

An Android-based Remote Desktop for IOS Platforms

Siew-Chin Chong

Faculty of Information Science & Technology, Multimedia University, Malaysia
E-mail: chong.siew.chin@mmu.edu.my

Boon-Keang Leong

Faculty of Information Science & Technology, Multimedia University, Malaysia
E-mail: 1081107980@student.mmu.edu.my

Wee-Kiat New

Faculty of Information Science & Technology, Multimedia University, Malaysia
E-mail: aven.wknew@yahoo.com

Yong-Jian Chin

Faculty of Information Science & Technology, Multimedia University, Malaysia
E-mail: yjchin@mmu.edu.my

Abstract— Thanks to globalization, mobile devices have become an inseparable entity of our daily life. We often expect our smart devices (mobile phone, tablet, portable media player) to possess the functionalities of a personal computer. As the technology is getting cheaper, owning multiple mobile devices, each for a specific purpose, is becoming the current trend. For instance, an Android smartphone to fulfill a user's communication needs on-the-go, an iPad could serve the user's reading hobby, and lastly, a laptop for productivity activities. As such, to switch among different devices could be the emerging problem of the current generation. With our proposed Android based remote control app, a user does not only able to control his Windows based office laptop, but he could access to his IOS based devices too. Besides click event and text input, this application also supports panning and zooming gesture inputs.

Index Terms— Remote, IOS, Android, Access, Operating System

I. Introduction

The growth of technology in mobile devices nowadays has increased radically, especially the applications in mobile phones. Compared to the cellular phones, mobile phones provide better facilities and conveniences. Users often expect mobile phone can be functioned as a personal computer. Recently, mobile phones which are running on an operating system either IOS or Android are dominated the mobile phones market sales [1]. In the fourth quarter of 2012, Apple

and Samsung together raised their worldwide smartphone market share to 52 percent from 46.4 percent in the third quarter of 2012 [2]. The increasing demand on mobile phone has made mobile phone to become a necessity in human life, rather than just a luxury or entertainment device. Every week, new application is designed for its own compatibility, make sure that the application is compatible to the particular mobile phone model and its specification.

There is always inflexibility issue for the users who are using their computers with wired or wireless keyboard and mouse. Users would find it inconvenient to access to their computers while they are sitting far away from the computers. A wireless keyboard and mouse has a fixed distance to control the computers. Hence, remote desktop application has come into the place to overcome this constraint. Remote desktop is one of the best features in mobile phones that allow users to access to remote computer system within either Wi-Fi network or 3G network. It allows users to operate the data that is being stored in the computer through the phone with just a few touches on the phone screen.

However, the existing remote desktop applications could not remotely access to different platforms such as IOS, one of the powerful operating systems in the current trend. This limitation creates problem for the users who own two different OS (Android and IOS) phones or tablets. This inspired the idea of developing the proposed application, which enables the Android users to remote access the files, photos and documents that stored in IOS devices, besides accessing to the windows-based platform. This application turns the Android mobile phone into a portable computer and access data easily just through the palm of the hand.

The rest of the paper is organized as follows. Some of the related Android applications are briefly reviewed in Section 2. The proposed system design is discussed in Section 3. In Section 4, the implementation process of the proposed system is elaborated. The testing scenario and result are reported in Section 5 and finally, the conclusion is given in Section 6.

II. Existing System

The world- wide sales of Android devices is estimated to grow drastically in the Year 2016, due to the fact that Smartphones have become truly pervasive in every aspect of an employee's life [3]. Since Android is an open source, it is inevitable to provide the opportunities for the Android developers to build or enhance the applications demanded in Android market. Therefore, the improvement and development of existing Android applications is common in Android platform nowadays. Remote desktop is the application that needs to be studied in this project. It allows users to access the computer or laptop from another device anytime anywhere. To better understand the remote access features and the demands of the current market trend, some of the existing remote desktop applications have been analyzed, with the comparison of their specification mentioned as below.

- i. Team viewer [4]: Team viewer is a remote desktop application which allows user to remote access computer by using Android devices. There are few features in this application. It supports transfer files from android mobile to any platforms of computer and vice versa. It can control remote computers using the multi-touch gestures such as left-click, right click, drag, drop, scroll wheel and zoom. Besides that, it also has complete keyboard control. It meets higher security standard 256 bits AES session encoding, 1024 bits RSA key exchange. User can effortlessly access computers behind firewalls and proxy servers. However there are several drawbacks from this application. This program might need fast and continuous internet connection to avoid lagging and it does not work through proxy servers such as the AUB proxy. Moreover, its business packages are quite expensive.
- ii. LogMeIn [5]: LogMeIn is another remote desktop application that supports Mac OS and Windows platform. Besides that, it has few features while connected to the personal computer which are move mouse on host computer, click, scroll and drag on host computer, zoom in and out via pinch or pan zoom navigation, access full onscreen keyboard, clipboard support (cut, copy & paste text across apps), save password for one-click access and stream sound from windows-based personal computers. The weaknesses of this application is the users need to pay for the file transfer and other features, and it is by default to share the computer

even when the user does not want people to access the computer.

