MOBMOUSE APPLICATION

MUHAMMAD SYAHMI BIN MOHD
KHAIRUL MUZAMMIL

Bachelor of Computer Science (Computer System
& Networking) With Honours

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name	: MUHAMMAD SYAHMI BIN MOHD KHAIRUL MUZAMMIL
Date of Birth	: 20 OCTOBER 1995
Title	: MobMouse Application
Academic Session	: 1819-1

I declare that this thesis is classified as:

□	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
□	RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☑	OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

(Student's Signature)

951020-10-6173

New IC/Passport Number
Date:10 JANUARY 2019

(Supervisor's Signature)

SYIFAK BINTI IZHAR HISHAM

Name of Supervisor
Date:10 JANUARY 2019

UMP
Universiti
Malaysia
PAHANG
Engineering • Technology • Creativity

SUPERVISOR’S DECLARATION

I/We* hereby declare that I/We* have checked this thesis/project* and in my/our* opinion, this thesis/project* is adequate in terms of scope and quality for the award of the degree of *Doctor of Philosophy/ Master of Engineering/ Master of Science in ……………………………..

____

(Supervisor’s Signature)

Full Name : Syifak Binti Izhar Hisham

Position : SUPERVISOR

Date : 10 JANUARY 2019

Universiti Malaysia PAHANG
Engineering • Technology • Creativity

STUDENT’S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

____

(Student’s Signature)

Full Name : MUHAMMAD SYAHMI BIN MOHD KHAIRUL MUZAMMIL

ID Number : CA15154

Date : 10 JANUARY 2019

MOBMOUSE APPLICATION

MUHAMMAD SYAHMI BIN MOHD KHAIRUL MUZAMMIL

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Bachelor of Computer Science (Computer System & Networking) with Honours

Faculty of Computer Systems & Software Engineering
UNIVERSITI MALAYSIA PAHANG

January 2019

ACKNOWLEDGEMENTS

All the praise and thanks be to Allah SWT, the Most Beneficent and the Most Merciful and the Holy Prophet Muhammad SAW, peace upon him.

First, I would like to take this grateful opportunity to express my appreciation to all the peoples that have been contributing and supporting me in order to successfully complete the PSM 1 & 2 Project.

In particular, I want to give an appreciation to my honorable Supervisors, Dr. Izzeldin Ibrahim Mohamed Abdelaziz and Dr. Syifak Binti Izhar Hisham for their guidance, advices, encouragements, supports and motivations during development of this project. We hope in this final documentation, all the requirements have been fulfilled and satisfy both Dr. Izzeldin Ibrahim Mohamed Abdelaziz and Dr. Syifak Binti Izhar Hisham. Thank you in advance.

ABSTRAK

Zaman tanpa wayar berkembang pesat berikutan dengan peningkatan teknologi. Adakalanya kita sebagai pengguna telah menghadapi beberapa kejadian yang tidak diingini yang boleh menggangu kita daripada kerja dan produktiviti kita. Misalnya, pad sentuh computer riba kita di mana ia boleh menjadi masalah akibat daripada masalah perkakasan atau perisian yang memaksa kita untuk meminjam sama ada komputer riba atau tetikus orang lain, atau membawa tetikus sandaran sendiri yang meningkatkan jumlah peranti yang dibawa, mempunyai tetikus tanpa wayar bersama bateri yang telah mati dan untuk bebas daripada kabel-kabel. Ini akan menjejaskan masa dan produktiviti semasa bekerja. Ideanya adalah untuk membangunkan sebuah applikasi mudah alih yang dipanggil applikasi MobMouse yang mampu berfungsi seperti pad sentuh dengan menggunakan skrin telefon pintar. Applikasi MobMouse boleh berfungsi dengan baik dengan kehadiran rangkaian tanpa wayar seperti WiFi, sebuah PC untuk berkomunikasi dengan telefon pintar yang bersambung dalam rangkaian yang sama, sebuah telefon pintar Android yang telah dipasang dengan Android OS versi 4.0 “Ice Cream Sandwich” dan keatas serta satu port UDP khas yang telah dibuka bagi membolehkan komunikasi antara PC dan telefon pintar. Metodologi yang digunakan ialah “Rapid Application Development (RAD)” yang mengambil kesempatan terhadap prototaip dan penggunaan semula kod yang menyebabkan pengurangan masa diperlukan untuk membangunkan projek ini. Keputusan yang dijangkakan daripada applikasi MobMouse akan dicatatkan. Akhir sekali, antara sebab untuk membangunkan applikasi MobMouse adalah untuk mengatasi masalah pad sentuh komputer riba yang rosak, untuk mengatasi masalah tetikus tanpa wayar yang kehabisan bateri, untuk mengelakkan pengurusan kabel dan untuk mengurangkan bilangan peranti yang dibawa dan untuk menentukan sama ada applikasi MobMouse selari dengan objektif dan keperluan yang ditetapkan, untuk senaraikan kekangan dan batasan yang dijumpai semasa pembangunan applikasi MobMouse dan untuk mencadangkan sebarang peningkatan yang boleh memberi manfaat kepada aplikasi MobMouse pada versi masa hadapan.

ABSTRACT

The age of wireless is growing rapidly along with the growing technologies. There are times where we as the users that have faced some unwanted incident that could interrupt us from our work and productivity. For instance, our laptop touchpad where it can become malfunction due to hardware or software issues which forced us to either borrow other people's laptop or mouse, or bring own backup mouse that increase the number of devices carried, having a wireless mouse with dead battery and to be free from cables. This will affect our time and productivity when working. The idea is to develop a mobile application called MobMouse Application which functions as a touchpad by using smartphone's screen. The MobMouse Application can work well with the present of a wireless network such as WiFi, a PC to communicate with the smartphone that are connected in the same network, an Android smartphone with Android OS version 4.0 Ice Cream Sandwich and above installed and a dedicated UDP port opened on the PC to allow communication between the PC and the smartphone. The methodology used is Rapid Application Development (RAD) that takes advantages of prototyping and code reuse which reduce the development time of the project. The expected results from the MobMouse Application are noted. Lastly, the reason for developing MobMouse application is to overcome the issue of having a malfunction laptop's touchpad, to overcome a dead battery wireless mouse, to avoid cable management and to reduce the amount of device carried and to determine whether the MobMouse application aligns with the objectives and requirements, to list down the limitations or constraints discovered during MobMouse application's development and to suggest any future improvement which can benefit the MobMouse application in future versions.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

GHz	Gigahertz
RM	Ringgit Malaysia

LIST OF ABBREVIATIONS

WiFi	Wireless Fidelity
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
PC	Personal Computer
USB	Universal Serial Bus
OS	Operating System
APK	Android Package
SDK	Software Development Kit
JDK	Java Development Kit
API	Application Programming Interface
BASIC	Beginners All Purpose Symbolic Instruction Code
IEEE	Institute of Electrical and Electronics Engineers
IDE	Integrated Development Environment
IR	Infrared
TV	Television
IP	Internet Protocol
RM	Ringgit Malaysia
JRE	Java Runtime Environment
Li-ion	Lithium-ion
GUI	Graphical User Interface
UI	User Interface
GPS	Global Positioning System

CHAPTER 1

INTRODUCTION

1.1 Introduction

Nowadays, technology has become a cohesive part in people’s lives. It has and still continues to affect many fields of daily life and has permitted better social communication, luxury of transportation, the capability to treat in entertainment and media as well as helping in the progress of medication field. The formation of many devices such as mobile phones and computers have initiated many people to depends on technology to be in touch with their friends and to keep information such as pictures, movies, documents, and music.

By 2018, the amount of smartphone users throughout the globe is estimated approximately 2.53% of the whole people in the world. In billion terms, the amount of smartphone user is 2.5 billion globally and roughly 80 percent of them are Android users while the rest are iOS and Windows Phone. With that amount of smartphone users in current years, it is expected to increase in the upcoming years. These numbers are collected from Statista website. (Statista, 2016).

WiFi is a technology which uses radio waves for providing network connectivity. WiFi connection is established using a wireless adapter to create ***hotspots*** which are known as areas in the vicinity of wireless router coverage that are connected to the network and allow users to access internet services. With WiFi, it is able to provide wireless connection to the users’ devices which in this case the smartphone by emitting frequencies between 2.4GHz – 5GHz, based on the amount of data on the network. Wireless technology has widely spread lately and can get users connected almost anywhere; at home, at work, in libraries, schools, airports, hotels and even in restaurants.

Wireless networking is known as WiFi or 802.11 networking as it covers the IEEE 802.11 technologies. The major advantage of WiFi is that it is compatible with almost every operating system, game device, and advanced printer.

1.2 Problem Statement

Nowadays, the amount of smartphone users is a lot across the globe and this piece of technology has become a necessity in human's daily lives. Unfortunately, not many users are able to fully utilize their smartphone's potential to its peak performance given the fact that the smartphones bought comes with better hardware components built-in and have significantly better performance throughout the years of manufacturing.

In some scenario, first and foremost there are several types of users which some who do not like the hassle of bringing or carrying multiple devices anywhere such as mouse in this case and some who carries mouse for backup purposes if any issue occurs. For instance, people who use laptop may have a malfunction touchpad and need to bring their mouse anywhere they go just to do their works and activities on their laptop. This causes the user to carry more devices, takes up space in their backpack and increases weight to carry their workstation everywhere. There is also a situation where user may experience where their wireless mouse died due to dead battery in the middle of work causing the user to get a replacement battery to swap with the old one. On a similar case, user that uses wireless mouse also want to be free from cables which make wireless mouse so portable, travel friendly and requires no cable management.

To address this issue, this work proposed a mobile application that provides or enables the smartphone to act and function as a physical mouse by using the smartphone's screen as the touchpad to operate and move the mouse cursor across the monitor's screen.

1.3 Goals & Objectives

1. To develop a MobMouse application functioning as a touchpad using the smartphone's touch-screen display.

2. To evaluate the functionality of the proposed system.

1.4 Scope

For this project which is the MobMouse application, there are some requirements that are needed to complete this project. The hardware and client needed are listed below as follows:

1. An Android smartphone is required to run the MobMouse application.
2. Any users that own an Android smartphone can be the client for this project.
3. A wireless connection is required to establish connectivity between the smartphone and a PC.
4. The limitation of the wireless range between the smartphone and the PC is approximately 3m.
5. Compatible with Android version 4.0 (Ice Cream Sandwich) and above.
6. Both the smartphone and the PC must be connected to the same network to run the MobMouse application.
7. A UDP port must be opened on the PC to allow connection between the PC and the smartphone.

1.5 Thesis Outline

In this chapter, the Introduction subchapter explained regarding of the PSM Project. Next, the Problem Statement which took account in some scenarios or real life issues faced by users. Following that, the Goals & Objectives of this project were outlined as to what is the push factor in developing the mobile application based on the issues mentioned in the Problem Statement. Lastly, the Scope of this project which discussed on the project and user's boundaries.

The second chapter is the Literature Review that focuses on the existing systems and to list down all the comparisons between the MobMouse application with the existing systems found and to list down a few development softwares and make comparison to decide which software is better to develop the MobMouse application.

The third chapter is the Methodology where developer decides which methodology is best suit for the development of MobMouse application, listing down the requirements for both hardware and software necessary for MobMouse application’s development and to implement the necessary functions into MobMouse application.

The fourth chapter is the Testing and Result where testing plans are made for the testing phase that will be perform by both developer and users to test the functionality of the MobMouse application, to find any bugs or errors throughout the usage of the application and giving necessary and constructive feedbacks to the developer to further improve the MobMouse application.

The fifth and final chapter is the Conclusion which summarizes the project development as a whole and determines whether the MobMouse application aligns with the goals or objectives and requirements, to list down the limitations or constraints discovered during MobMouse application’s development and to suggest any future improvement which can benefit the MobMouse application in future versions.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter discusses on the details that are relevant and relatable to this project. Analyses are done in this chapter based on the hardware, technologies and tools that are suitable for this proposed project.

2.2 Cord/Wired Mouse

Mouse is a handheld hardware input device used to control cursor in a graphical user interface (GUI) which can move and select texts, icons, files and folders on a computer. For desktop computers, the mouse is placed on a flat surface such as mouse pad or on a desk in front of the computer, as shown in Figure 2.1, which mimics the surface on the computer screen where the cursor moves in conjunction with the way it is moved on a flat surface by the user. This motion is called two-dimensional motion relative to a surface. The motion is translated into the motion of a pointer on a display which allows a smooth control of the GUI. (Johnston, 2018)

The purpose of using a mouse with a computer is to increase the computer usability among all users because they do not need to memorize commands such as those utilized in a text-based command line environment like MS-DOS. For example, in MS-DOS, users must know the cd command and *dir* command and type those commands on the keyboard to navigate to a directory (folder) and view the files inside. However, Windows users are only required to double-click on the left button on a mouse to open a folder and see its contents. With the invention of mouse, the use of keyboard to navigate the files in a computer using command lines have been reduced in this age. (Johnston, 2018)

The type of connection used to connect between a mouse and a computer for the earliest version of mouse is by using a physical wire or cord to connect the mouse to the computer via the USB port which is a direct connection to the motherboard of a computer. Aside from the physical wired connection, another type of connection for the mouse to connect to the computer is via wireless which will be explained in the next subchapter. (Fisher, 2018)

Figure 2.1 Cord/Wired Mouse

2.3 Wireless Mouse

Aside from the physical cord/wired mouse, a new type of mouse was manufactured which uses wireless connectivity to connect between the mouse and the computer. This type of mouse sends signals to the computer without a cord. There are some of the wireless connections used by the wireless mouse to communicate with the computer such as Bluetooth, radio frequency or radio waves. Usually it comes with a USB receiver that is plugged into the computer, as shown in Figure 2.2, to receive signals from the wireless mouse. (Johnston, 2018)

Besides the connection type the wireless mouse uses, all of its functions provided are basically the same as the physical wired mouse counterpart. Another important aspect to mention is that wireless mouse needs batteries in order to work and typically uses AA batteries, AAA batteries, or rechargeable Li-ion batteries. Wireless mouse that is rechargeable needs a base station which is included by the manufacturer in order to charge the batteries inside the mouse. (Johnston, 2018)

Wireless mouse has since become standard in many different personal computing environments, adding the convenience of wire-free experience as well as the enhanced ergonomic opportunity of not being tethered with the main computing device and the hassle of cable management.

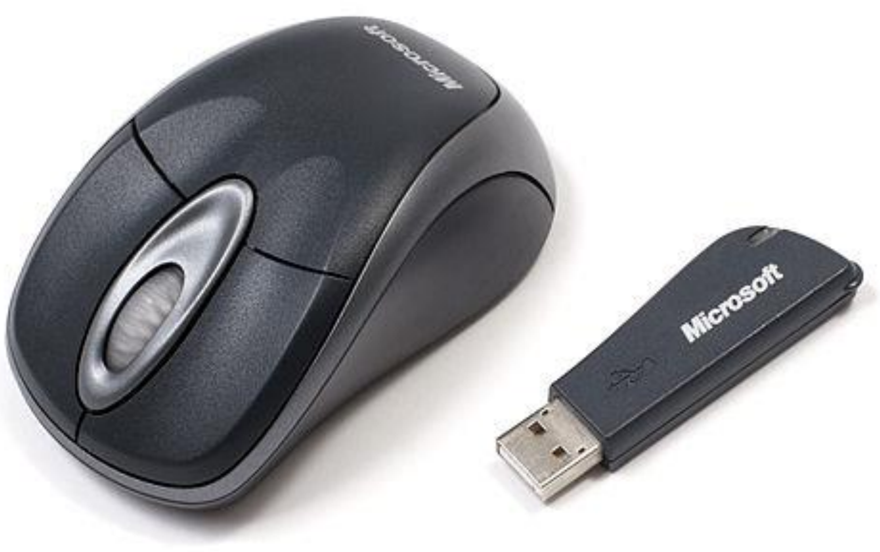


Figure 2.2 Wireless Mouse

2.4 Android Smartphone

Android smartphone is a smartphone that runs on Android OS, as shown in Figure 2.3, that was developed and owned by Google since 2005 and initially had its first release on September 2008 (Callaham, 2018). Android OS is used by a variety of mobile phone manufacturers such as Samsung, Sony, Asus, LG, Motorola, HTC and many more. Usually Android smartphones have a wide range of price in the smartphone markets starting from the low budget smartphones that starts at the price of RM200 to the high end premium smartphones which cost RM1000 and above. The price range of the Android smartphones determines the performance of the smartphone. Some of the Android devices include tablets, smartphones, smartwatches and many more. The Android major competitor in mobile operating system is the iOS. Both Android OS and iOS have been competing in the mobile devices market for a long time (Burrell, 2018).

