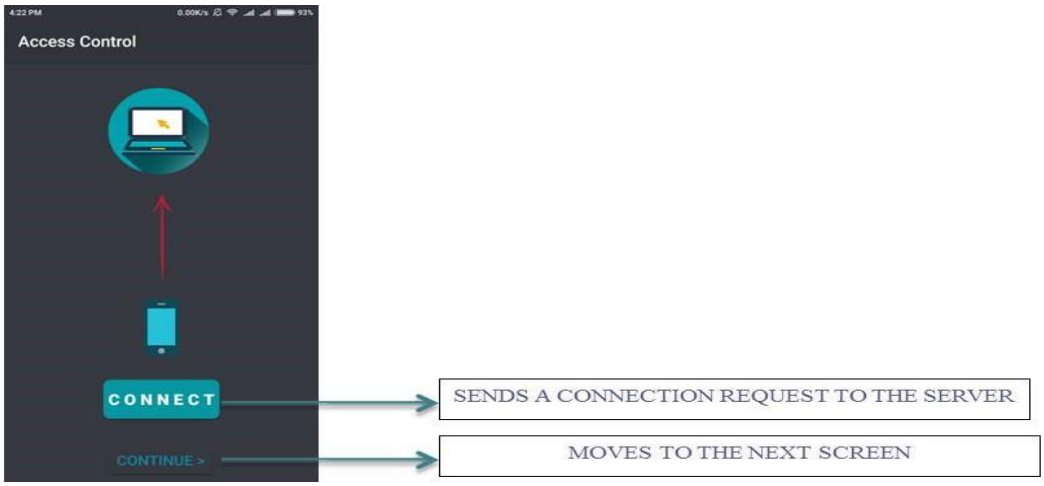

Figure 5: Connection Screen

5. After a connection request has been successfully sent by the client, the server receives it and accepts the request and displays the connection status in a message box on the computer screen.

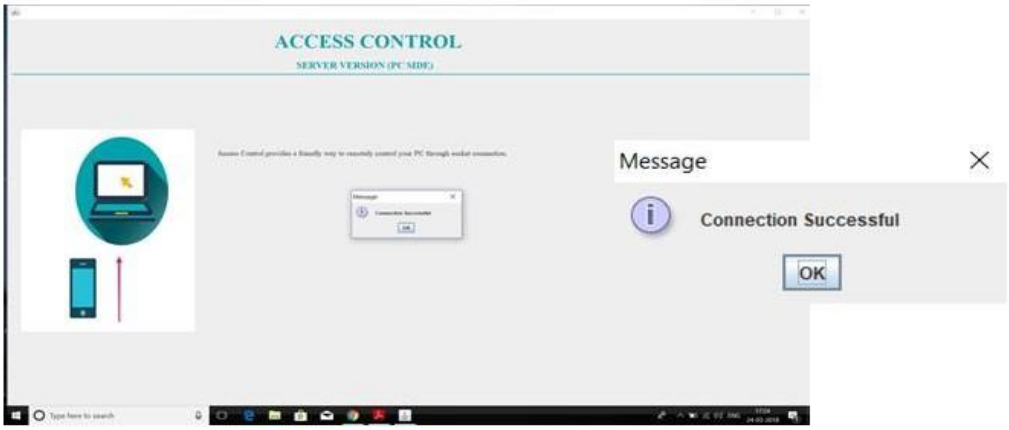

Figure 6: Server side (message box 3)

6. The next page of the Access Control application is the TOUCHPAD screen. It performs all kinds of operations related to the mouse.

