Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 3, March 2015, pg.732 – 738

RESEARCH ARTICLE

Android based PC Remote Control using Wi-Fi

Damini S. Jawale[1], Mahesh J. Deshmukh[2], Shruti R. Joshi[3]

[1]Department of Computer Engineering, Pune University, India

[2]Department of Computer Engineering, Pune University, India

[3]Department of Computer Engineering, Pune University, India

[1] damini7766@gmail.com; [2] mahesh15dn@gmail.com; [3] shrutijoshi269@gmail.com

Abstract— In a day to day life, there are various ways of interaction between Android mobile phones, which are used to communicate with users of electronic devices like laptops or PCs. Generally, user selects the option from an android application installed on their Android phone and send required commands to Laptop. More specifically, in this system, there is point to point interconnection between Android mobile and laptop. Basically it follows client-server architecture, in which client communicates with server using Wi-Fi. Firstly, user needs to execute server-side application and after that he is supposed to login (the user must be authorized). Then only, user can execute client application. For executing client application user needs current IP address of server. The client-side application is in Android and the server-side application is in Java. Sockets are used at both the ends i.e.; at server-side and client-side. The system provides user different options like Mouse Mode, Windows media player, MS PowerPoint, Image Viewer, Typing Mode, Utility Mode and many other commands. Some of these processes can communicate with other processes on the basis of InterProcess Communication Mechanism.

Keywords— Inter Process Communication, socket programming, Wi-Fi, Java, Android Mobile Phone

I. Introduction

In this 21st century, the maximum population all over the world has electronic devices like Android mobiles, tablet PCs, Notes, iPhones around them, these devices are said to be intelligent electronic devices. Wi-Fi is the wireless fidelity, which is largely used for connecting those electronic devices with each other wirelessly. So, now-a-days, Wi-Fi is widely accepted technology everywhere having range 30metres. The developed system called Android based PC Remote Control; user can use this application by connecting his laptop and Android phone using Wi-Fi. For server-side application java is used because everything in the java is open like from the language, to the standards, to the core libraries, to the virtual machine and the development tools. Since open source mindset is so pervasive in java world, present at its core and tested at enterprise level. And the mobile application is in android which is largest installed base of any mobile platform and growing fast every day. Android gives the best platform for creating apps for android users everywhere as well as an open market place for distributing those apps. To the specific, here for controlling laptop using Android phone user is expected to follow some initial steps. Firstly, user needs to execute server-side application which is developed in Java and after that he is supposed to login with the valid Username and Password. If either the username or password does not matched with the database, then that user can not be treated as valid one. Secondly, user should execute client application which will be there in Android phone. For executing client application user needs current IP address of server system. When IP address is entered in its field, user should click on connect button. The new window opens having client login. In the same way as server, user needs to login with valid username and password. After successful login, the major part of connection establishment is done, now user can see list of multiple options that the given system is providing, such as Mouse Mode, Typing Mode, Windows media player, MS PowerPoint, Image Viewer, explicitly provided Turn off option and various other commands are provided in the utility mode.