Android has a growing selection of third-party applications, which can be acquired by users by downloading and installing the application's APK (Android application package) file, or by downloading them using an application store program that allows users to install, update, and remove applications from their devices. Google Play Store is the primary application store installed on Android devices that comply with Google's compatibility requirements and license the Google Mobile Services software. Google Play Store allows users to browse, download and update applications published by Google and third-party developers. There are more than one million applications available for Android in Play Store. Due to the open nature of Android, a number of third-party application marketplaces also exist for Android, either to provide a substitute for devices that are not allowed to ship with Google Play Store, provide applications that cannot be offered on Google Play Store due to policy violations, or for other reasons. Examples of these third-party stores have included the Amazon Appstore, GetJar, and SlideMe. F-Droid, another alternative marketplace, seeks to only provide applications that are distributed under free and open source licenses. The sheer range of Android applications available is a key selling point of Android mobiles. Best of all, many of these applications are totally free to download.

Android is designed to manage processes to keep power consumption at a minimum. When an application is not in use the system suspends its operation so that, while available for immediate use rather than closed, it does not use battery power or CPU resources. Android manages the applications stored in memory automatically. When memory is low, the system will begin invisibly and automatically closing inactive processes, starting with those that have been inactive for the longest amount of time.

Android's source code is released by Google under an open source license, and its open nature has encouraged a large community of developers and enthusiasts to use the open-source code as a foundation for community-driven projects, which deliver updates to older devices, add new features for advanced users or bring Android to devices originally shipped with other operating systems (Callaham, 2018).

Figure 2.3 Android Smartphones & Tablets

2.5 iOS Smartphone

iOS smartphone is a smartphone that uses Apple Operating System namely as iOS which stands for iPhone OS. iOS is one of the most popular mobile operating systems, developed and created by Apple Inc. Some of the Apple iOS devices include such as iPad, iPhone and many more as shown in Figure 2.4. The iOS major competitor in mobile operating system is the Android OS. Both iOS and Android OS have been competing in the mobile devices market for a long time (Johnson, 2017).

iOS is the main software in all Apple's mobile devices which allows users to interact with their iPhones or iPads. iOS also allows users to run any applications that the users download from the Apple's Application Store. While the applications are running, iOS does all the hard work, managing the user's iPhone or iPad's memory to ensure that their device runs smoothly (Barraclough, 2015).

The benefit of using iOS compared to Android is the security it offers to the users. The reason is that iOS is developed as a closed system while Android is developed as an open source system. By using closed system architecture, Apple has the authority to approve any mobile applications whether it is safe to install on the users' iPhone or iPad which makes the Apple's Application Store mostly free from malicious applications. This makes iOS less vulnerable to malicious threats compared to Android. Conversely, Android users can download and install applications from unknown sources, in the form of APK files found online. These haven't been approved by Google and therefore could contain viruses (Barraclough, 2015).

Figure 2.4 iPhones & iPads

2.6 Mobile Application

Mobile application in short is commonly known as mobile application and it is a software application that is developed and run for small mobile devices platform such as smartphones and tablets compared to their counterpart desktop PC and laptops. Most smartphones sold came with pre-installed mobile applications that are necessary and commonly used applications provided by their manufacturers. Aside from the preinstalled applications installed in the smartphone, users can acquire more mobile applications from the mobile application store such Google Play store, Apple Application store and Amazon Application store which have varieties of mobile applications for different purposes and functionality. This application turns the user's smartphone's screen display into a touchpad to navigate the cursor's movement on the computer and each dedicated touch-screen buttons for left-click and right-click mouse buttons.

The development of mobile applications are designed based on demands and constraints of the smartphones or tablets and taking full advantage of the features or capabilities that is available only in smartphones (Rouse, Mobile App Definition, 2013). During the early stage of smartphone's era in around 2010, the technologies were not powerful enough to support heavy and multitasking applications due to the limited hardware resources back then. However, that limitation is no longer an issue since the technologies of the hardware now have improved vastly as of late 2017. With the improved in smartphone's technologies, more mobile applications are being focus and developed in this platform.

The purposes of mobile applications development covers a wide range of fields such as entertainment, sports, fitness, utility, productivity, navigation and many more. Furthermore, the most popular mobile application category is the social media field such as Facebook application. Facebook application is widely used 2017 across all platforms (Viswanathan, 2017). Mobile applications usually have the functionality that is more suitable for small mobile device such as smartphone than desktop PC and laptop. For instance, a location-based mobile application which uses the GPS technology that is available in a smartphone based on the portability, convenience and form factor aspects.

2.7 Existing Systems Review

Based on some of the few existing systems found, there are some aspects and point of view to be considered and compare between the existing systems or applications that have been created which can function similarly and replace physical devices. Each existing systems are analyzed to identify the similarities and differences to compare against the MobMouse application.

2.7.1 Bluetooth Touchpad

In little insight on one of the existing systems which is the Bluetooth Touchpad. Bluetooth Touchpad is a mobile application, as shown in Figure 2.5, developed for android smartphones. This mobile application can turn the user’s smartphone into a touchpad for PC, laptop or netbook and supports both Windows and Linux.

To use this application, user must install its dedicated server with download link provided into the PC, laptop or netbook. After that, run the installed server and run the Bluetooth Touchpad application on the smartphone. Then, on the smartphone, connect to the server installed in the PC, laptop or netbook. The server needs Java to be installed on Windows devices while on Linux devices needs Java and BlueZ to be installed.

Some of the features offered by this application are that users are able to control the media player by pausing, adjusting volume and skip and rewind tracks. Moreover, users also can control the photos by switching to next or back to view the photos. In addition, additional features that is available for the Pro version of this application such as the dragging, scrolling, multi-touch zooming and multi-flicking. To unlock these features, users must purchase the Pro version which cost RM6.55 (Fedor, 2011).

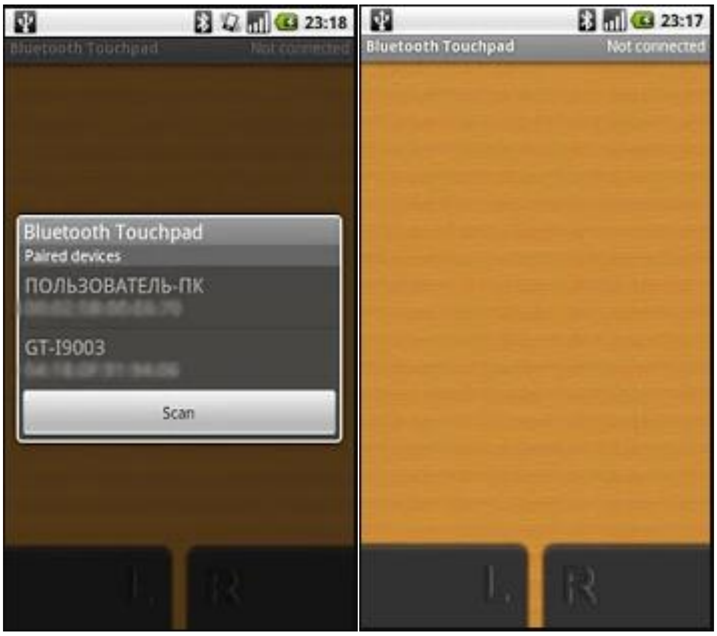


Figure 2.5 Bluetooth Touchpad (Fedor, 2011)

2.7.2 Universal TV Remote

A short explanation on the introduction of this mobile application called as Universal TV Remote which also developed for android smartphones. This mobile application has the function to act as similar as to a television remote, as shown in Figure 2.6, by taking advantage on the built-in IR transmitter inside the smartphone. This application is free to download from Google Play store and install on the smartphone.

How this application function is by running the Universal TV Remote application and selects the pairing remote of particular televisions to control the television. To achieve this, developers of this application take advantage on the IR transmitter or IR blaster inside the smartphone to connect with the television.

A quick description of IR blaster is that it lets the user use their phone as a universal remote to control television, air conditioner and many more. Example of scenario for this application such as if the user lost their television's remote, then this application can help solve the user's problem (Twinone, 2015).

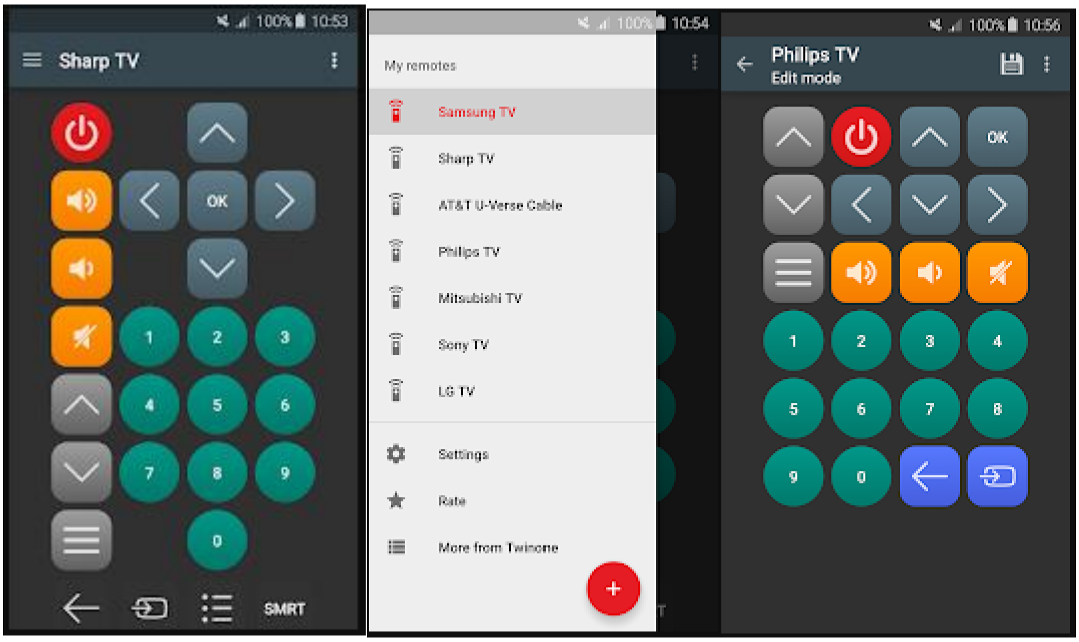


Figure 2.6 Universal TV Remote (Twinone, 2015)

2.7.3 Remote for Mac

For this particular mobile application which is known as the Remote for Mac is developed for android smartphones. This mobile application also has the function to control the Mac device's desktop remotely, as shown in Figure 2.7, by using the smartphone. The difference between this application and the Bluetooth Touchpad and MobMouse application is the Operating System (OS) installed on the PC whether it is using the Windows OS or Mac OS.

Remote for Mac application was developed to work and compatible with only Mac OSX devices. Other than that, this application requires the user's smartphone to have AppleScript support. Furthermore, this application requires the user's smartphone and the user's Apple's computer to connect in the same WiFi network to run the Remote for Mac application.

Some of the features provided by Remote for Mac application where users can experience such as controlling media playback on the Mac without needing to install additional programs on their computer. Other than that, this application give the users the ability to control the media players in an individually way such as iTunes, VLC, Iphoto, Spotify, Quicktime, MplayerX, PowerPoint, and Keynote. In addition, users can use this application to shutdown, suspend and adjust brightness and volume remotely.

Remote for Mac application is free to download from Google Play store and install on the smartphone but it offers an in-application purchase to upgrade the current application to a full version Remote for Mac application which provide additional features than the free version such as ads free experience, remote file browser and launcher, iTunes extended for selecting song or album to play and airfoil functionality (Oleg, 2017).

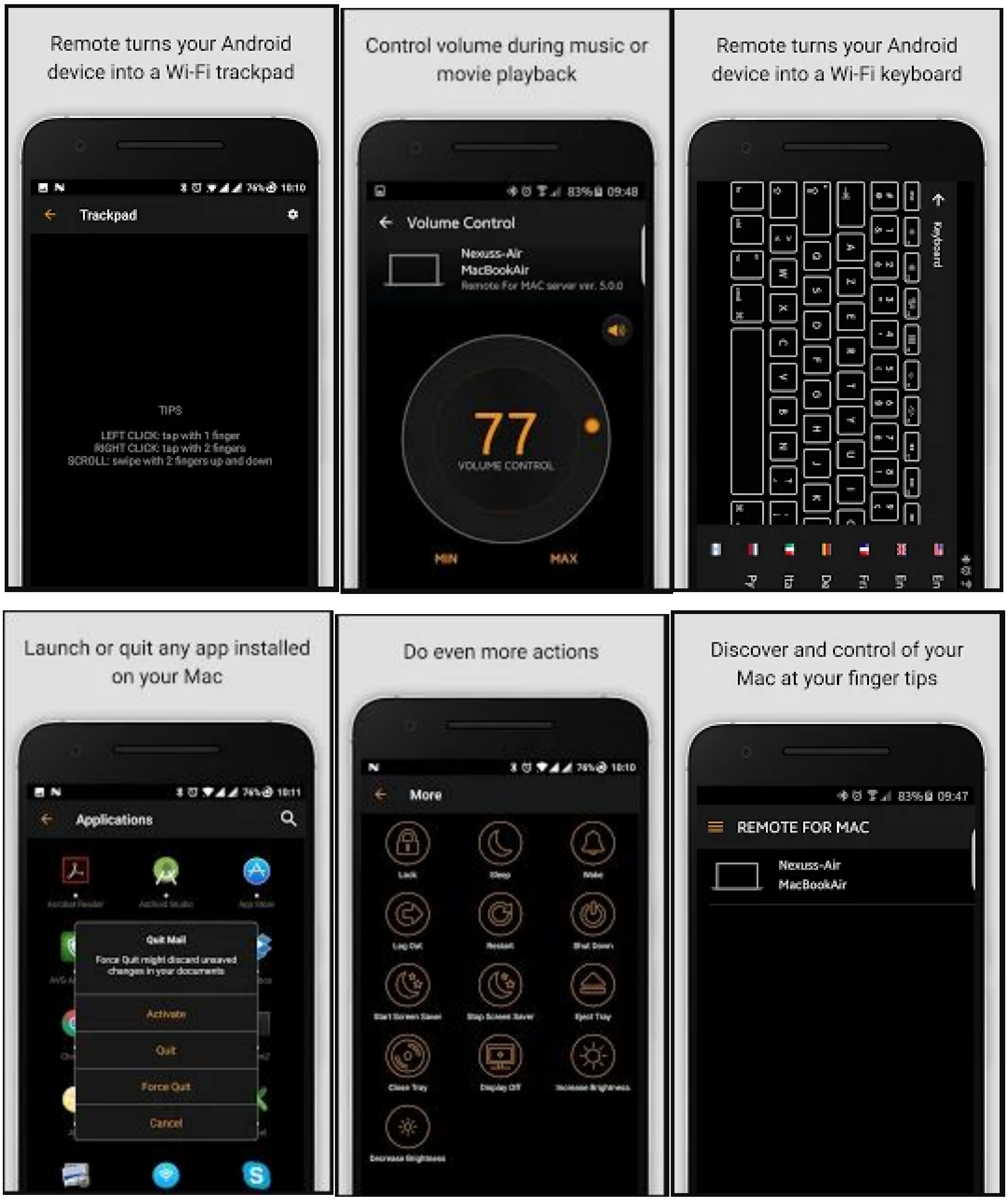

Remote turns your Android device into a Wi-Fi trackpad
Trackpad
TIPS
LEFT CLICK: tap with 1 finger
RIGHT CLICK: tap with 2 fingers
SCROLL: swipe with 2 fingers up and down
Control volume during music or movie playback
Volume Control
Nexuss-Air
MacBookAir
Remote For MAC server ver. 5.0.0
77
VOLUME CONTROL
MIN
MAX
Remote turns your Android device into a Wi-Fi keyboard
Keyboard
Launch or quit any app installed on your Mac
Applications
Quit Mail
Force Quit might discard unsaved changes in your documents
Activate
Quit
Force Quit
Cancel
Do even more actions
More
Lock
Sleep
Wake
Log Out
Restart
Shut Down
Start Screen Saver
Stop Screen Saver
Eject Tray
Close Tray
Display Off
Increase Brightness
Decrease Brightness
Discover and control of your Mac at your finger tips
REMOTE FOR MAC
Nexuss-Air
MacBookAir

Figure 2.7 Remote for Mac (Oleg, 2017)

2.8 Comparison between Existing Systems

This subtopic shows the comparison between three existing systems and the MobMouse application that may have similar functionality and features as the MobMouse application. The comparison between the existing systems and the MobMouse application are done based on some aspects, as shown in Table 2.1, that are essential to the system.