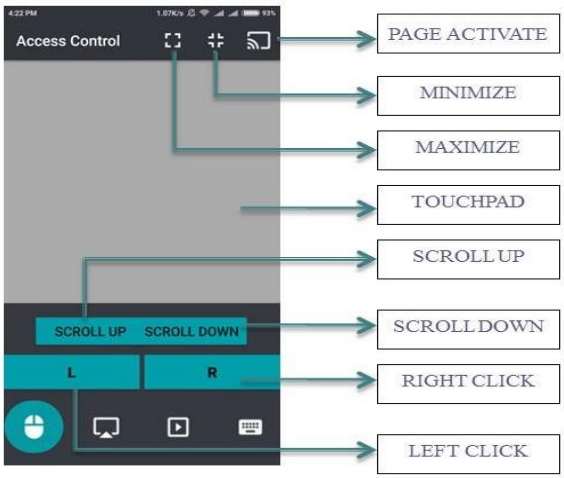

Figure 7: Touchpad Mode

- **Touchpad:** it captures the user's finger movement in the x and y directions of the text view and sends it to the server.
- **Scroll Up/Scroll Down:** sends the instruction "SCROLL_UP" / "SCROLL_DOWN" to the server. The server simulates the press and release of "pg up"/ "pg dn" keys respectively.
- **Maximize/Minimize:** sends the instruction "fullscreen"/"fullscreen_exit" to the server. The server simulates the press and release of "Win + pg up"/ "Win + pg dn" keys respectively.
- **Left Click/Right Click:** sends the instruction "MOUSE_LEFT_CLICK" / "MOUSE_RIGHT_CLICK" to the server. The server simulates the press and release of left button/ right button of mouse respectively.
- **Page Activate:** it activates the current page.

7. The next page of the Access Control application is the PPT screen. It performs operations related to PowerPoint slides.

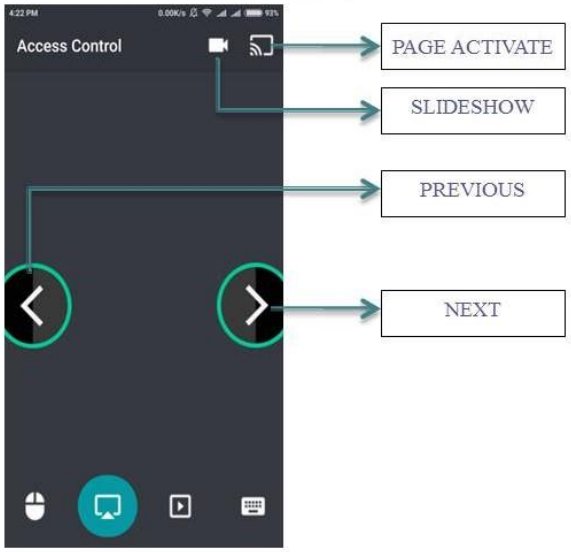

Figure 8: PPT Mode

- **Slideshow button:** sends the instruction "Slideshow" to the server. The server simulates the press and release of "F5" key.
- **Previous button:** sends the instruction "PREVIOUS" to the server. The server simulates the press and release of "Left Arrow" key.
- **Next button:** sends the instruction "NEXT" to the server. The server simulates the press and release of "Right Arrow" key.
- **Page Activate:** it activates the current page.

8. The next page of the Access Control application is the MEDIA screen. It performs all kinds of operations related to media.

Figure 9: Media Mode

- **Play/Pause button:** sends the instruction "play/pause" to the server. The server simulates the press and release of "Ctrl+P" key.
- **Previous button:** sends the instruction "prev" to the server. The server simulates the press and release of "Ctrl+B" key.
- **Next button:** sends the instruction "next" to the server. The server simulates the press and release of "Ctrl+F" key.
- **Page Activate:** it activates the current page.

9. The next page of the Access Control application is the KEYPAD screen. It performs all kinds of operations related to the keypad.

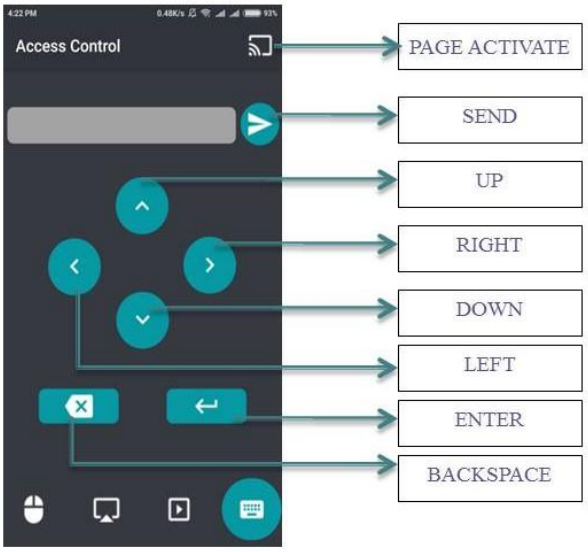

Figure 10: Keyboard Mode

- **Up button:** sends the instruction "UP" to the server. The server simulates the press and release of "Up Arrow" key.
- **Down button:** sends the instruction "DOWN" to the server. The server simulates the press and release of "Down Arrow" key.