II. System Architecture

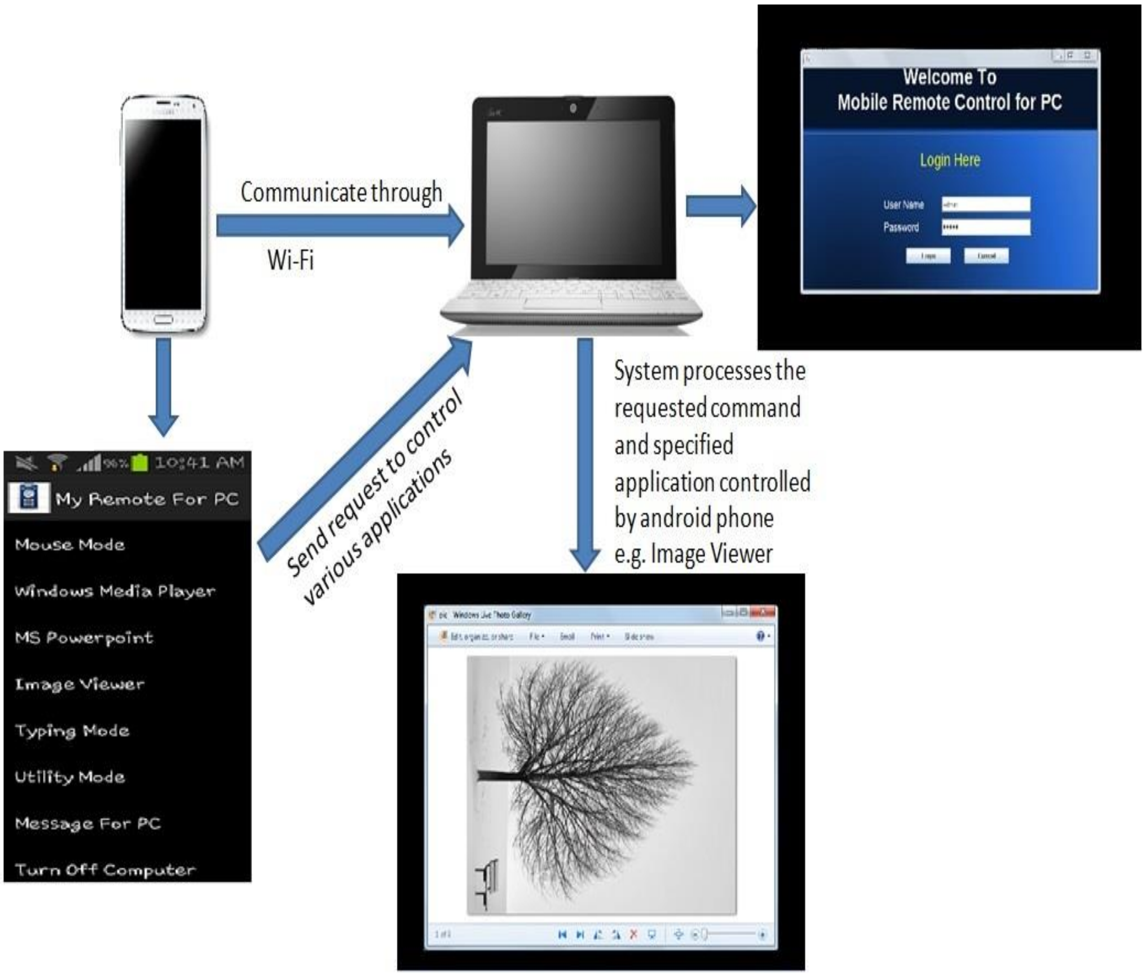

Fig.1 System Architecture

The system is based on client-server architecture. Client and server communicates through Wi-Fi. After connection establishment, firstly user needs to login at server system. After successful login, user needs a IP address of server system and then he has to login at client system. On successful login, user can control various applications provided in list. Server system is in Java and Client system is in Android. For example, when user controls the MS power point from his android phone then the request is sent to the server system, server system accepts that request and process that request to control the power point. On background, for processing this request server system uses keyboard shortcuts and this is done through Inter process Communication (IPC). This process is similar to all other operations like Windows Media Player, MS power point and Message for PC. For mouse mode, the x-y co-ordinates of touch pad are sent to the system then system accepts that co-ordinates and system moves the mouse according to that co-ordinates. In utility mode, the valid command is sent to the system i.e. the executable command, by receiving that command system executes that commands and opens the requested application.

III. Algorithm and Strategies Used

A. Robot

"This class is used to generate native system input events for the purposes of test automation, self-running demos, and other applications where control of the mouse and keyboard is needed. The primary purpose of Robot is to facilitate automated testing of Java platform implementations."

"Using the class to generate input events differs from posting events to the AWT event queue or AWT components in that the events are generated in the platform's native input queue. For example, Robot.mouseMove will actually move the mouse cursor instead of just generating mouse move events." Basically, a Robot object makes it possible for your program to temporarily take over control of the mouse and the keyboard. However, this may not be possible on all operating systems. Note that some platforms require special privileges or extensions to access low-level input control. If the current platform configuration does not allow input control, an AWTException will be thrown when trying to construct Robot objects."Sun goes on to provide some examples of cases where an AWTException might be thrown. A runaway Java Robot object has the ability to wrest control away from the human user, so you need to be a little careful. For example, if you allow your Java Robot program to go into an infinite loop, making mouse moves, clicking the mouse, and entering keystrokes, you may find that the only practical way to regain control of your computer is to either turn off the power or press the reset button to force your computer to restart.

(According to current planning, this miniseries on the Robot class will consist of three lessons. This first lesson will demonstrate the low-level nature of the behaviour of an object of the Robot class. This will be accomplished by showing you how to create a Java robot that can manipulate other non-Java programs, such as Windows Notepad and Internet Explorer. The second lesson in the miniseries will show you how to use a robot to perform automatic testing on a Java GUI. The third lesson will show you how to write an animated robot program to provide a visual demonstration of the use of a Java GUI.