Table 2.1 Comparison between Existing System

Systems/ Aspects	MobMouse Application	Bluetooth Touchpad	Universal TV Remote	Remote for Mac
Features	▪ Control cursor using smartphone's screen. ▪ Scrolling using 1-finger gesture. ▪ Left and right clicks (buttons).	▪ Controls photo viewer & media player. ▪ Left and right clicks (buttons). ▪ Dragging, scrolling, multi-touch zooming and multi-flicking (Pro version).	▪ Control TV using smartphone.	▪ Control media players. ▪ Adjust brightness & volume. ▪ Able to shut-down and suspend. ▪ Ads free, remote file browser & launcher, iTunes Extended and airfoil functionality (Full version).
Connection type	WiFi	Bluetooth	IR	WiFi
Price	Free	Free (Offer in-	Free	Free (Offer in-

		application purchases)		application purchases)
Advertisement	No	No	Yes	Yes
Compatible OS	Android	Android	Android	Android
Target Device/OS	Windows	Windows & Linux	Television	Mac OSX devices
Limitations	▪ Wireless connection range is only up to 3m. ▪ Does not support Mac OS, Linux OS and TV.	▪ Does not support Mac OS. ▪ Connection via Bluetooth connectivity only. ▪ Limited features for free version.	▪ Does not support Windows OS, Mac OS and Linux OS. ▪ Does not support computers.	▪ Does not support Windows OS and Linux OS. ▪ Connection via WiFi connectivity only.

2.9 Development Software Options

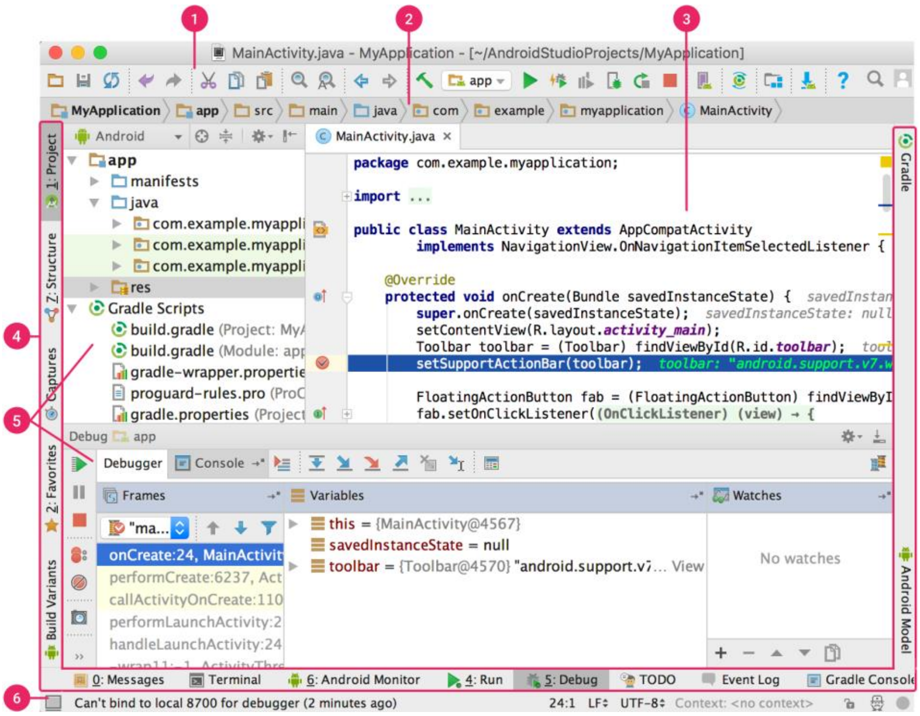
Nowadays, there are many development software tools existed which can be used by users or developers to create and program computer applications, websites, databases, mobile applications and many more. For this project which is the MobMouse application, the type of development is mobile application which supports on Android OS. Therefore, the development tools listed must be able to develop mobile application for Android OS.

2.9.1 Android Studio

Android studio is an official IDE, as shown in Figure 2.8, which was developed by Google and JetBrains. This IDE is used for developing native Android's applications. Android Studio was developed to replace Eclipse Android Development Tools as the primary IDE for native Android application development. Android Studio's first public version was released on December 2014. The programming languages supported by Android Studio are Java, Kotlin and C++. Android Studio is available on Windows, Mac OS and Linux based operating systems (Rouse, Android Studio, 2018).

With Android Studio users can develop applications such as:

- Games
- Databases
- Sensors
- Connectivity



```
package com.example.myapplication;

import ...

public class MainActivity extends AppCompatActivity
        implements NavigationView.OnNavigationItemSelectedListener {

    @Override
    protected void onCreate(Bundle savedInstanceState) {   savedInstan
        super.onCreate(savedInstanceState);   savedInstanceState: null
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);   tool
        setSupportActionBar(toolbar);   toolbar: "android.support.v7.w

        FloatingActionButton fab = (FloatingActionButton) findViewByI
        fab.setOnClickListener((OnClickListener) (view) → {
```

```
onCreate:24, MainActivit
performCreate:6237, Act
callActivityOnCreate:110
performLaunchActivity:2
handleLaunchActivity:24
```

```
this = {MainActivity@4567}
savedInstanceState = null
toolbar = {Toolbar@4570} "android.support.v7... View
```

```
Can't bind to local 8700 for debugger (2 minutes ago)        24:1  LF÷  UTF-8÷  Context: <no context>
```

Figure 2.8 Android Studio Interface

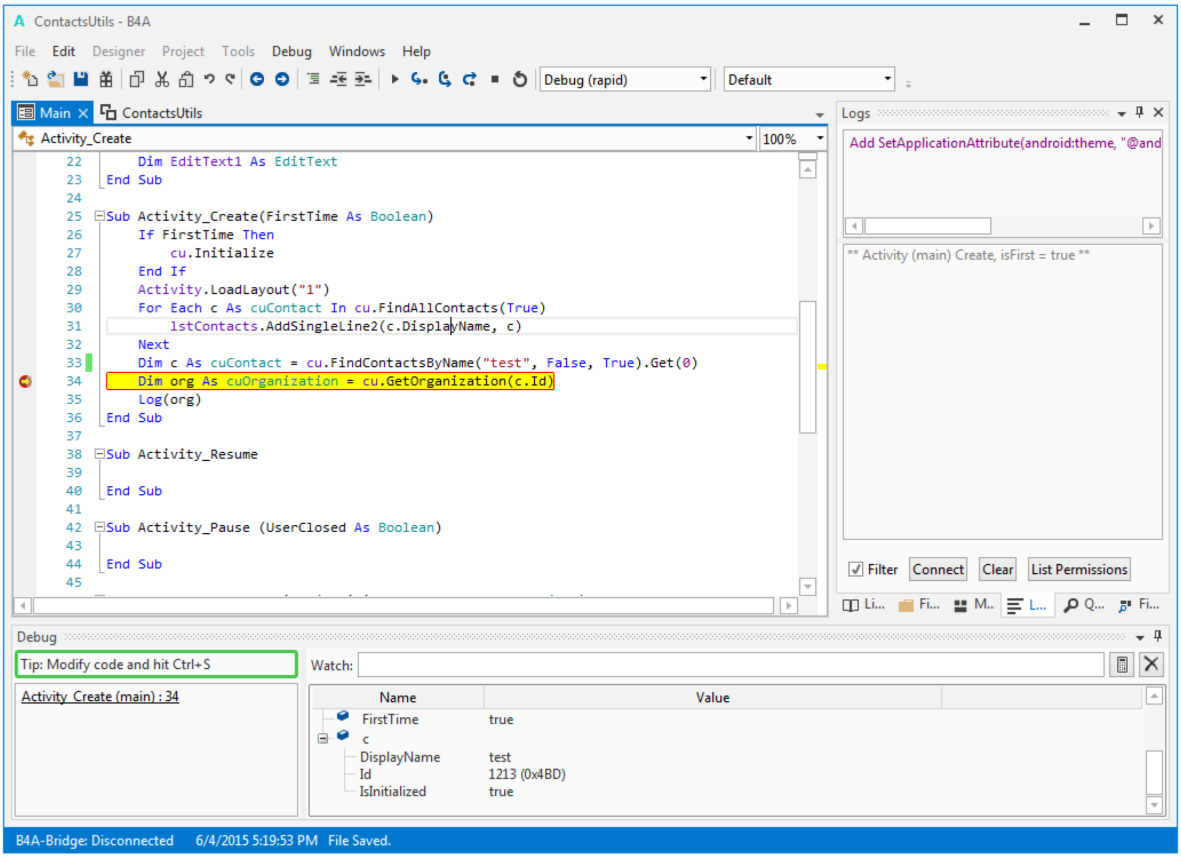
2.9.2 B4A

B4A is formerly known as Basic4Android, is the second IDE, as shown in Figure 2.9 and Figure 2.10, produced by Anywhere Software. Its first public version was released on 7 December 2010 which allows users to code native Android applications in B4X. B4A is able to create applications, widgets, and games. There is also a beginner guides that is updated after each new release of the IDE. B4A is categorized as one of the B4X development tools. B4X is a suite of rapid application development IDEs that allows the creation of applications on platforms such as Google Android, Apple iOS, Java, Raspberry Pi and Arduino. B4X uses a proprietary dialect of Visual Basic.

B4A is an alternative to programming with Java. It includes a visual designer that simplifies the process of building user interfaces that target smartphones and tablets with different screen sizes. The language itself is similar to Visual Basic and Visual Basic .Net though it is adapted to the native Android environment. B4A is an object-based and event-driven language. B4A generates standard signed Android applications which can be uploaded to application stores like Google Play, Samsung Applications and Amazon Appstore. There are no special dependencies or runtime frameworks required. B4A is available on Windows OS. B4A interacts with the native API through Java libraries. B4A libraries consist of two files: the Java jar file and a XML file that is produced by a tool provided with B4A (Seagrave, 2015).

B4A includes all the features needed to quickly develop any type of Android application. B4A is used by tens of thousands of developers from all over the world, including companies such as NASA, HP, IBM and others. Together with B4i you can now easily develop applications for both Android and iOS. With B4A users can create applications such as:

- Games
- Databases
- Sensors
- Hardware
- Connectivity



```
    Dim EditText1 As EditText
End Sub

Sub Activity_Create(FirstTime As Boolean)
    If FirstTime Then
        cu.Initialize
    End If
    Activity.LoadLayout("1")
    For Each c As cuContact In cu.FindAllContacts(True)
        lstContacts.AddSingleLine2(c.DisplayName, c)
    Next
    Dim c As cuContact = cu.FindContactsByName("test", False, True).Get(0)
    Dim org As cuOrganization = cu.GetOrganization(c.Id)
    Log(org)
End Sub

Sub Activity_Resume

End Sub

Sub Activity_Pause (UserClosed As Boolean)

End Sub
```

```
Add SetApplicationAttribute(android:theme, "@and
** Activity (main) Create, isFirst = true **
```

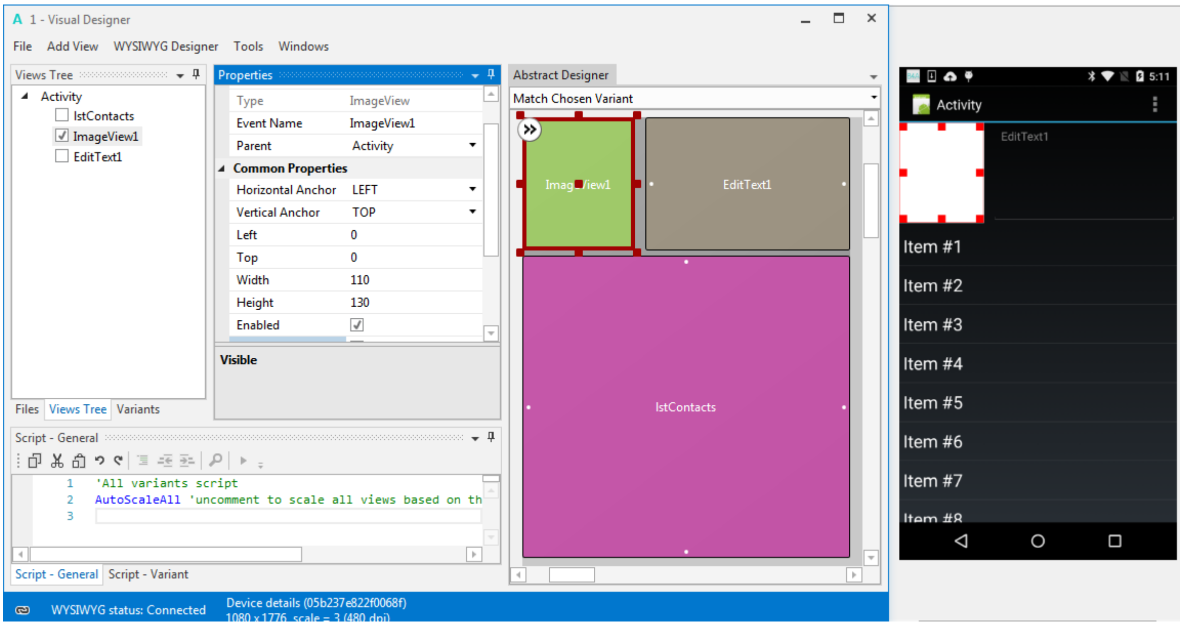
Tip: Modify code and hit Ctrl+S

Activity_Create (main) : 34

Name	Value
FirstTime	true
c	
DisplayName	test
Id	1213 (0x4BD)
IsInitialized	true

B4A-Bridge: Disconnected 6/4/2015 5:19:53 PM File Saved.

Figure 2.9 B4A Interface



Type	ImageView
Event Name	ImageView1
Parent	Activity
Common Properties	
Horizontal Anchor	LEFT
Vertical Anchor	TOP
Left	0
Top	0
Width	110
Height	130
Enabled	☑

Visible

```
'All variants script
AutoScaleAll 'uncomment to scale all views based on th
```

WYSIWYG status: Connected Device details (05b237e822f0068f) 1080 x 1776, scale = 3 (480 dpi)

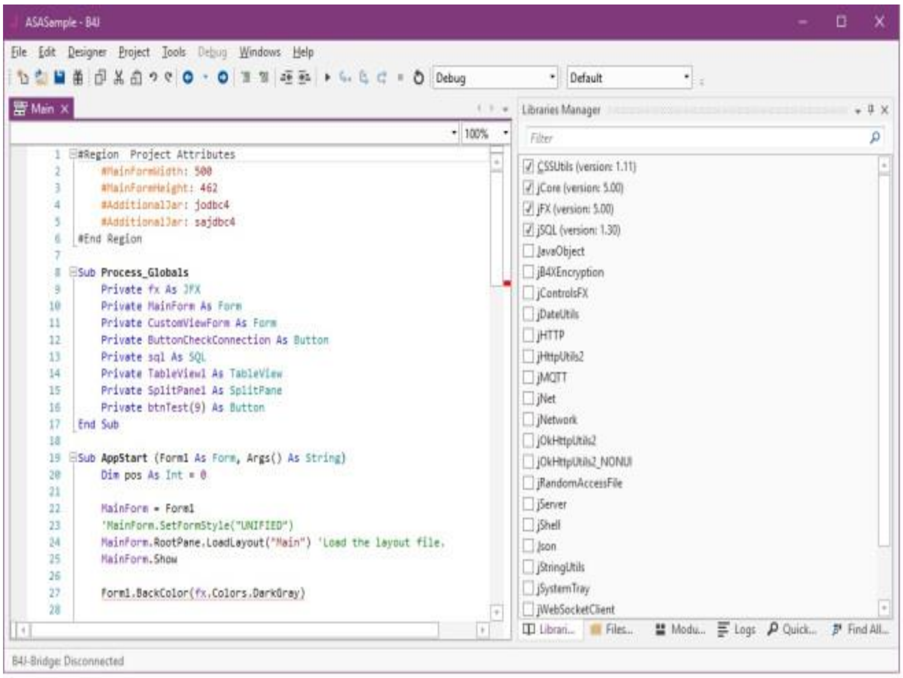
Figure 2.10 B4A Visual Designer Interface

2.9.3 B4J

B4J is the third IDE, as shown in Figure 2.11 and Figure 2.12, also produced by the same developer as B4A which is Anywhere Software. Its first public version was released on 4 December 2013. B4J targets the following platforms such as desktops, web servers, and ARM boards such as Raspberry Pi. B4J is also categorized as one of B4X IDE. It’s free and can be used to develop desktop applications, server and IoT solutions. B4J offers an IDE with a full set of features that include a visual designer, debugger, compiler, hundreds of libraries, and a packager that creates self-contained installers with no dependencies which simplifies the development. It generates JAR files, which means that applications developed with B4J, can run in windows, linux and Mac OS. It’s very powerful. B4J is available on Windows OS (MacNN, 2015).