- iii. Proxy Pro Gateway Edition [6]: This application supports remote computers with multiple Network Interface Controller (NIC) cards or multiple Internet Protocol (IP) address per NIC card. It uses Secure Socket Layer (SSL) certificates for possible authentication for proxy users trying to reach remote computer inside or outside the domain. This application uses highly efficient screen capture algorithms, and provides tools to limit bandwidth usage (and improve performance) over slow network links. However, this application does not support Android technology and audio features.
- iv. Remote Desktop Services [7]: Remote Desktop Services is one of the services that created by Microsoft windows that allow users to access the computer over network remotely by using Remote Desktop Protocol (RDP). It is commonly built into most of the versions of window. It transmits faster, as it sends data rather than images. However it cannot support multiple connections at once, and will use a lot of bandwidth, which affects the speed and responsiveness.

Based on the study on the listed applications, there is no application which is able to provide the privilege for Android users to remote access the Apple proprietary's operating system (IOS). This motivates the idea of designing a remote desktop application which is able to remote different platforms such as Android to IOS. Furthermore, the proposed prototype is able to perform various types of operations such as, text input, pinch gesture, zoom-in and out, and multi-touching. In addition, it also has different input modes such as, touch mouse pan mode, touchpad mode, desktop panning mode, trackball mode and D-pad pan mode. There are some features also being included in this proposed prototype which are support Wifi, 3G connection, able to remotely access IOS devices (Iphone, Ipad) and able to run through a secure network.

III. Proposed System Design

This section presents two important factors of the proposed application which are: Client Application and Server Application. The major components of the proposed application and some of the important algorithm are highlighted in detail.

3.1 The Client Application

The client side application developed using the Android SDK and installed on user devices as an .apk. Based on the platform using Android Software Development kit [7], this application was written using the Java Programming Language and was designed to

run on customized Linux-based virtual machine called Dalvik. The client application developed has an .apk extension. The interested users can download it from the developer's store and install directly to their mobile phone devices.

This section explains how the client application works and flows. First, the client application has to connect to the Wi-Fi connection so that it can connect to the server and run in parallel. The client application extracts the XML file from the .apk file to generate the main screen which is Graphical User Interface (GUI). The application GUI will then await an input from the user. After the action has been performed, it will send a command to the server. After it has been processed, the server sends back the action to application GUI and updates the input back to the user as expected. The process of client and server has to be worked in parallel, because client will not carry out its task if server does not own an operation.

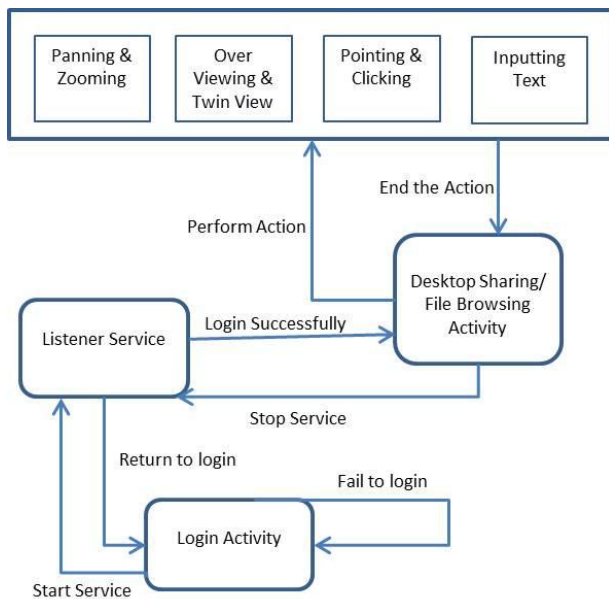


Fig. 1: Client Application Flow Diagram

Fig. 1 shows the program flow of client when it starts the service and performs actions. Client always send request to the listener when it wants to establish connection or start service. Before the client able to send request to the listener, it has to get the IP address and port number from the user. After the client and the server get connected successfully, the user can perform zooming, pointing, clicking and file browsing so that client not just observe the server's desktop passively but actively interact with it.

3.2 The Server Application

For the server side, the server is working parallel with client. The server has to create a socket and listen to the particular port. The server has to authenticate the

password before accepting the connection. Lastly, server will process the request and send the response to the client.