B. How to use robot

RateJava.awt.Robot class is used to take the control of mouse and keyboard. Once you get the control, you can do any type of operation related to mouse and keyboard through your java code. This class is used generally for test automation. This sample code will show the use of Robot class to handle the keyboard events.

Example:

If you run this code and open a notepad then this code will write hi budy in the notepad.

```
Import java.awt.AWTException;
Import java.awt.Robot;
Import java.awt.event.KeyEvent;

public class RobotExp{

public static void main(String[]args)
 {
        try{
                        Robot robot =newRobot();// Creates the delay of 5 sec so that you can
                                                  open notepad before
                         Robot start writtingrobot.delay(5000);
                        robot.keyPress(KeyEvent.VK_H);
                        robot.keyPress(KeyEvent.VK_I);
                        robot.keyPress(KeyEvent.VK_SPACE);
                        robot.keyPress(KeyEvent.VK_B);
                        robot.keyPress(KeyEvent.VK_U);
                        robot.keyPress(KeyEvent.VK_D);
                        robot.keyPress(KeyEvent.VK_Y);
                }
                catch(AWTException e)
                 {
                        e.printStackTrace();
                }
}
}
```

C. java.awtClass Robotjava.lang.Objectjava.awt.Robotpublic class Robotextends Object

This class is used to generate native system input events for the purposes of test automation, self-running demos, and other applications where control of the mouse and keyboard is needed. The primary purpose of Robot is to facilitate automated testing of Java platform implementations. Using the class to generate input events differs from posting events to the AWT event queue or AWT components in that the events are generated in the platform's native input queue.
For example, X-Window systems will throw the exception if the XTEST 2.2 standard extension is not supported (or not enabled) by the X server. Applications that use Robot for purposes other than self-testing should handle these error conditions gracefully.

D. Constructor and Description

Robot() - Constructs a Robot object in the coordinate system of the primary screen. ***Robot(GraphicsDevicescreen)*** - Creates a Robot for the given screen device.

E. public abstract class KeyAdapter extends Object implements KeyListener

An abstract adapter class for receiving keyboard events. The methods in this class are empty. This class exists as convenience for creating listener objects. Extend this class to create a KeyEventlistener and override the methods for the events of interest. (If you implement the KeyListenerinterface, you have to define all of the methods in it. This abstract class defines null methods for them all, so you can only have to define methods for events you care about.) Create a listener object using the extended class and then register it with a component using the component's addKeyListenermethod. When a key is pressed, released, or typed, the relevant method in the listener object is invoked, and the KeyEvent is passed to it.

F. How to Write a Key Listener

Key events indicate when the user is typing at the keyboard. Specifically, key events are fired by the component with the keyboard focus when the user presses or releases keyboard keys. For detailed information about focus, see How to Use the Focus Subsystem.

Note: To define special reactions to particular keys, use key bindings instead of a key listener. For further information, see How to Use Key Bindings. Notifications are sent about two basic kinds of key events:*.The typing of a Unicode character*.The pressing or releasing of a key on the keyboard. The first kind of event is called a key-typedevent. The second kind is either a key-pressed or key-releasedevent.

In general, you react to only key-typed events unless you need to know when the user presses keys that do not correspond to characters. For example, to know when the user types a Unicode character ??? whether by pressing one key such as 'a' or by pressing several keys in sequence ??? you handle key-typed events. On the other hand, to know when the user presses the F1 key, or whether the user pressed the '3' key on the number pad, you handle key-pressed events.