With B4J users can create applications for these platforms:

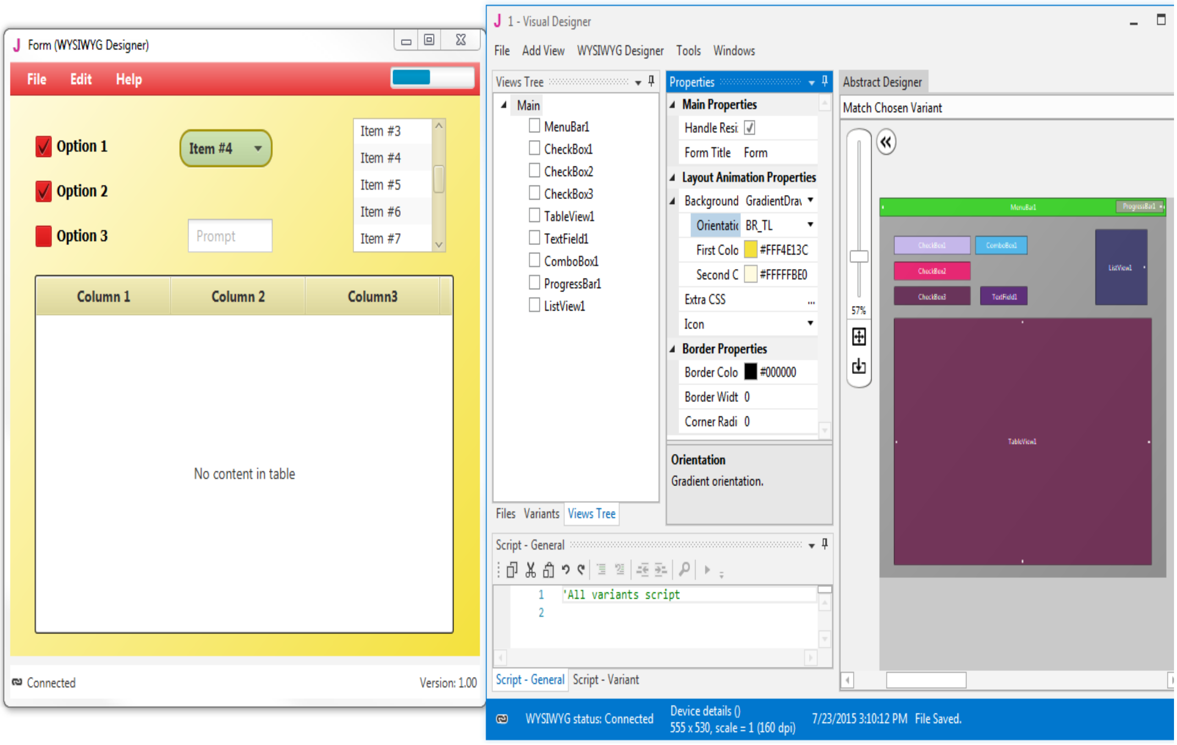
- desktop applications (UI)
- console programs (non-UI)
- web server solutions



```
#Region  Project Attributes
    #MainFormWidth: 500
    #MainFormHeight: 462
    #AdditionalJar: jodbc4
    #AdditionalJar: sajdbc4
#End Region

Sub Process_Globals
    Private fx As JFX
    Private MainForm As Form
    Private CustomViewForm As Form
    Private ButtonCheckConnection As Button
    Private sql As SQL
    Private TableView1 As TableView
    Private SplitPane1 As SplitPane
    Private btnTest(9) As Button
End Sub

Sub AppStart (Form1 As Form, Args() As String)
    Dim pos As Int = 0

    MainForm = Form1
    'MainForm.SetFormStyle("UNIFIED")
    MainForm.RootPane.LoadLayout("Main") 'Load the layout file.
    MainForm.Show

    Form1.BackColor(fx.Colors.DarkGray)

```

☑ CSSUtils (version: 1.11)
☑ jCore (version: 5.00)
☑ jFX (version: 5.00)
☑ jSQL (version: 1.30)
☐ JavaObject
☐ jB4XEncryption
☐ jControlsFX
☐ jDateUtils
☐ jHTTP
☐ jHttpUtils2
☐ jMQTT
☐ jNet
☐ jNetwork
☐ jOkHttpUtils2
☐ jOkHttpUtils2_NONUI
☐ jRandomAccessFile
☐ jServer
☐ jShell
☐ Json
☐ jStringUtils
☐ jSystemTray
☐ jWebSocketClient

Figure 2.11 B4J Interface



Handle Resi ☑
Form Title Form
Background GradientDraw
Orientatic BR_TL
First Colo #FFF4E13C
Second C #FFFFFBE0
Extra CSS
Icon
Border Colo #000000
Border Widt 0
Corner Radi 0

Orientation
Gradient orientation.

```
'All variants script
```

Figure 2.12 B4J Visual Designer Interface

2.9.4 Software Chosen

Based on the options listed in the above subtopics, only selected software will be used in the development of MobMouse application throughout this PSM2. Decision is made based on the comparisons between the development software tools mentioned.

The software development tools used are B4A and B4J because the programming language used is B4X language which reduces the complexity of the systax in coding compared to Java language. In addition, due to the use of B4X language, a wide skill set of users ranging from beginner and advanced users are able to use this IDE much quicker than Android Studio. Other than that, the interfaces for both B4A and B4J are more user-friendly and less daunting. Furthermore, both B4A and B4J are RAD tools that belong under B4X. In addition, both B4A and B4J are uses less RAM compared to Android Studio that prone to high RAM usage which causes slow and sluggish computer to operate.

2.10 Conclusion

From this study, a decision was made regarding which software development tools to use to proceed the development of MobMouse application after some consideration. The software development tools used are B4A and B4J because the programming language used is B4X language which reduces the complexity of the systax in coding compared to Java language. In addition, due to the use of B4X language, a wide skill set of users ranging from beginner and advanced users are able to use this IDE much quicker than Android Studio. Other than that, the interfaces for both B4A and B4J are more user-friendly and less daunting. Furthermore, both B4A and B4J are RAD tools that belong under B4X. In addition, both B4A and B4J are uses less RAM compared to Android Studio that prone to high RAM usage which causes slow and sluggish computer to operate.

Based on the comparison done, a short description for some of the existing systems which cover the features, target devices or OS, price, connection types and many more. The Bluetooth Touchpad application is available on Google Play store for free and with Pro version which users need to purchase to use the full features provided. Other than that, this application is compatible with android smartphones and can run on both Windows and Linux OS. It uses a Bluetooth connection type by installing a dedicated server into the computer and connects with the user’s smartphone.

Next, the Universal TV Remote application is available on Google Play store for free and comes with in-application purchases to unlock the full features. This application also compatible with android smartphones that is equipped with built-in IR Blaster and can only run on TV, cable box and many more. This application uses IR transmitter signal for connection type to pair both the TV and smartphone.

After that, the Remote for Mac application is also available on Google Play store for free and with Full version with additional features unlocked if purchases in the application. In addition, this application is compatible with android smartphones and able to run on Windows OS only by using the WiFi connection type to connect the smartphone with the Mac OSX devices.

On the other hand, the MobMouse application will be available for free and it is compatible with android smartphones and will be able to run on Windows OS only by using the WiFi connection type to connect both the smartphone with the computer. This application only supports on Android version 4.0 (Ice Cream Sandwich) and above.

For the limitations, Bluetooth Touchpad must use Bluetooth connectivity, does not support Mac OS and the full features can only be used after the users have purchase the Pro version. Universal TV Remote only uses IR signal to connect to the TV which only smartphones that has built-in IR transmitter can use the application. This application only supports TV, cable box and so on. Remote for Mac uses WiFi connectivity only and full features must be purchase to unlock it and does not support Windows and Linux. Lastly, MobMouse application has around 3m of wireless connection range and does not support Mac OS, Linux and TV.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, we will be discussing on the development of MobMouse application based on the requirements analyzed which is to develop a MobMouse application functioning as a touchpad using the smartphone's touch-screen display. In order to do that, the application must be installed in the smartphone and the PC must be installed with the MobMouseHandler.bat program. Next, both the smartphone and PC must be connected in a same network. Once that step is done, user is able to operate their smartphone as a mouse by performing simple hand movement gesture such as using the smartphone's touch-screen to move the cursor on the PC, pressing the smartphone's touch-screen for left and right mouse buttons and using one finger gesture on the smartphone's touch-screen for scrolling up and down.

3.2 Project Methodology

Given the nature of this project, the methodology used is the Rapid Application Development (RAD) Methodology to develop the MobMouse application. Rapid Application Development (RAD) Methodology is known as an adaptive software development approach. This method is to allow rapid development, as shown in Figure 3.1, which put less emphasis on planning and more emphasis on an adaptive process unlike the traditional method which is the Waterfall Methodology. Waterfall methodology must finish one stage of development before able to continue the next phase of development. Due to this nature, this method consumed a lot of time in order to make any changes or issues discovered during the software development. RAD is especially well suited for (although not limited to) developing software that is driven by user interface requirements (Anderson, 2017).

Rapid Application Development (RAD) Methodology was created in order to overcome the inefficiencies present in the Waterfall Methodology. This methodology focused on speed and used strategies such as prototyping, iterative development and time boxing. This method can also be applied to hardware development as well. It includes requirements gathering tools, prototyping tools, computer-aided software engineering tools, language development environments such as Java platform. This method also implements object-oriented programming methodology which enables software reuse and some of the popular programming languages used are C++ and Java that are offered in visual programming packages to provide rapid application development (Rouse, rapid application development (RAD), 2016).

There are some scenarios where Rapid Application Development (RAD) Methodology can be suitable for MobMouse application project such as:

- Time constraint of the project.
- Goals are interchangeable over time.
- Main requirements were identified.

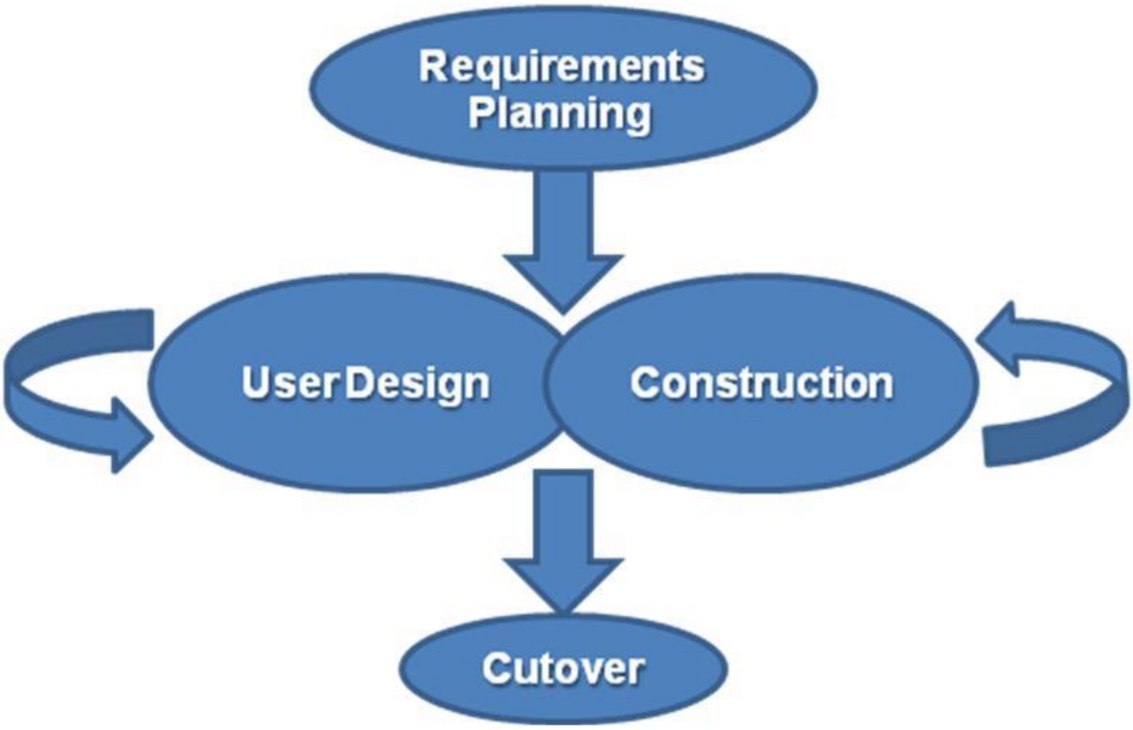


Figure 3.1 Rapid Application Development (RAD) Methodology

The development of the software begins by specifying the user requirements following with the designing and construction phase where each of the phases will be reviewed in order to identify and gather further requirements if necessary. This process is repeated to produce a newer and low risk version of the software for each cycle of the Rapid Application Development (RAD) Methodology.

3.2.1 Advantages of Rapid Application Development (RAD) Methodology

There are several advantages when applying Rapid Application Development (RAD) Methodology for this project's development (Powell-Morse, 2016), such as:

- Reduced development time due to prototyping.
- Provides greater flexibility as redesign is done according to the developer.
- Possibility of lesser defects due to prototyping in nature.
- Able to get reliable users' feedback.
- Results in reduction of manual coding due to code generators and code reuse.
- This project can be started with minimum budget.

3.3 Requirements

In this subtopic, we will discuss on the requirements needed for this project to be able to start the development. The smartphone and PC are needed to be operable. To achieve that, there are a few softwares that are necessary for this project, which are:

- B4A
- B4J
- Android SDK
- JDK
- JRE

Moreover, the development of the application requires some programming language that is suitable based on the project requirement gathered and analyzed such as:

- B4X

B4X programming language is chosen to develop this application because it is derived from BASIC programming language that can be learned quickly, its statements are easy to read by other programmers. B4X language is used in many applications such as games, databases, connectivity, sensors and hardware on platforms such as Google Android, Apple iOS, Java, Raspberry Pi, Arduino, desktops and web servers.

B4X is an object-oriented BASIC which supports the object-oriented and event-driven programming paradigm. Furthermore, allowing developers to adopt and reuse code, able to run the code in secure manner on many platforms, regardless of the Operating System (Sinicky, 2016).

As for the Android SDK, it is required because it is a set of development tools used to develop applications solely for Android platform. To develop an Android application, the programming language used mostly is Java (Sinicki, 2017). It also includes:

- Required libraries
- Debugger
- An Android emulator
- Relevant documentation for the Android APIs
- Sample source code
- Tutorials for Android OS

Next, the JDK is needed because it is used for developing Java applications including Android application and includes Java Runtime Environment (JRE). JRE is an interpreter (java), a compiler (javac), an archiver (jar), a documentation generator (javadoc) and other tools needed in Java development which also responsible for translating the code into an application which makes the multi-platform possible (Tyson, 2018).

3.3.1 Hardware Requirements

The hardware requirements for MobMouse application project are described or stated as in Table 3.1 below.

Table 3.1 Hardware Requirements

Hardware	**Description**
Smartphone with Android OS installed	• To install the MobMouse applications into the smartphone. • To run the MobMouse applications. • To control the mouse or cursor's inputs by smartphone.
PC with WiFi card built-in	• PC is needed for this project because the applications must pair with a PC in order to fully function. • To provide the IP address of the PC to be inserted into the applications for pairing. • To install the applications' server into the PC.
Wireless Router	• To provide connection between the smartphone and PC.

3.3.2 Software Requirements

The software requirements for MobMouse application project are described or stated as in Table 3.2 below.