Fig. 2 tells that the program flow of server interacts with client. Client gets connected to the server with the server's IP address and password via Wifi or 3G. Server is always in listening mode until it gets request from the server. With the request sent from the client, the handshaking is established. The IP address of the server and the password transmits the first frame after handshaking is done.

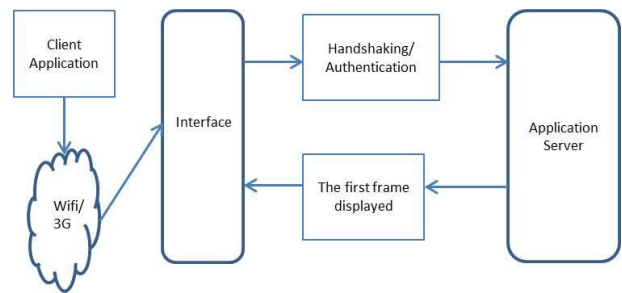


Fig. 2: Server Application Flow Diagram

3.3 Software Development Kit

To build applications for the Android platform, the developers use a Java based software development kit, which is the Android SDK. The Android SDK includes development tools, source codes for sample project, an emulator and the necessary libraries to create Android applications. Applications are run on Dalvik which is a custom-built virtual machine that runs on top of a Linux Kernel [8]. The libraries in the SDK using XML and the graphical input assist in building the graphical user interface of an Android application. This ensures that the scheme created by the developer is uniform for the application and between applications. View and ViewGroup objects are used to create the user interface in an Android application. View objects are the fundamental units of user interface expression on the Android platform [9]. Views have attributes which can be used to configure their appearance and behaviour. A ViewGroup is responsible for arranging other views. ViewGroups is also called layout managers. The base class for these layout managers is the android.view.ViewGroup class which extends the View Class [10].

3.4 Important Source Codes Implementation

The codes for each major activity in the client-side application are shown in the figures below. All of the source codes are written in Java programming language and run in Eclipse.

Fig. 3 shows the code of OnTouchEvent activity. OnTouchEvent will be called when a touch screen motion event occurs. When there is a drag motion event

occurs, users are no longer stay on the same coordinates and the `ChangeTouchCoordinatesToFrame` function will be called. This function is used for adjusting the

event location and setting location of the event. It ensures the scale of screen in the Android is similar to the scale of the screen in the platform.

```
public boolean onTouchEvent(MotionEvent e) {
    if (dragMode) {
        vncCanvas.changeTouchCoordinatesToFullFrame(e);
        if (e.getAction() == MotionEvent.ACTION_UP)
            dragMode = false;
        return vncCanvas.processPointerEvent(e, true);
    } else
        return super.onTouchEvent(e);
}

MotionEvent changeTouchCoordinatesToFullFrame(MotionEvent e)
{
    //Log.v(TAG, String.format("tap at %f,%f", e.getX(), e.getY()));
    float scale = getScale();

    // Adjust coordinates for Android notification bar.
    e.offsetLocation(0, -1f * getTop());

    e.setLocation(absoluteXPosition + e.getX() / scale, absoluteYPosition + e.getY() / scale);

    return e;
}
```

Fig. 3: onTouchEvent Activity

`OnCreateOptionsMenu` code is shown in Fig. 4. This `OnCreateOptionsMenu` will be called when the user has started to remote the desktop. Then, the

`vncCanvasActivity` which stored in menu folder in XML format will be called. This code is mainly about graphical layout.

```
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.vncCanvasActivityMenu, menu);

    if (vncCanvas.scaling != null)
        menu.findItem(vncCanvas.scaling.getId()).setChecked(true);

    Menu inputMenu = menu.findItem(R.id.itemInputMode).getSubMenu();

    inputModeMenuItems = new MenuItem[inputModeIds.length];
    for (int i = 0; i < inputModeIds.length; i++) {
        inputModeMenuItems[i] = inputMenu.findItem(inputModeIds[i]);
    }
    updateInputMenu();
    menu.findItem(R.id.itemFollowMouse).setChecked(
        connection.getFollowMouse());
    menu.findItem(R.id.itemFollowPan).setChecked(connection.getFollowPan());
    return true;
}
```

Fig. 4: Option Menu Bar

Fig. 5 shows two elements that appear in the Android Manifest which are `uses-sdk` and `uses-permission`. `MinSdkVersion` for this application is API 5 (Android 2.0), which means for those API level lower than API 5 will not compatible. Besides that, if users are running on API higher than `TargetSdkVersion`, the system may

enable the compatibility behaviour to make sure the application is running in a good manner. Moreover, this application is required to access to the internet, `android.permission.INTERNET` means to request permission so that the network socket can be opened.

```
<uses-sdk
    android:minSdkVersion="5"
    android:targetSdkVersion="8" />
<supports-screens />
<uses-permission android:name="android.permission.INTERNET"/>
```

Fig. 错误!文档中没有指定样式的文字。: Uses-SDK and Uses-permission Activity

IV. Implementation Process

The main objective of constructing this application is to remotely access the resources on IOS platform (iPad or iPhone) via Android phones or tablets. This

application enables the IOS proprietary applications sharing to Android users in a legal way. The system architecture of the proposed application is shown in Fig. 6.