IV. Flow of Proposed System

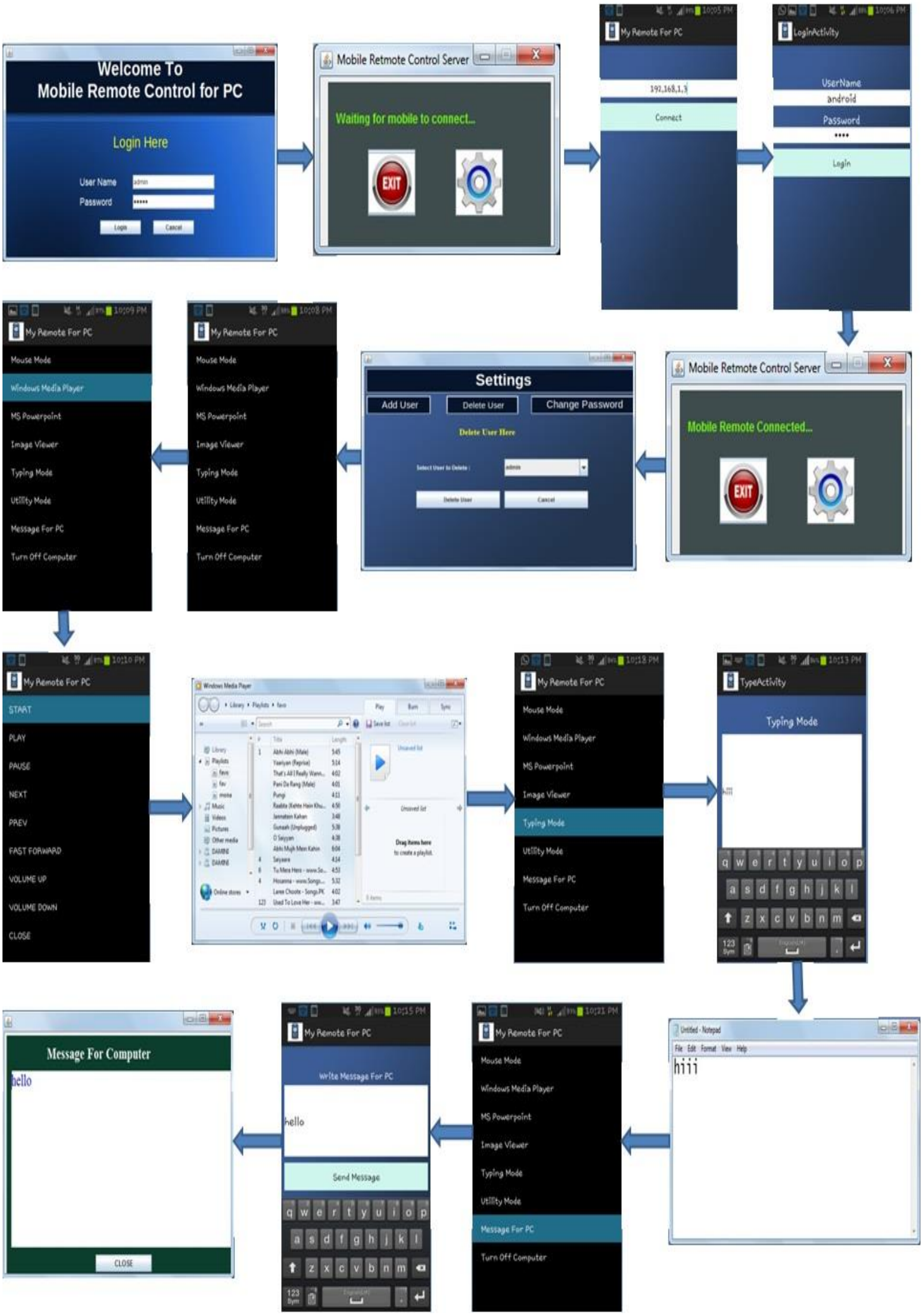

Fig.2 Flow of Proposed system

V. Mathematical Model

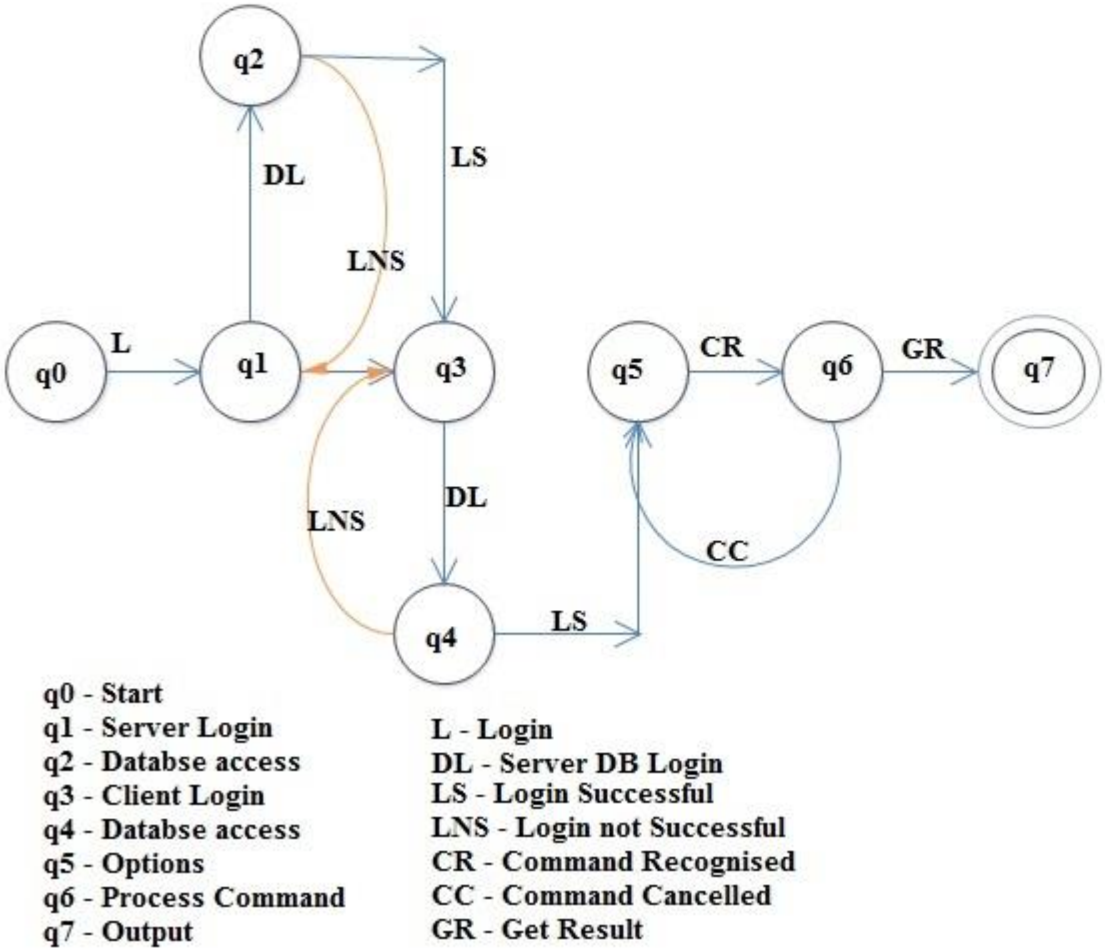

Fig.3 Mathematical Model

VI. Conclusion

Thus, from all discussion this paper proposed the system which controlled the various applications of the running system through android smart phone. The overall requirement of the hardware and software for implementing the system was very low as the system was implemented in Java and Android and the hardware required was near about zero. The system included the implementation of two different technologies and the communication of different processes through sockets. The implemented system is useful in various fields

Acknowledgement

We have a great pleasure in presenting the paper “Android based PC Remote Control using Wi-Fi” under the guidance of Prof. Mr. Rahul Mandlik. We would also like to thanks GES RHS COE MSR for providing all required facilities.

REFERENCES

[1] “Mobile Web-Based System for Remote-Controlled Electronic Devices and Smart Objects “, Jordán Pascual Espada & Vicente García-Díaz & Rubén González Crespo & Oscar Sanjuán Martínez & B. Cristina Pelayo G-Bustelo & Juan Manuel Cueva Lovelle, Published online: 28 May 2014, Springer Science+Business Media New York 2014

[2] Angel Gonzalez Villan, and Josep Jorba Estev, “Remote Control of Mobile Devices in Android Platform” IEEE transactions on Mobile