Table 3.2 Software Requirements

Software	Description
B4A	• To provide comprehensive facilities. • To develop the application.
B4J	• To provide comprehensive facilities. • To develop the application.

3.4 Context Diagram

Below is the context diagram designed for the MobMouse application:

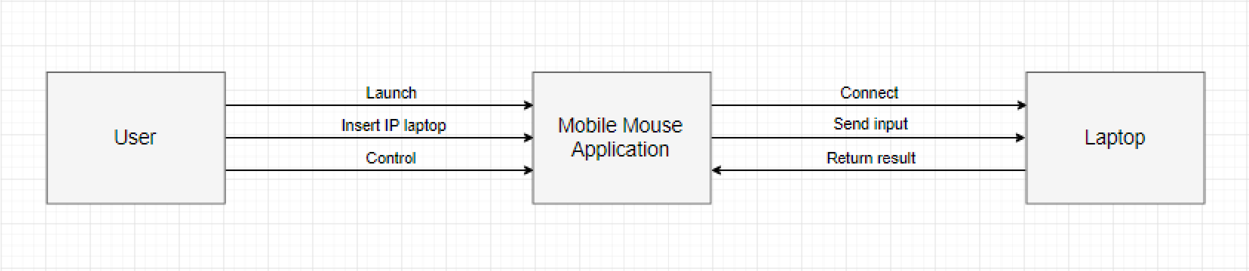


Figure 3.2 Context Diagram

Figure 3.2 shows the context diagram for this project. The user will launch the applications and insert their PC's IP address into the smartphone and it will connect both the smartphone and the PC. User then can control the movement of the mouse's cursor via the smartphone and the applications will send the input to the PC and register the input. Then, it returns the result of the input displayed on the PC's screen.

3.5 Use Case Diagram

In this subtopic, a use case diagram for the MobMouse application is shown below in Figure 3.3.

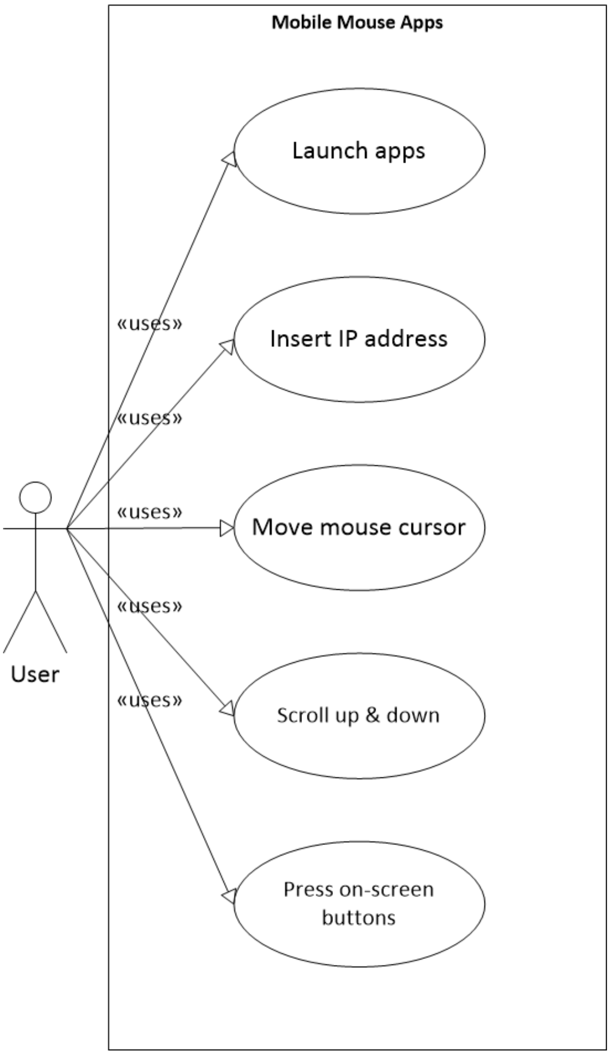
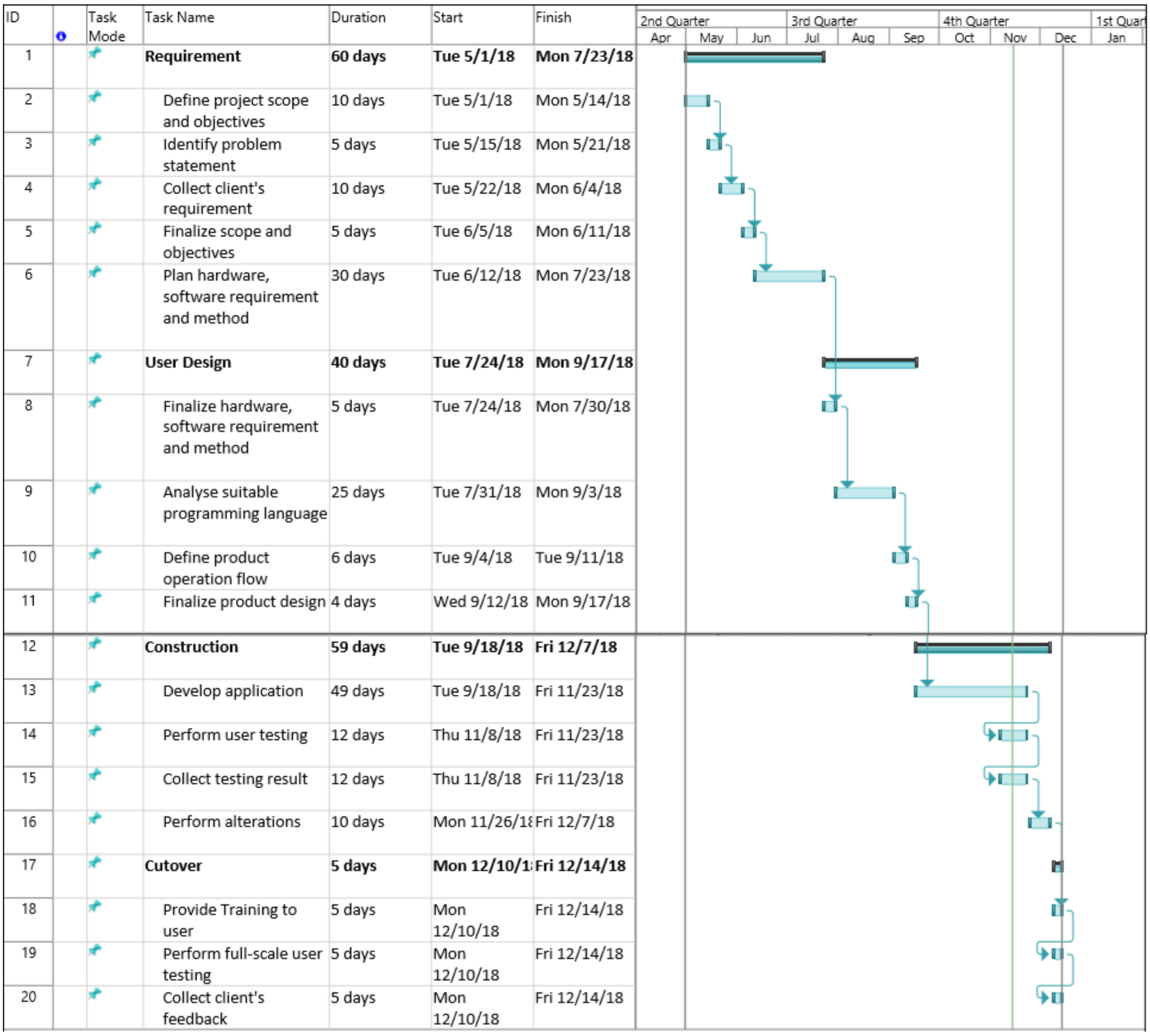


Figure 3.3 Use Case Diagram

As shown in Figure 3.3, the user will start the interaction with the MobMouse applications by launching the applications in his or her smartphone that is connected to the same network with the PC. Then, the user needs to insert the IP address of the PC into the application to connect to the PC. User is able to move the mouse cursor using the smartphone’s screen, press left and right buttons on the smartphone’s screen and scroll the browser using one finger.

3.6 Gantt Chart

In this subtopic, we will show the progress of the project via Gantt chart. Figure 3.4 displays the activities against the date throughout the project progress.



ID		Task Mode	Task Name	Duration	Start	Finish
1			**Requirement**	**60 days**	**Tue 5/1/18**	**Mon 7/23/18**
2			Define project scope and objectives	10 days	Tue 5/1/18	Mon 5/14/18
3			Identify problem statement	5 days	Tue 5/15/18	Mon 5/21/18
4			Collect client's requirement	10 days	Tue 5/22/18	Mon 6/4/18
5			Finalize scope and objectives	5 days	Tue 6/5/18	Mon 6/11/18
6			Plan hardware, software requirement and method	30 days	Tue 6/12/18	Mon 7/23/18
7			**User Design**	**40 days**	**Tue 7/24/18**	**Mon 9/17/18**
8			Finalize hardware, software requirement and method	5 days	Tue 7/24/18	Mon 7/30/18
9			Analyse suitable programming language	25 days	Tue 7/31/18	Mon 9/3/18
10			Define product operation flow	6 days	Tue 9/4/18	Tue 9/11/18
11			Finalize product design	4 days	Wed 9/12/18	Mon 9/17/18
12			**Construction**	**59 days**	**Tue 9/18/18**	**Fri 12/7/18**
13			Develop application	49 days	Tue 9/18/18	Fri 11/23/18
14			Perform user testing	12 days	Thu 11/8/18	Fri 11/23/18
15			Collect testing result	12 days	Thu 11/8/18	Fri 11/23/18
16			Perform alterations	10 days	Mon 11/26/18	Fri 12/7/18
17			**Cutover**	**5 days**	**Mon 12/10/1**	**Fri 12/14/18**
18			Provide Training to user	5 days	Mon 12/10/18	Fri 12/14/18
19			Perform full-scale user testing	5 days	Mon 12/10/18	Fri 12/14/18
20			Collect client's feedback	5 days	Mon 12/10/18	Fri 12/14/18

Figure 3.4 Gantt Chart

3.7 Software Installation

In this subtopic, detailed step by step instructions will be included in order to install the development tools which are the B4A and B4J to begin the development of this project.

3.7.1 B4A Installation

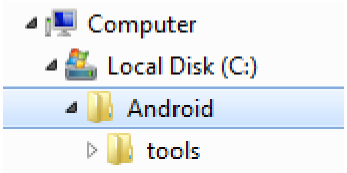
Below are the installation steps to install B4A:

1. Java JDK v8

- Open the Java 8 JDK download link.
- Check the Accept License Agreement button.
- Select "Windows x86 (32-bit) or Windows x64" in the platforms list.
- Download the file and install it.

2. Android SDK

- Download Android SDK command line tools.
- Unzip it in a folder such as C:\Android as shown in Figure 3.5.

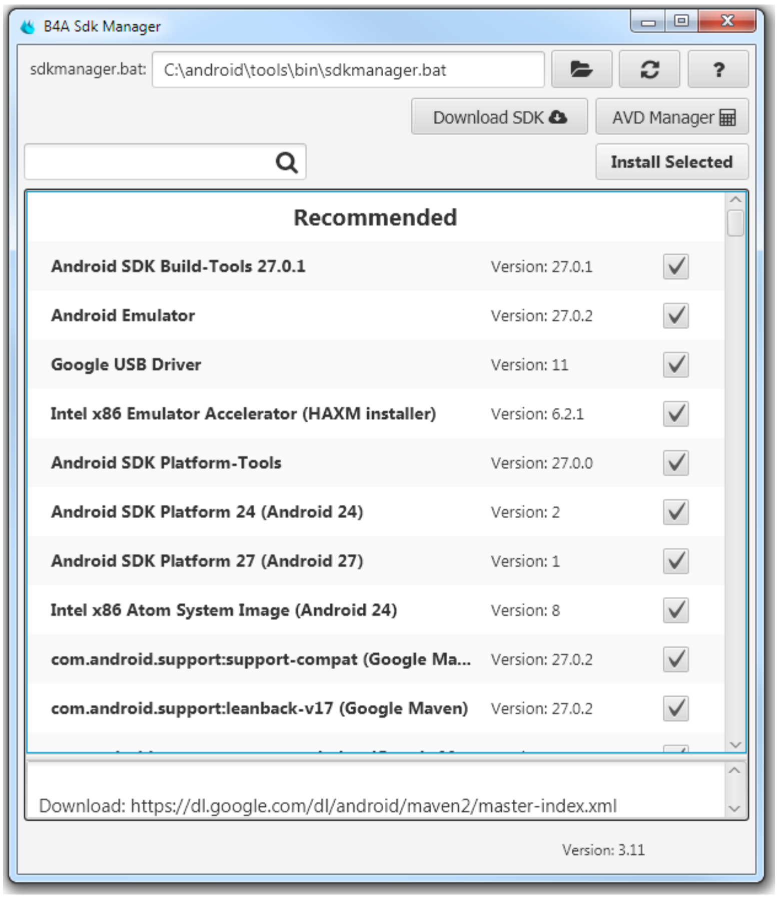


Computer
Local Disk (C:)
Android
tools

Figure 3.5 Set Android SDK location folder

3. B4A

- Download B4A Trial Version
- Open B4A.
- Choose Tools > Configure Paths.
- Set the path to "javac.exe" if it is empty (C:\Program Files\Java\jdk1.8.0_102\bin\javac.exe).
- Click on Open Sdk Manager button as shown in Figure 3.6.
- Set the path to "sdkmanager.bat" (C:\Android\tools\bin\sdkmanager.bat) and install all recommended items.
- Read and accept the licenses when asked for.
- Return to B4A and set the path to "android.jar" (C:\Android\platforms\android-28\android.jar). Make sure to set the path to android-28.



sdkmanager.bat: C:\android\tools\bin\sdkmanager.bat

Recommended

Android SDK Build-Tools 27.0.1	Version: 27.0.1	✓
Android Emulator	Version: 27.0.2	✓
Google USB Driver	Version: 11	✓
Intel x86 Emulator Accelerator (HAXM installer)	Version: 6.2.1	✓
Android SDK Platform-Tools	Version: 27.0.0	✓
Android SDK Platform 24 (Android 24)	Version: 2	✓
Android SDK Platform 27 (Android 27)	Version: 1	✓
Intel x86 Atom System Image (Android 24)	Version: 8	✓
com.android.support:support-compat (Google Ma...	Version: 27.0.2	✓
com.android.support:leanback-v17 (Google Maven)	Version: 27.0.2	✓

Download: https://dl.google.com/dl/android/maven2/master-index.xml

Version: 3.11

Figure 3.6 B4A SDK Manager

4. B4A-Bridge (optional)

Figure 3.7 B4A Bridge on Google Play Store

- B4A-Bridge allows the IDE to connect to the device over the wireless network. This is an alternative to USB debug mode (which is also supported).
- Download B4A-Bridge from Google Play as shown in Figure 3.7. You can search for B4A Bridge or press on the above button. You can also download the apk file directly.
- Make sure that installation of applications from unknown sources is enabled. This can be set under Settings - Security or Settings - Applications.
- Run B4A-Bridge on the device. See this tutorial for more information.
- Connect the IDE to the device: Tools - B4A Bridge - Connect
- You are ready to run your first Android program!