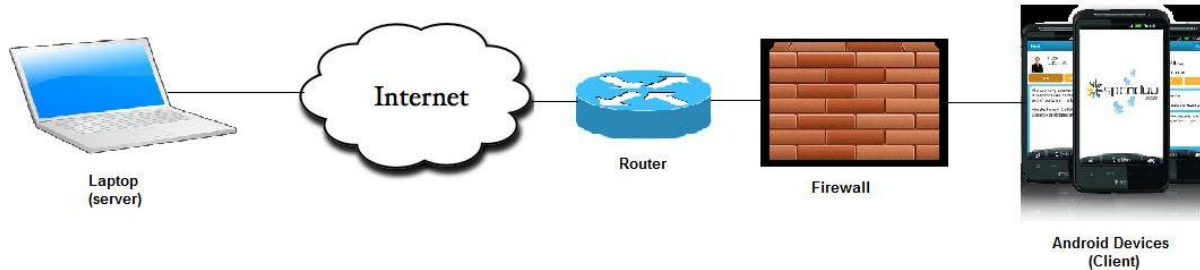


Fig. 6: System Architecture

4.1 Core Features

Cross platforms connection is the strength in this application. It is able to connect and access to computer, tablets or mobile phones. Several types of remote desktop activities are listed as below:

1. Remote Apple iPhone via emulator
2. Remote Apple iPad via Android mobile phone
3. Remote Window-based computer via emulator
4. Remote Window-based computer via Android mobile phone

desktop applications. In order to show the compatibility with the current applications, other platforms are tested as well: remote Window-based computer via emulator and via Android mobile phone. Fig. 7, Fig. 8 and Fig. 9 show the screenshots of connecting to different platforms through Android device or emulator. All these remote desktop activities are done through Wi-Fi connection, EDGE, GPRS or 3G. On the other hand, the server application has to be installed and executed on the targeted platforms in order to allow the access from the users. If the users connect over the internet such as 3G or connect from different networks, router has to be configured such that forward network communication is allowed.