[3] "Remote computer access through Android mobiles," Jaya Bharathi chintalapati, Srinivasa Rao , IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 3, September 2012

[4] Chaitali Navasare, Deepa Nagdev and Jai Shree, "PocketDroid – A PC Remote Control", 2012 International Conference on Information and Network Technology (ICINT 2012) IPCSIT vol. 37 (2012) © (2012) IACSIT Press, Singapore

[5] Sareddy Deepthi, A. Vasanthi, "Mobile Phone Controlling PC-A Review", IJCSMC, Vol. 1, Issue. 1, December 2012

[6] Anjumara Inamdar, Heena Aggarwal, Sayali Kadam, Mayuri Kadhane, " COMPDROID -Remote Desktop Access through Android Mobile Phone", International Journal of Science and Modern Engineering (IJISME) ISSN: 2319-6386, Volume-2, Issue-1,

[7] Qadeer, M.A.; Agrawal, R.; Singhal, A.; and Umar, S, "Application Remote Control Using Bluetooth",International conferences on Advanced Computer Control, 2008.IEEE DOI 10.1109/

[8] Yenel Yildirim and Ibrahim Korpeoglu, "PocketDrive: A System for Mobile Control of Desktop PC and its Applications Using PDAs"

[9] Yonggao Yang and Lin Li, "Turn Smartphones into Computer Remote Controllers", International Journal of Computer Theory and Engineering, Vol. 4, No. 4, August 2012

[10] SonamGavhane, RasikaPhanse, Monica Sadafule, B.W.Balkhande, "Remote Desktop on Mobile", International Journal of Innovations in Engineering and Technology (IJIET) Vol. 2 Issue 1 February 2013