3.7.2 B4J Installation

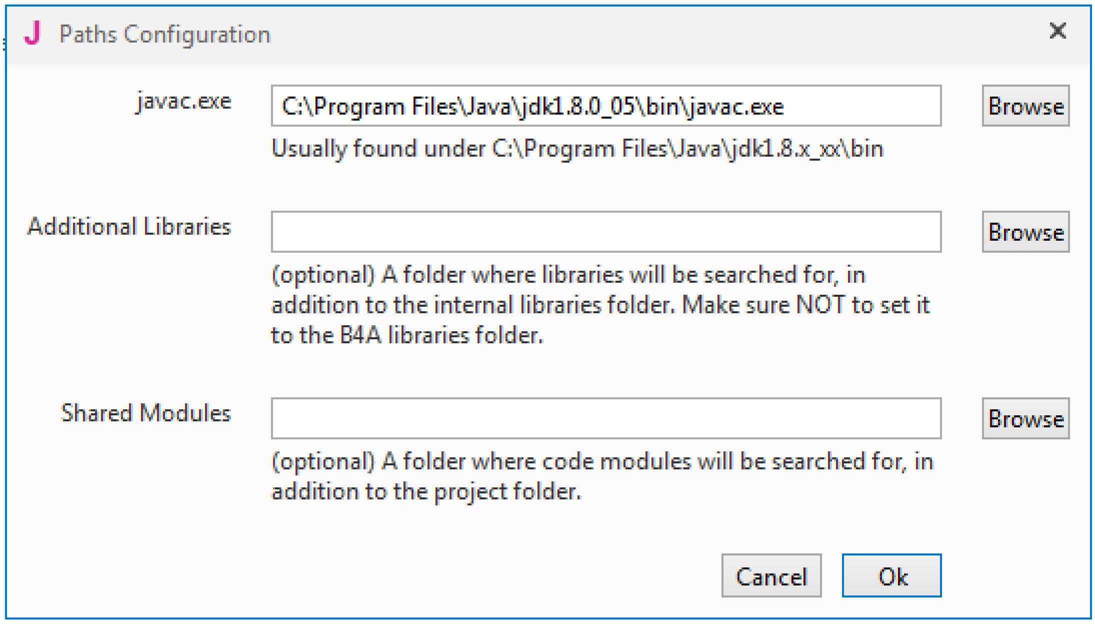
Below are the installation steps to install B4J:

1. Java JDK v8 (8u40+)

- Open the Java 8 JDK download link.
- Check the Accept License Agreement button.
- Select "Windows x86" or "Windows x64" (for 64 bit machines) in the platforms list.
- Download the file and install it.

2. Install and configure B4J

- Download B4J Full Version
- Open B4J.
- Choose Tools > Configure Paths as shown in Figure 3.8.



Paths Configuration

javac.exe: C:\Program Files\Java\jdk1.8.0_05\bin\javac.exe

Usually found under C:\Program Files\Java\jdk1.8.x_xx\bin

Additional Libraries:

(optional) A folder where libraries will be searched for, in addition to the internal libraries folder. Make sure NOT to set it to the B4A libraries folder.

Shared Modules:

(optional) A folder where code modules will be searched for, in addition to the project folder.

Figure 3.8 Configure javac folder paths

- Configure Paths
- Use the Browse button to locate "javac.exe".
- javac is located under <java folder>\bin.

3.8 Implementations

In this subtopic, more details will be covered regarding on the implementation inside MobMouse application. The overview of this system starts with the user as shown in Figure 3.9. The user must run the MobMouse application that is installed in their smartphone. Then, the user need to enter their PC’s IP address in the MobMouse application to connect to the PC. Once connected, user will be able to control the mouse cursor, scroll vertically and pressing left and right mouse button via the application installed in their smartphone.

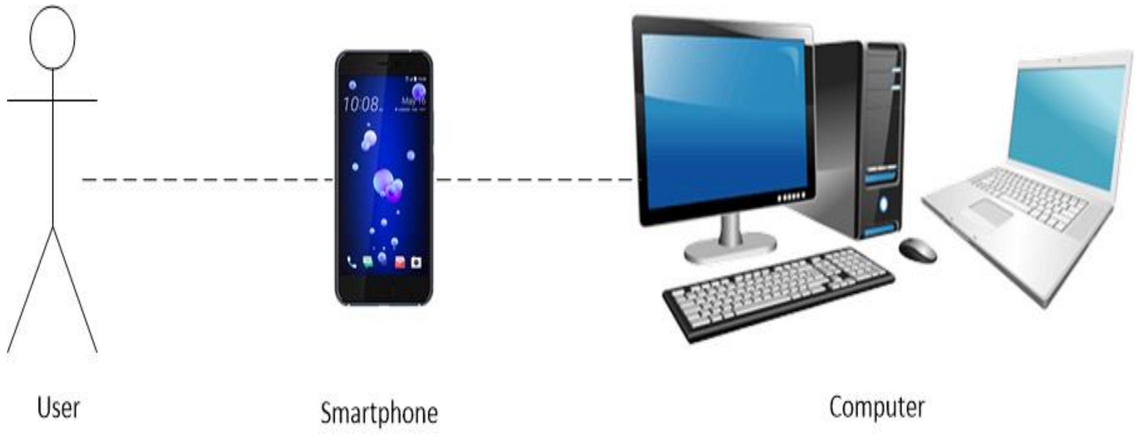


Figure 3.9 Overall User Interaction

Since the old days, when user wants to use a computer it should involve the usage of the computer keyboard and the mouse. These two things are used by the user to control all over the computer. Therefore, by using these two hardware, user can perform the computer process respectively.

As an example, when user moves around the mouse, the cursor on the screen will move respectively. Besides that, when user double click the left click button on specific folder or file, the computer will open the file. However, if the user just single click the left button of the mouse, it simply means that the user is selecting the file or folder. If the user wants to close an application, the user should click the close button on the upper right of the window or application. Apart from that, if the user performs single right click on the mouse, the computer simply will show the drop down menu, which the user can choose to copy, paste, create new file or folder and others. To show a simple form the set of basic actions that user is able to perform are in Table 3.3 below:

Table 3.3 Basic Mouse Actions

User Actions	**Computer Actions/Outputs**
Move the mouse.	Move the cursor across the screen.
Scroll vertically.	Scroll the screen up and down.
Double click left mouse button.	Open file, folder or application.
Single click left mouse button.	Select file, folder or application.
Single right click mouse button.	Show the drop down menu.

Since this project propose the use of mobile application via smartphone to control the mouse, these actions are needed to implement in the mobile application depending on the smartphone functionality. Below are several models proposed in order to perform different actions as stated in Table 3.4 below:

Table 3.4 Proposed Actions

User Actions	**Computer Actions/Outputs**
Use a finger to touch & move across the smartphone’s screen.	Move the cursor across the screen.
Use a finger to touch & move across the smartphone’s screen vertically.	Scroll the screen up or down.
Double press left button on the smartphone’s screen.	Open file, folder or application.
Single press left button on the smartphone’s screen.	Select file, folder or application.
Single press right button on the smartphone’s screen.	Show the drop down menu.

3.8.1 Flowcharts

Some flowcharts will be showed later in this subtopic to visualize the overall algorithm of the application.

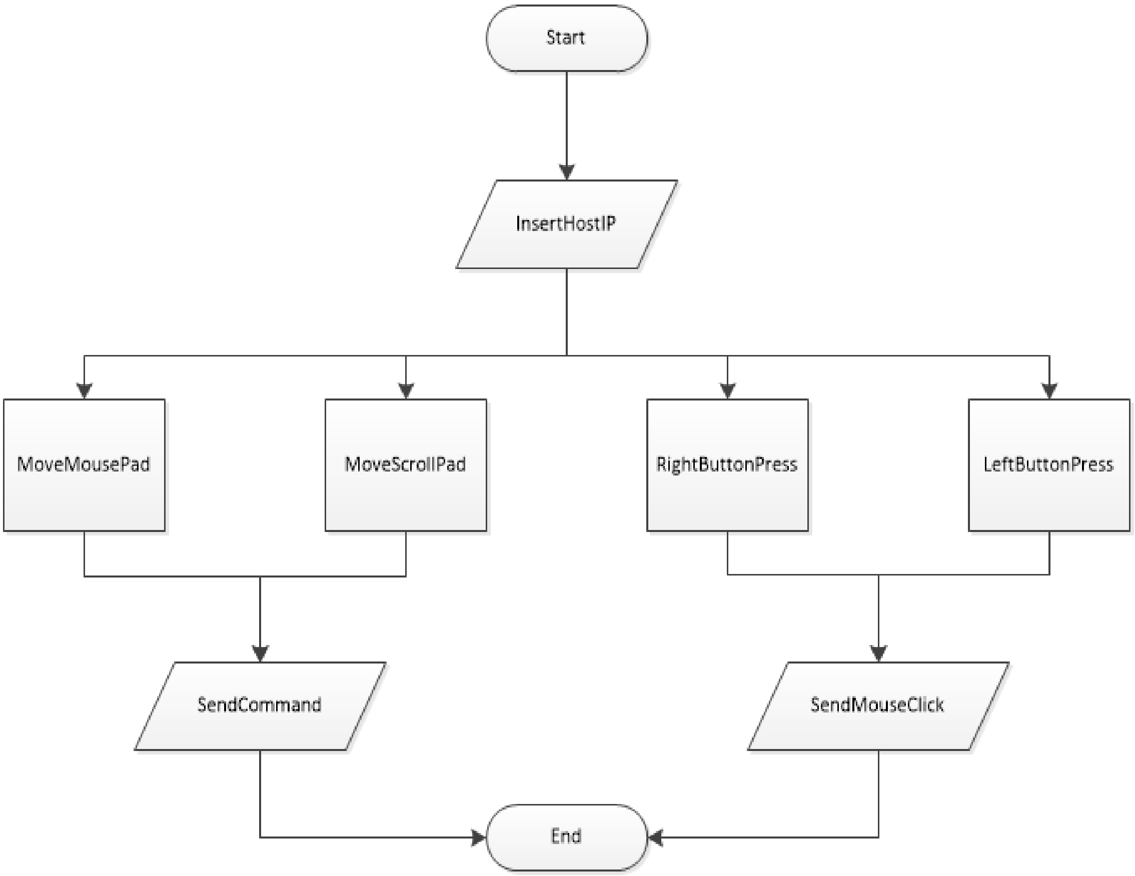


Figure 3.10 Overall Algorithm

Figure 3.10 above shows the overall algorithm of this application that will be used by the user to control the mouse. The system starts with the user run and insert their PC's IP address in the application. After that, the application will create a robot to take the command perform by the user and send to the computer for output. Next, user is able to perform actions such as mouse buttons click, scroll vertically and moving the cursor on their smartphone via the application which then will be send to the computer for output.

3.8.2 Move Cursor Algorithm

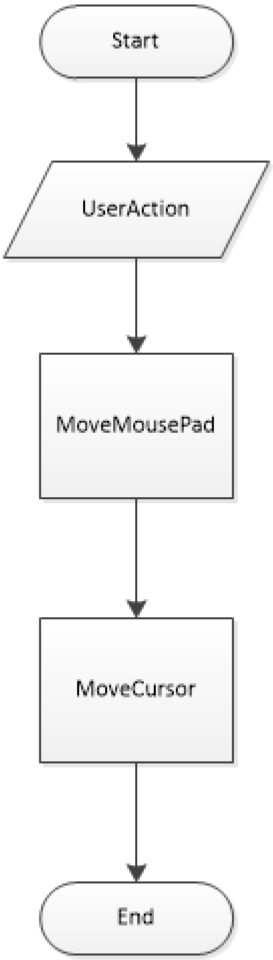


Figure 3.11 Move Cursor Algorithm

Figure 3.11 above shows the algorithm of moving the mouse's cursor. This algorithm is used when the user touches and moves across their smartphone's screen with a finger. The application will register that particular action and move the cursor on the monitor according to the finger movement respectively.

3.8.3 Scroll Algorithm

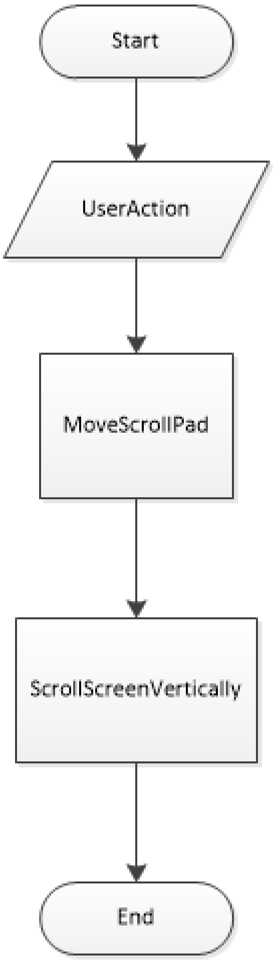


Figure 3.12 Scroll Algorithm

Figure 3.12 above shows the algorithm of scrolling the monitor screen up and down. This algorithm is used when the user touches and moves up or down their smartphone's screen with a finger. The application will register that particular action and scroll the screen up or down according to the finger movement respectively.

3.8.4 Double Press Left Button Algorithm

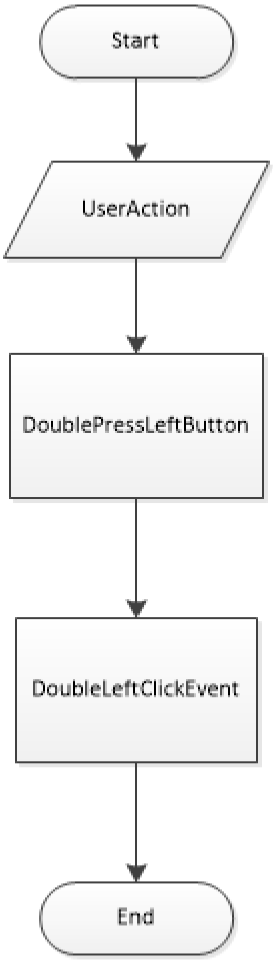


Figure 3.13 Double Press Left Button Algorithm

Figure 3.13 above shows the algorithm of pressing the smartphone’s on-screen left button. This algorithm is used when the user press twice the smartphone’s on-screen left button. The application will register that action and will open the folder, file or application on the computer.

3.8.5 Single Press Left Button Algorithm

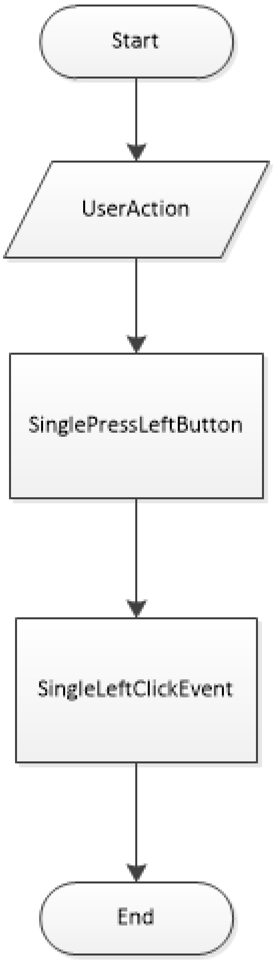


Figure 3.14 Single Press Left Button Algorithm

Figure 3.14 above shows the algorithm of pressing the smartphone’s on-screen left button. This algorithm is used when the user press the smartphone’s on-screen left button once. The application will register that action and will select the folder, file or application on the computer.

3.8.6 Single Press Right Button Algorithm

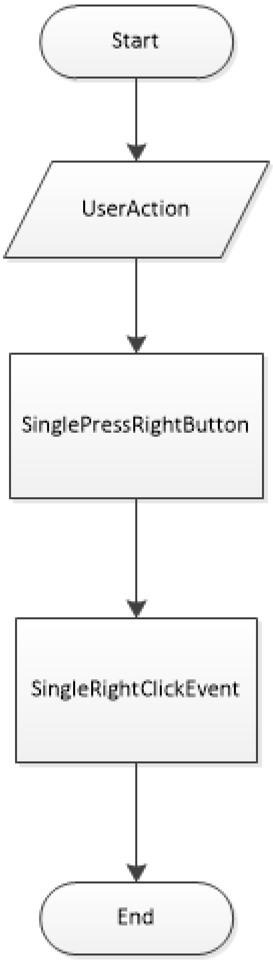


Figure 3.15 Single Press Right Button Algorithm

Figure 3.15 above shows the algorithm of pressing the smartphone’s on-screen right button. This algorithm is used when the user press the smartphone’s on-screen right button once. The application will register that action and will open the drop down menu.

CHAPTER 4

TESTING & RESULT

4.1 Introduction

This chapter documents the testing process and the results of the testing. The goal is to evaluate the functionality of the proposed system by performing the testing phase to see if the outcomes from the MobMouse application meet the user requirements, to identify any bug or error within the application and to perform necessary actions to fix those errors and bugs before releasing and delivering to the user for use.

4.2 Developed Product

For this subtopic, the components of the MobMouse application will be explained with figures to help distinguishing between other components.