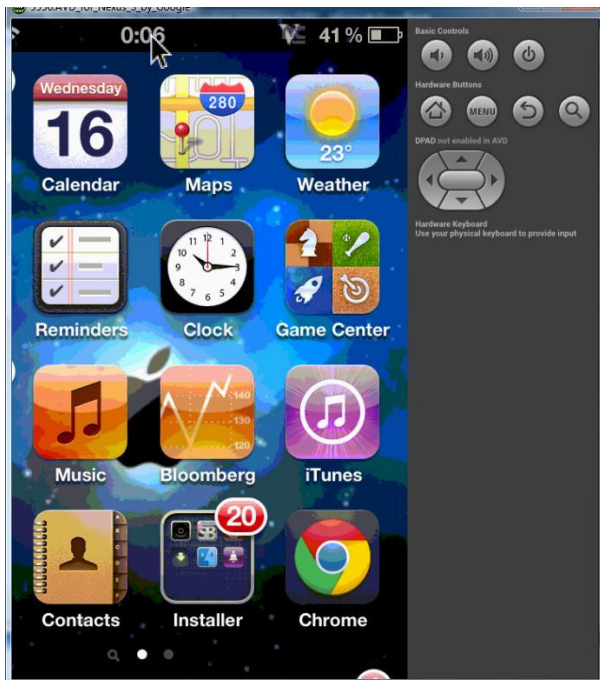


Fig. 7: Remote Apple iPhone Via emulator

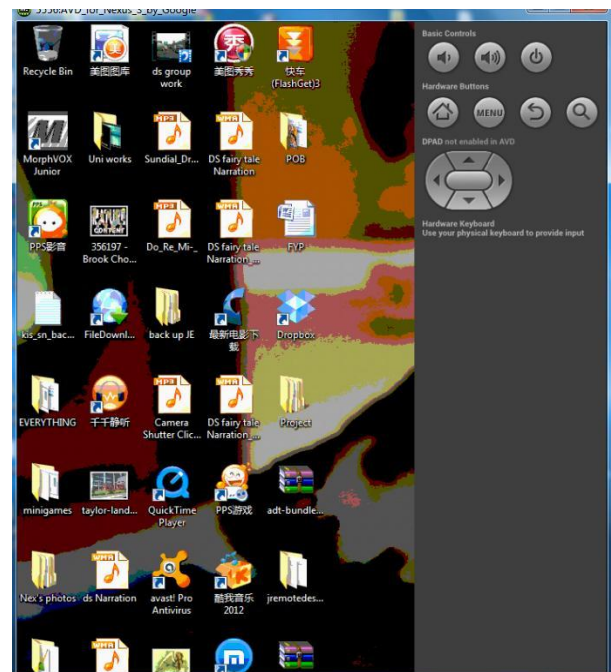


Fig. 8: Remote desktop via emulator

Remote access to Apple iPhone through Android device is our main focus in this application, which is the additional feature to further improve the current remote



Fig. 9: Remote desktop via Android mobile

4.2 Application Prototype

There are a few major actions which can be performed after the application has been launched on the device, which are:

1. IP address configuration page
2. Menu bar
3. Input text layout page
4. Mouse input mode menu

4.2.1 IP address configuration page

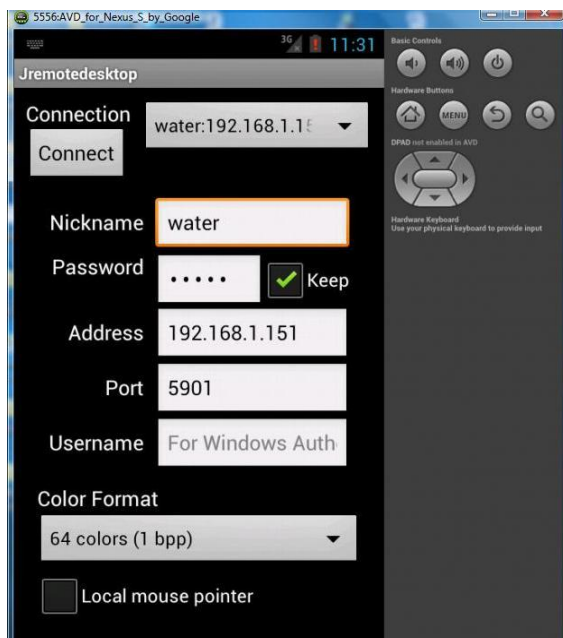


Fig. 10: IP address configuration

Fig. 10 shows the IP address login page via emulator. In order to connect to the server, users are required to insert some of the details, such as nickname, password, and local network address. If the client and the server are in different networks, users are required to do some configuration at the router (Detail explanation see Fig. 14). At last, users are required to insert the correct port number. The default port number will be 5900.

4.2.2 Menu bar

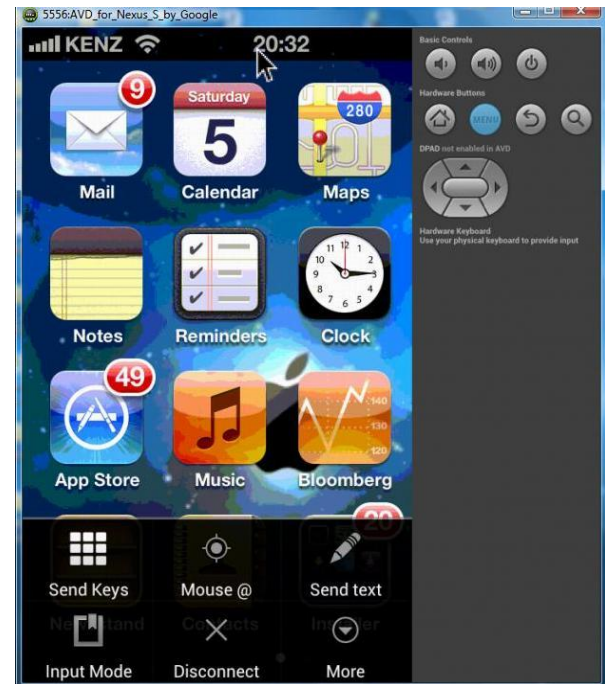


Fig. 11: Menu bar

Fig. 11 shows the menu bar of the application via emulator. After the users press the connect button, this client application will try to connect to the particular server. If the client is able to handshake and authenticate successfully, then menu bar will be displayed. With the menu bar, users can have different functions interact with the server but not only observe it passively.