- **Left button:** sends the instruction "LEFT" to the server. The server simulates the press and release of "Left Arrow" key.
- **Right button:** sends the instruction "RIGHT" to the server. The server simulates the press and release of "Right Arrow" key.
- **Enter button:** sends the instruction "ENTER" to the server. The server simulates the press and release of "Enter" key.
- **Backspace button:** sends the instruction "BACK" to the server. The server simulates the press and release of "Backspace" key.
- **Send button:** sends the message typed in the textbox to the server. Server stimulates the press and release of the corresponding keys for each character.
- **Page Activate:** it activates the current page.

VI. CONCLUSION AND FUTURE SCOPE

Access Control works on the concept of Android Application Development coupled with wireless socket programming. In order for Access Control to work, both the android device and the PC must be connected to a common network connection (not any local hotspot). This makes the application more efficient since it facilitates the software to work not only for a laptop, but also a desktop; the only requirement being a common IPv4 address. Access Control functions in the form of a client – server system which is implemented using wireless socket programming. However, the connections can also be established through Wi-Fi or Bluetooth. The above mentioned action however remains as an area of our project that could be probed into later.

ACKNOWLEDGEMENT

We would like to extend our gratitude to the Department of Computer Science at St. Xavier's College (Autonomous) for helping with the development of the software project. We are thankful to all our department professors and colleagues for their constant support and encouragement.

REFERENCES

[1]Suhas Hola, Mahima M Katti, "Android Based Mobile Application Development and its security", International Journal of Computer Trends and Technology, Vol.3, Issue.3, pp.486-490, 2012.

[2]Limi Kalita, "Socket Programming", International Journal of Computer Science and Information Technologies, Vol.5, Issue.3, 4802-4807, 2014.

[3]Mahesh Kumar, Rakhi Yadav, "TCP and UDP Packets Analysis Using Wire shark", International Journal of Science, Engineering and Technology Research (IJSETR), Vol.4, Issue.7, 2015.

[4]Prathap M., A.S. Thanamani, "Real-time Packet Behavior Response in Socket Application", International Journal of Computer Sciences and Engineering, Vol.5, Issue.5, pp.143-146, 2017.

[5]Mayur Khatpe, Brij Patel, Tushar Jain and Varun Shah, "Remote Controlled Home Automation Using Android application via Wi-Fi Connectivity", International Journal of Computer Sciences and Engineering, Vol.4, Issue.4, pp.191-194, April, 2016.

[6]P Joby, Abby. (2016). Socket Programming WIFI Chat APP For Android Smartphone. 10.13140/RG.2.1.3692.2483.

Author's Profile

Shalabh Agarwal is an Associate Professor in the Department of Computer Science, St. Xavier's College, Kolkata. He is also the Director, Computer Centre & Central Computing Facilities and Head of the Department of Computer Science at the College. His research Interests are Green Computing, Pervasive Computing, Software Engineering, IOT and Internet security.

Sayantika Sengupta is pursuing Bachelor in Computer Science from St. Xavier's College (Autonomous), Kolkata. She has currently completed third year of the course. She is interested in higher studies in the field of computer.

Swarnali Sanyal is pursuing Bachelor in Computer Science from St. Xavier's College (Autonomous), Kolkata. She has currently completed third year of the course. She is interested in higher studies in the field of computer.

Naireeta De is pursuing Bachelor in Computer Science from St. Xavier's College (Autonomous), Kolkata. She has currently completed third year of the course. She is interested in higher studies in the field of computer.