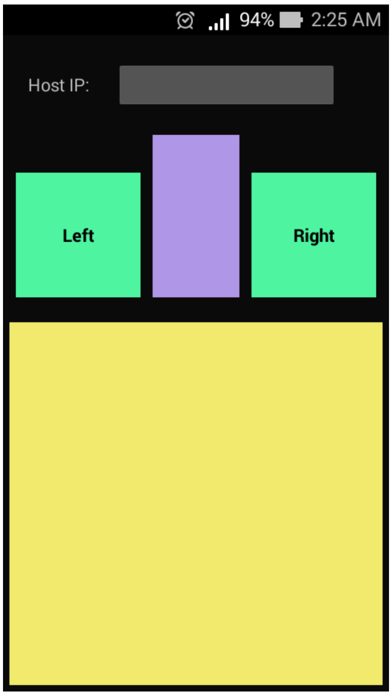


Figure 4.1 MobMouse UI

Based on the Figure 4.1 above, it is the MobMouse application’s interface when it is launched by a user. It consists of two buttons, a scroll pad, a touchpad and a textbox for inserting IP address of the user’s PC to pair with the smartphone. The big yellow square at the bottom half of the smartphone’s screen is the touchpad for user to control the movement of the mouse’s cursor on the computer screen. The purple rectangle located between the left and right buttons is the scroll pad. This is where the user can perform scrolling action to scroll the computer screen from top to bottom.

Next is the greenish square which has the word “Left” on it which indicates the left button that is similar to a physical mouse that performs both the double left click events and single left click event. User only needs to press on that Left button on the smartphone’s screen to invoke the double click events or single click event depending on the number of presses by the user. For double click events, the outcomes or computer actions are such as opening folder, file or application. For single click event, the outcome or computer action will be selecting folder, file or application.

Besides that, another greenish square which has the word “Right” indicates the right button that is similar to a mouse that can perform the single right click event. User can press on that Right button on the smartphone’s screen to invoke the single right click event to perform the respective computer action which is to display a drop down menu which have multiple options.

Lastly, the input field for a host IP address as mentioned before is where the user insert the IP address of their PC so that the smartphone can pair with the PC that has the inputted IP address. Overall, the UI design for this application is fairly simple and straightforward as to ease the usability of this application by the user by reducing the learning curve with a fair amount since most people are familiar with the functions provided by the physical mouse.

4.3 Testing Phase

This subtopic will be covering more regarding the testing that had been done towards the MobMouse application. Testing phase is very important in any software development. It determines how acceptable and satisfying the developed system so far. Apart from that, testing phase is where errors or bugs can be found inside the system. Therefore, necessary actions can be made to fix the defective parts of the system before delivering it to the end users.

For this project, the MobMouse application must be able to function properly as intended by the user ensuring that each actions performed by the user is registered by the system and successfully executes the computer actions on the computer. This is because the objective of this project is to develop a mobile application that allows user to use their smartphone to act as a computer mouse. Therefore, every actions implemented inside the application will be tested to make sure that from each action, the application is able to perform the computer action respectively. Below is the user's workflow on how to use the MobMouse application and the testing plan of what should be tested by the user and observe the results whether it is within expectation or not based on the user requirements

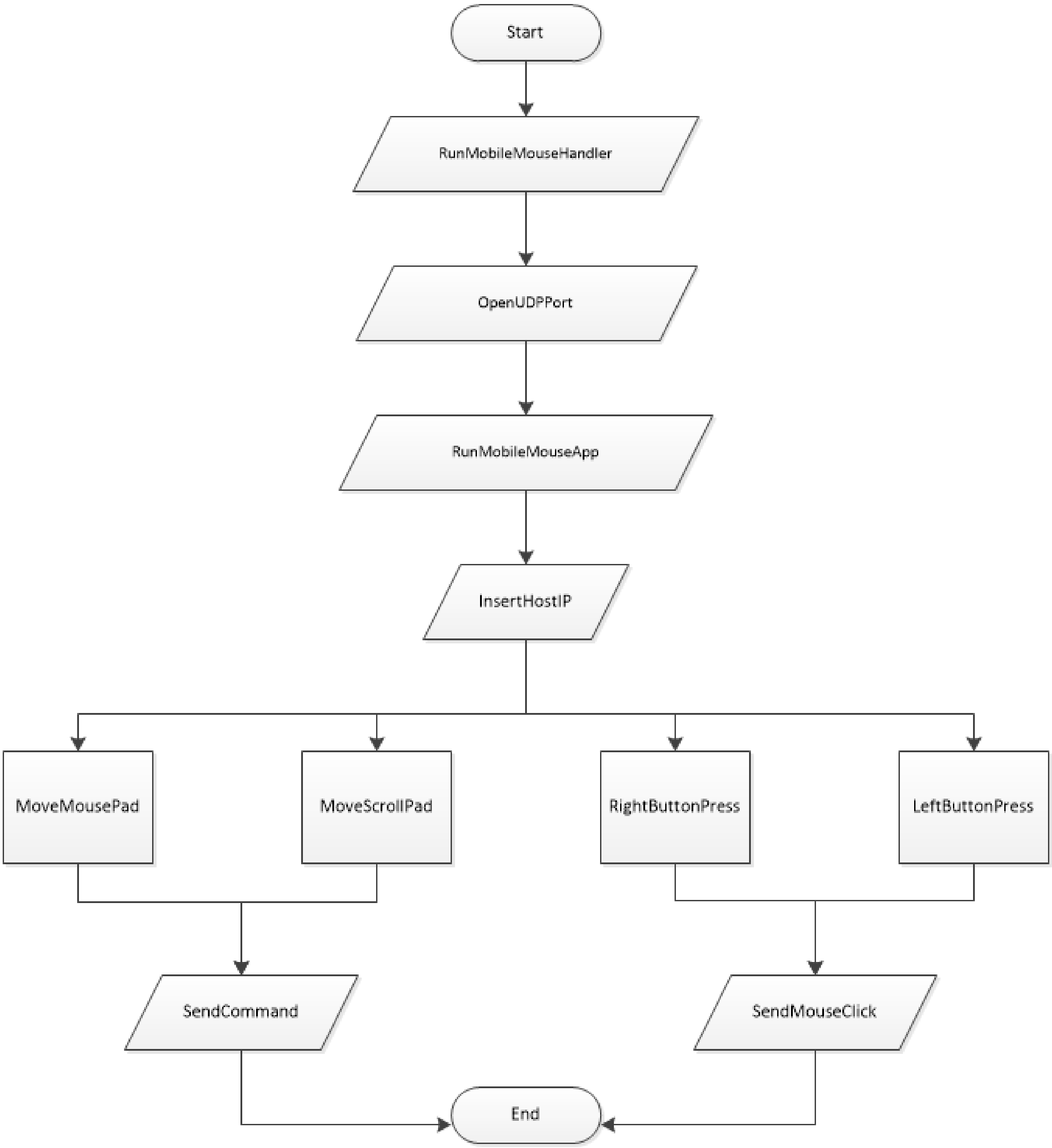


Figure 4.2 User Workflow

Figure 4.2 above shows the workflow on how to use the MobMouse application assuming that both the user’s PC and smartphone are connected in the same network as this application will only work when both devices are in the same network. Firstly, user needs to run the MobMouseHandler.bat application on their PC which will display a notification window on the computer screen about what is the IP address of the PC and the specific port number that needs to be opened so that the smartphone can connect to the user’s PC through the opened port number. User must open a UDP port on their PC based on the port number displayed by the MobMouseHandler.bat application.

The testing done will be based on Table 4.1 below. Once opened, user can run the MobMouse application on their smartphone. Then, user must insert their PC’s IP address displayed by the MobMouseHandler.bat application into the application to connect. Once connected, user can then perform various actions that are similar to a physical mouse such as moving the mouse’s cursor on the computer screen, scrolling the computer screen up and down, pressing once or twice on the left button and pressing once on the right button. When user presses the buttons, scroll or move the cursor, the application will send the input to the computer.

Table 4.1 Testing Plan

User Actions	Testing
Use a finger to touch & move across the smartphone’s screen.	Must ensure that the application is able to perform mouse movement across the computer screen when user touches and moves a finger across the smartphone’s screen.
Use a finger to touch & move across the smartphone’s screen vertically.	Must ensure that the application is able to perform mouse movement across the computer screen when user touches and moves their finger vertically on the smartphone’s screen.
Single press left button on the smartphone’s screen.	Must ensure that the application is able to execute single click left mouse button events to perform the computer action respectively when user single presses the left button on the smartphone’s screen.
Double press left button on the smartphone’s screen.	Must ensure that the application is able to execute double click left mouse button events to perform the computer action respectively when user double presses the left button on the smartphone’s screen.
Single press right button on the smartphone’s screen.	Must ensure that the application is able to execute double click right mouse button events to perform the computer action respectively when user single presses the right button on the smartphone’s screen.

4.4 Results/Outcomes

Based from the testings that were done above, the results will be recorded in this subtopic. The test was done by the developer based on the testing plan provided to gather more information on the usability of the application, the functionality of the application and the effectiveness of the application as well as to detect any error or bug that may present during the testing phase with the aim that appropriate actions can be taken to fix it before releasing to users for use.

The first test is to move the mouse’s cursor on the computer screen by touching and moving a finger on the smartphone’s screen. The result is shown in the figure below both before and after.

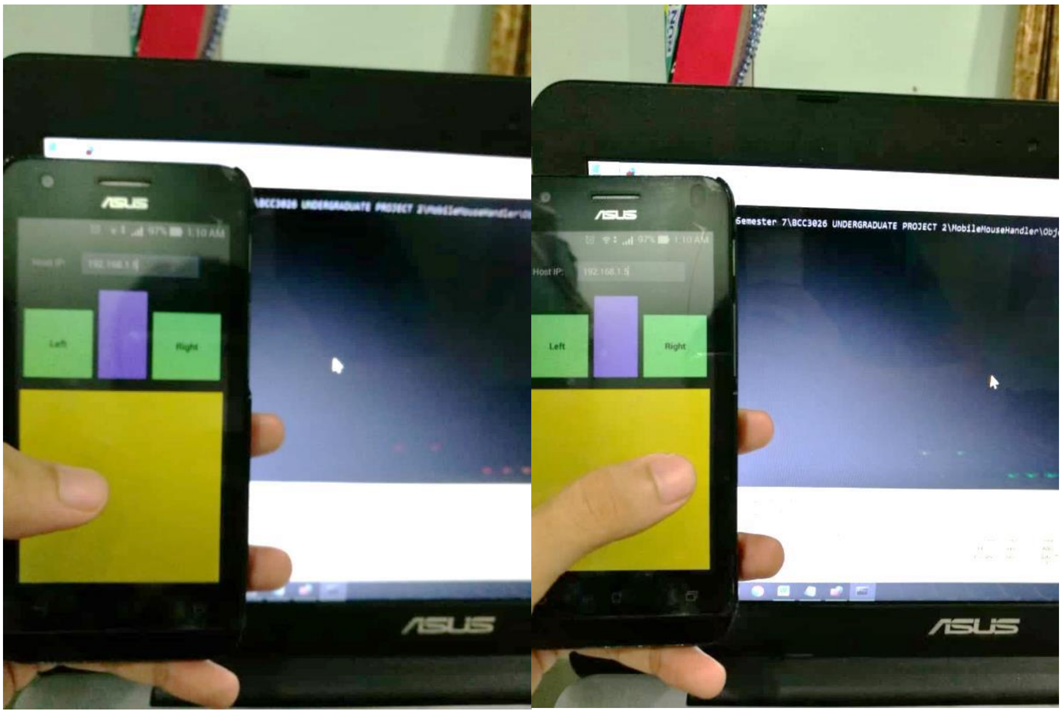

ASUS
Host IP:
Left
Right
Semester 7\BCC3026 UNDERGRADUATE PROJECT 2\MobileMouseHandler\Obj
ASUS

Figure 4.3 Moving cursor to the right (before & after)

Based on Figure 4.3 above, the touchpad is able to perform its function as expected when it was tested to move the cursor from centre to the right direction successfully.

The second test is to scroll the computer screen up and down direction by touching and moving a finger on the smartphone's screen vertically. The result is shown in the figure below for both downward and upward directions.

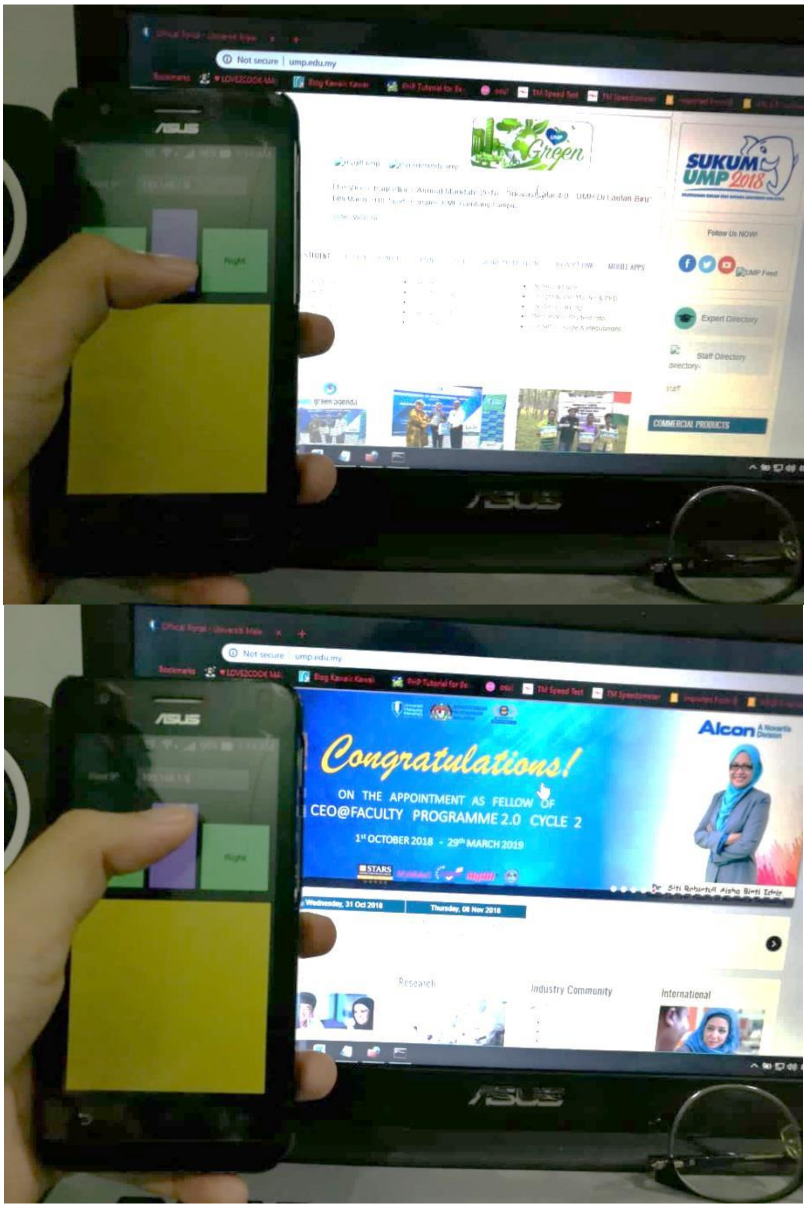


Figure 4.4 Scrolling down and up

Based on Figure 4.4 above, the scroll pad is able to perform its function as expected when it was tested to scroll the browser on the computer screen downward and then upward directions successfully.

The third test is to invoke the single left click event by pressing once the left button on the smartphone's screen. The result is shown in the figure below.