4.2.3 Input text layout page

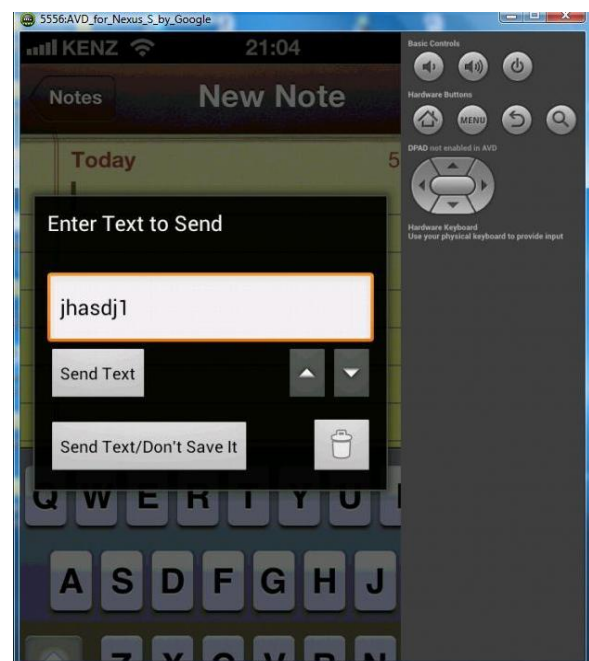


Fig. 12: Input text layout page

Fig. 12 shows the input text layout page via emulator. Users always interact with the server by sending some text to the server. In order to send text to the server, users will need to tap on the text box and then the keyboard will turn up. As such, users can send text to the server with symbols, numbers and alphabets.

4.2.4 Mouse input mode menu

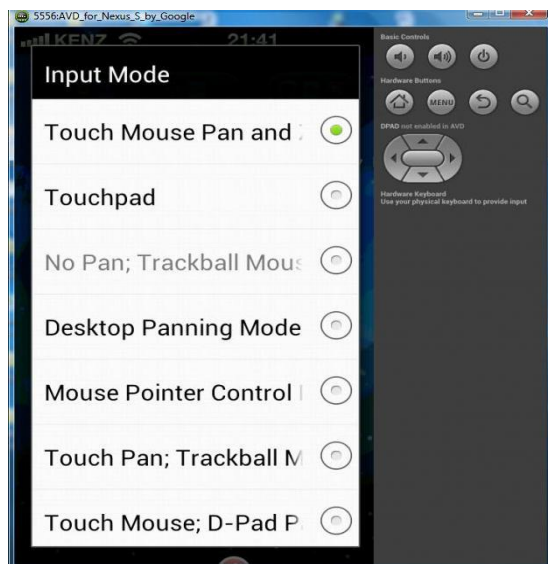


Fig. 13: Mouse input mode menu

Fig. 13 shows the mouse input mode menu via emulator. There are several mouse input modes for adapting the control of the device. Touch Mouse Pan and Zoom is the default mode. With this mode, users can pan display and control the mouse by using touch screen and gestures. Users pan by dragging on the screen and select the particular file by tapping on it. Touchpad Mode is Gesture-Based Interface. It works almost like the touchpad in the laptop. Touch Mouse D-Pad Pan uses directional pad to pan the display. This can be done by touching the screen to send a mouse click and; touching and sliding to send a mouse drag.

4.3 Router Configuration

DDNS (Dynamic Domain Name Server) is a service allows dynamic IP address to a static hostname in any of the many domains, allowing you DSL router to be more easily accessed from various locations on the internet.

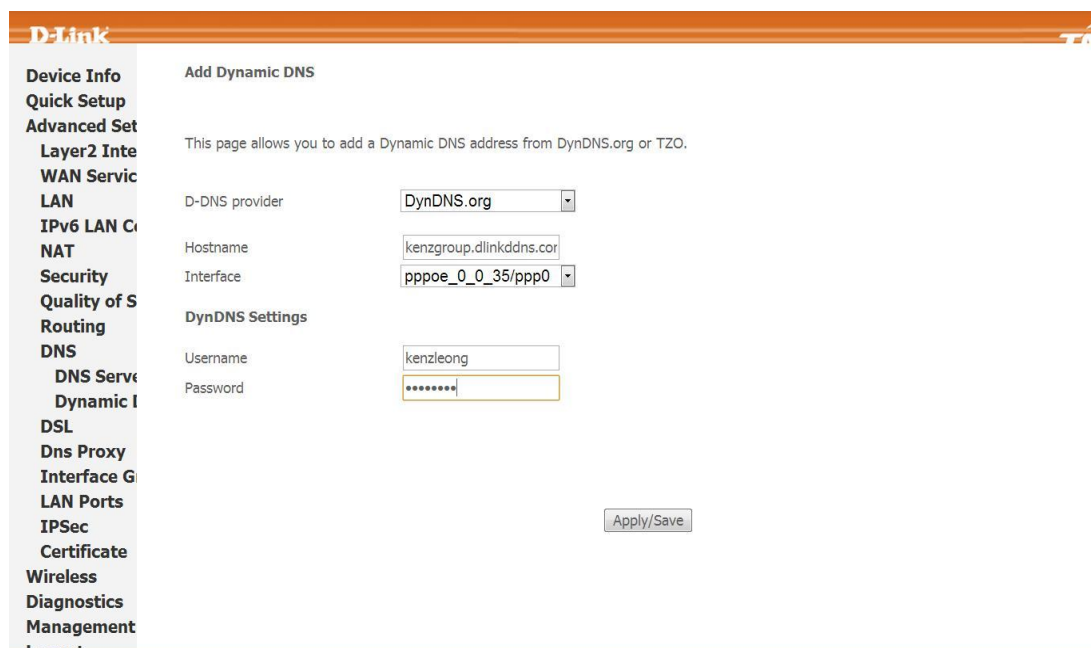


Fig. 14: DDNS Configuration Page

Fig. 14 explains the configuration of DDNS. First, users have to go www.dlinkddns.com to register an account for free hostname. After registering is successful, users are required to insert the username, password and hostname in this DDNS configuration page. Then, click on Apply once finished inserting.

Virtual server setup or port forwarding setup allows incoming traffic from WAN side to the internal server with private IP address on the LAN side.

NAT -- Virtual Servers Setup

Virtual Server allows you to direct incoming traffic from WAN side (identified by Protocol and External port) to the Internal server with private IP address on the LAN side. The Internal port is required only if the external port needs to be converted to a different port number used by the server on the LAN side. A maximum 32 entries can be configured.

Server Name	External Port Start	External Port End	Protocol	Internal Port Start	Internal Port End	Server IP Address	WAN Interface	Remove
RealVNC	5900	5900	TCP	5900	5900	192.168.1.150	ppp0	<input type="checkbox"/>
AirVideo	45631	45631	TCP	45631	45631	192.168.1.150	ppp0	<input type="checkbox"/>
RealVNC	5901	5901	TCP	5901	5901	192.168.1.151	ppp0	<input type="checkbox"/>
remotedesktop	5902	5902	TCP	5902	5902	192.168.1.153	ppp0	<input type="checkbox"/>

Fig. 15: Virtual server configuration page

Fig. 15 shows the virtual server configuration. Users are required to insert the server's port number into the external port start and internal port start, 5900, 5901 or 5902. Furthermore, users have to specify the server IP address. The correctness of the IP address and port number are very important so that the incoming traffic can reach to the particular server.

V. Result

In order to evaluate the stability and the validity of the application, several system testings have been conducted. The testing results are shown in Table 1. The performance of the application is guaranteed through the testing results and the application is worked well in different scenarios. It is proven that it is possible to access to different platforms anywhere anytime through an Android device, just like a portable computer on hand, with the 3G or Wifi.

Table 1: Test Cases

Use Case	Functions being tested	Inputs	Expected Outputs	Outcome
Application Running	Launch client application on Android device to IOS device and connect to the Wifi connection	User inserts IP address and port number then clicks on the connect button immediately	IP address is valid and connected to the server successfully	Successful
Application Running	Launch client application on emulator to IOS device and connect to the Wifi connection	User inserts IP address and port number then clicks on the connect button immediately	IP address is valid and connected to the server successfully	Successful
Application Running	Launch client application on Android device to Window-based computer and connect to the Wifi connection	User inserts IP address and port number then clicks on the connect button immediately	IP address is valid and connected to the server successfully	Successful
Application Running	Launch client application on emulator to Window-based computer and connect to the Wifi connection	User inserts IP address and port number then clicks on the connect button immediately	IP address is valid and connected to the server successfully	Successful
Application Running	Connect to the server by using 3G connection.	User inserts the DNS name and port number then clicks the connect button	IP address is valid and connected to the server successfully	Successful
Application Running	Resolution setting	Changing to different resolution settings	The resolution settings work well. Resolution 256 would be the best for the display	Successful
Application Running	Colour format	User selects 24 bits colour (4 bits per pixel)	Colour sharp, can see more clearly on the object. The cursor moves slowly	Successful
Application Running	Colour format	User selects 8 colour (1 bits per pixel)	Noise, blur and cannot see the wallpaper on the screen clearly	Successful

Use Case	Functions being tested	Inputs	Expected Outputs	Outcome
Application Running	Menu bar	Pull the menu bar from bottom of the screen	Menu bar properly displayed	Successful
Application Running	Desktop panning	Touch on the screen to move the desktop display, without moving the cursor	The desktop display is viewed nicely	Successful
Application Running	Send text to the server	Tap on the textbox, the keyboard will be popped out users can send any text by tapping the keyboard.	The text sent by the client will appear on the server machine.	Successful
Application Running	Multi-scrolling and Pinch Gesture	User applies two fingers and pinch on the screen	Both multi-scrolling and pinch gesture works well	Successful
Application Running	Touch Mouse Pan and Zoom	User applies two fingers and pinch on the screen	Zoom into the screen and zoom out from the screen	Successful
Application Running	Touch Mouse, DPad Pan	User applies double tap on the screen.	Double left-clicked function is enabled	Successful

VI. Conclusion

The rapid growth of the technology nowadays is driving the users to higher expectation towards the functionality of the mobile phones. Different operating systems of the mobile phones would allow different applications to be installed. Currently, the famous operating systems such as IOS, Android and Windows have played an important role in the advancement of mobile applications. This proposed application fulfills the demand of the market trend in the sense to control IOS device from the Android phone or tablet. It is aimed to provide a simple and convenience way for the users to have everything in one device on hand for the purpose of working, entertainment or studying. The greatest achievement from the development of this application is the flexibility of remote access to different platforms through the Android device, especially the Apple iPhone or iPad. Besides that, Wi-Fi and 3G functions are successfully implemented in the application. Furthermore, the proposed application is completed with various types of operations such as, text input, pinch gesture, zoom-in and out, and multi-touching. In addition, different types of input modes have been included such as touch mouse pan mode, touchpad mode, desktop panning mode, trackball mode and D-pad pan mode. As conclusion, this application is claimed to successfully perform real-time and responsive communication across different platforms. Yet it is enhanced to allow the sharing of IOS proprietary applications to Android users in a legal way.

Acknowledgments

The authors would like to express their thankfulness to the anonymous reviewers for their careful reading of this paper and for their valuable comments. This work was supported by Multimedia University, Malaysia.

References

- [1] A. Livingston. Smartphones and other Mobile Devices: the Swiss army knives of the 21st Century. *Educase Quarterly (EQ)*, 2004, 27(2).
- [2] Gartner. Gartner Newsroom: Press Release 2013 [Online]. Retrieved from: <http://www.gartner.com/newsroom/id/2335616> (Accessed 16 February 2013)
- [3] Gartner. Gartner Newsroom: Press Release 2012[Online]. Retrieved from: <http://www.gartner.com/newsroom/id/2227215> (Accessed 16 February 2013)
- [4] Teamviewer. Retrieved from: <http://www.teamviewer.com/en/products/remotecontrol.aspx> (Accessed 16 February 2013)
- [5] LogMeIn. Retrieved from: <https://secure.logmein.com/products/ignition/android/features.aspx> (Accessed 16 February 2013)
- [6] Proxynetworks. Retrieved from: <http://www.proxynetworks.com/products/proxy-gateway> (Accessed 16 February 2013)
- [7] Remote Desktop Services. Retrieved from: [http://msdn.microsoft.com/en-us/library/windows/desktop/bb892075\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/bb892075(v=vs.85).aspx) (Accessed 20 March 2013)
- [8] Webopedia, Introduction to Mobile Devices: Android SDK [Online]. Retrieved from: http://www.webopedia.com/quick_ref/mobile_OS.asp (Accessed 12 February 2013)
- [9] Webopedia. Android SDK. Retrieved from http://www.webopedia.com/TERM/A/Android_SDK.html (Accessed 12 February 2013)
- [10] Android Developers. The Guide. Retrieved from <http://developer.android.com/training/basics/firstapp/building-ui.html> (Accessed 12 February 2013)

- [11] Lars Vogel. Vegella.com. Retrieved from Android Development Tutorial <http://www.vogella.com/articles/Android/article.html> (Accessed 12 February 2013)

Authors' Profiles



Siew-Chin Chong (1980 -): Lecturer of Faculty of Information Science & Technology in Multimedia University, Malaysia, majoring in Software Engineering and is currently working towards mobile applications and pattern

recognition projects.



Boon-Keang Leong (1989 -): Student for bachelor degree from Faculty of Information Science & Technology in Multimedia University, Malaysia, majoring in Data Communication & Networking.



Wee-Kiat New (1989 -): Research Assistant at Department of Electrical Engineering, Faculty of Engineering, University Malaya. His research directions include Mobile Computing and Wireless Network.



Yong-Jian Chin (1985 -): Lecturer of Faculty of Information Science & Technology in Multimedia University, Malaysia, majoring in Information Systems Engineering. His research interests are multimodal biometrics and mobile computing.