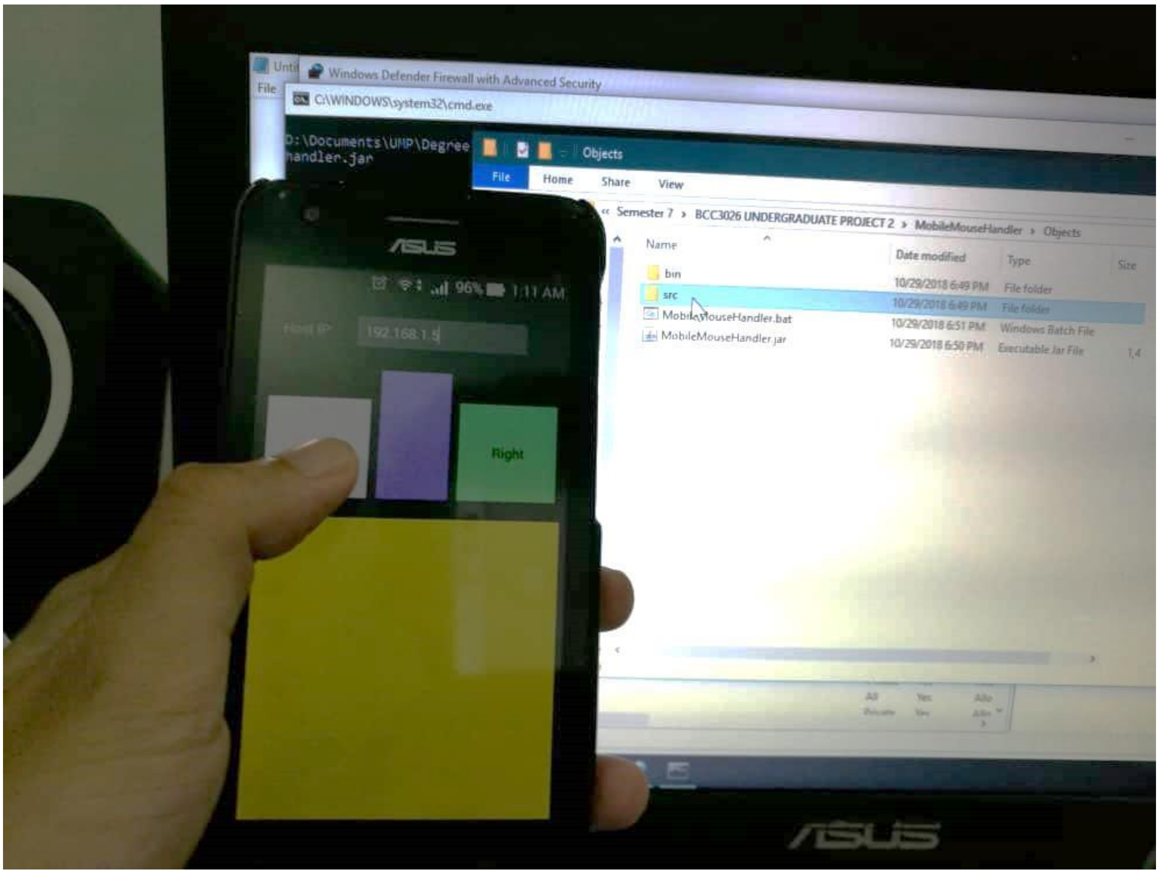


Figure 4.5 Single Press Left Button

Based on Figure 4.5 above, the left button works successfully. It is able to select a folder named "src" when pressing the left button once which invoked the single left click event.

The forth test is to invoke the double left click event by pressing the left button twice on the smartphone’s screen. The result is shown in the figure below.

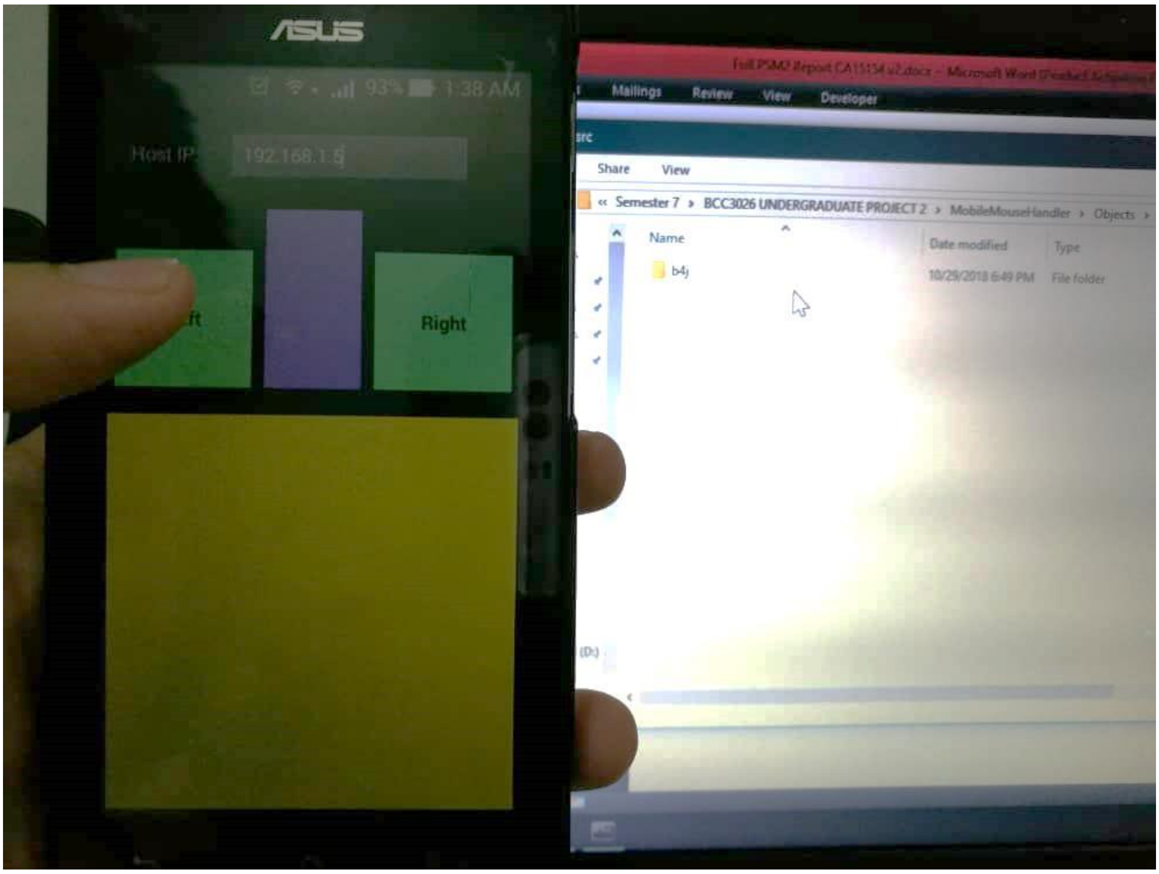


Figure 4.6 Double Press Left Button

Based on Figure 4.6 above, left button works successfully. It is able to open a folder named “src” that was selected by the single click event before and view the selected folder’s content which has a folder named “b4j” inside the “src” folder. Pressing twice on the left button invoked the double left click event.

The fifth test is to invoke the single right click event by pressing the right button once on the smartphone’s screen. The result is shown in the figure below.

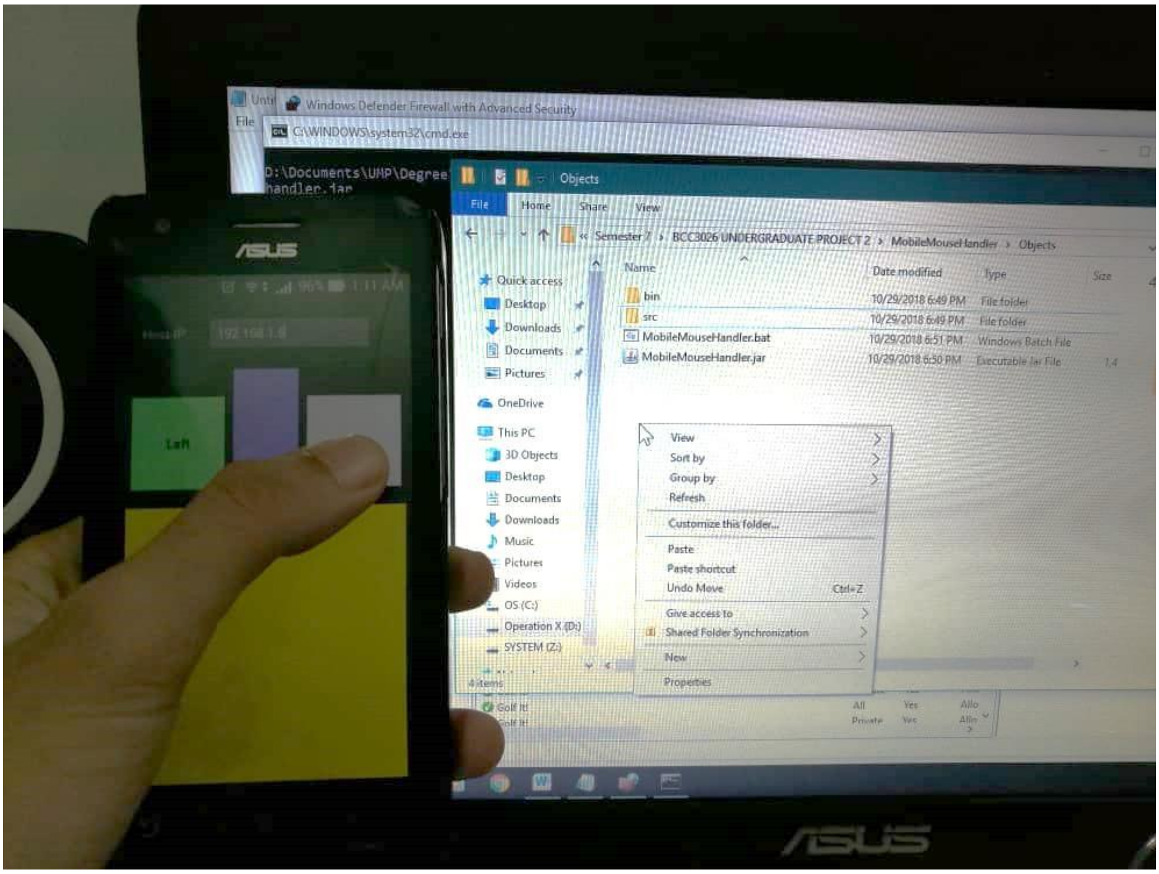


Figure 4.7 Single Press Right Button

Based on Figure 4.7 above, the right button works successfully. It is able to view the drop down menu when pressing once on the right button which invoked the single right click event.

Based on Table 4.2 below, the results were recorded from the clients that had tested the MobMouse application.

Table 4.2 Events Testing and Results

User Actions	Expected Result	Actual Result
Use a finger to touch & move across the smartphone’s screen.	Successfully move mouse cursor.	Similar as expected result.
Use a finger to touch & move across the smartphone’s screen vertically.	Successfully scroll browser.	Similar as expected result.
Single press left button on the smartphone’s screen.	Successfully select a file, folder or application.	Similar as expected result.
Double press left button on the smartphone’s screen.	Successfully open a file, folder or application.	Similar as expected result.
Single press right button on the smartphone’s screen.	Successfully show drop down menu by pressing right button once.	Similar as expected result.

4.5 Conclusion

From this development, the MobMouse application is able to perform as expected. However, some limitations have been identified and will be discussed further in the next chapter which is Chapter 5.

CHAPTER 5

CONCLUSION

5.1 Introduction

From this project development, the goals or objectives set were to develop a mobile application which is the MobMouse application that can function as a touchpad via the smartphone's screen and to evaluate the functionality of the proposed system. The reason for developing MobMouse application is to overcome the issue of having a malfunction laptop's touchpad, to overcome a dead battery mouse, to avoid cable management and to reduce the amount of device carried. The MobMouse application has successfully developed and properly tested for identifying errors or bugs if any.

From the results obtained from the testing done in Chapter 4, the MobMouse application was able to function based on the user requirements where users are:

- Able to move the cursor around on the screen.
- Able to scroll up and down.
- Able to select a folder, file and application by single pressing left button.
- Able to open a folder, file and application by double pressing left button.
- Able to display drop down menu with by single pressing right button.
- Able to connect user's smartphone to their PC.

This project development greatly benefits from RAD methodology because RAD helps reduce development time due to prototyping, reduce manual coding with the help of code generators and code reuse, providing greater flexibility as redesign is done according to the developer, possibility of lesser defects due to prototyping and able to get reliable users' feedback.

From the testing phase of the MobMouse application, there are some limitations found and identified which should be noted for future references. Due to those limitations, there are some future suggestions and enhancements that can be made for MobMouse application for further improvement in the next future versions. The details will be discussed in Chapter 5.3.

5.2 Research Constraints/Limitations

Throughout this project, there are some limitations or constraints that have been discovered during the development of MobMouse application. In this subtopic, the limitations identified during the project development are listed down below:

- MobMouse application will be able to install and run on Android smartphones only.
- MobMouse application can function when user already install the MobMouseHandler.bat program on their PC.
- MobMouse application can function when connected to a wireless network such as WiFi.
- MobMouse application can function when both the user's PC and smartphone are connected to the same network.
- The wireless network range is limited based on the channel frequency used which is 2.4 GHz provided by the router.
- MobMouse application is compatible on Android OS version 4.0 Ice Cream Sandwich and above.

- User must open a specified UDP port for first time setup on their PC to allow communication between the MobMouse application and their PC.
- MobMouse application can communicate with user's PC via UDP only.

5.3 Future Work

This subtopic further discusses on the suggestions and enhancements mentioned in Introduction section which can help improving MobMouse application in the next future versions. Some of the suggestions and enhancements that can be made and implemented to further improve MobMouse application for future updates are listed below:

- Provide support for MobMouse application to be able to install and run on iOS smartphones.
- Improve and simplify user's first time setup on their PC where the MobMouseHandler.bat program can open the specified UDP and TCP ports automatically without user needs to manually open those ports in Windows Firewall.
- Provide support for MobMouse application to communicate with user's PC via TCP.
- Add keyboard support for MobMouse application.
- Add back and forward buttons for MobMouse application that allows user to go back or forward in a folder, PowerPoint slides and browser.

5.4 Conclusion

From this project, the MobMouse application works as intended where user is able to move the cursor on the computer's screen by using a finger to touch & move across the smartphone's screen, user is able to scroll up and down on the computer's screen by using a finger to touch & move vertically on the smartphone's screen, user is able to open file, folder or application by pressing the left button twice, user is able to select file, folder or application by pressing the left button once and user is able to show the drop down menu on the computer's screen by pressing the right button once.

REFERENCES

Anderson, K. (2017, October 4). *What is Rapid Application Development and When Should You Use It?* Retrieved from Capterra: https://blog.capterra.com/what-is-rapid-application-development/

Android (operating system). (2018, October 18). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Android_(operating_system)

Android SDK. (n.d.). Retrieved from Technopedia: https://www.techopedia.com/definition/4220/android-sdk

Android Studio. (2018, October 16). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Android_Studio

B4A. (n.d.). Retrieved from B4X: https://www.b4x.com/b4a.html

B4J. (n.d.). Retrieved from B4X: https://www.b4x.com/b4j.html#FAQ

B4X. (2018, September 13). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/B4X

Barraclough, C. (2015, December 15). *What is iOS and what does iOS stand for?* Retrieved from recombu: https://recombu.com/mobile/article/what-is-ios-and-what-does-ios-stand-for

BASIC. (2018, October 6). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/BASIC

Burrell, H. (2018, July 11). *What is Android?* Retrieved from TechAdvisor: https://www.techadvisor.co.uk/feature/google-android/what-is-android-3680422/

Callaham, J. (2018, July 3). *The history of Android OS: its name, origin and more*. Retrieved from Android Authority: https://www.androidauthority.com/history-android-os-name-789433/

Computer Hope. (2017, April 26). *Cordless Mouse*. Retrieved from Computer Hope: https://www.computerhope.com/jargon/c/cordless-mouse.htm

Computer Hope. (2018, March 8). *Computer Mouse*. Retrieved from Computer Hope: https://www.computerhope.com/jargon/m/mouse.htm

Computer mouse. (2018, October 11). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Computer_mouse

Fedor. (2011). *Bluetooth Touchpad*. Retrieved from Google Play: https://play.google.com/store/apps/details?id=ru.pheodor.android.bluetooth_touchpad

Fisher, T. (2018, September 7). *What Is a Computer Mouse?* Retrieved from Lifewire: https://www.lifewire.com/what-is-a-mouse-2618156

Java Development Kit (JDK). (n.d.). Retrieved from Technopedia: https://www.techopedia.com/definition/5594/java-development-kit-jdk

Johnson. (2017, January 14). *What is an iOS device?* Retrieved from Cleverfiles: https://www.cleverfiles.com/howto/what-is-ios-device.html

Johnston, L. (2018, August 21). *Wired or Wireless Mouse?* Retrieved from Lifewire: https://www.lifewire.com/wired-or-wireless-mouse-2640091

Kate Bruscke. (n.d.). *Why Is Java Important?* Retrieved April 27, 2018, from Chron: http://smallbusiness.chron.com/java-important-30466.html

Lucidchart Content Team. (2018, May 23). *4 Phases of Rapid Application Development Methodology*. Retrieved from Lucidchart: https://www.lucidchart.com/blog/rapid-application-development-methodology

MacNN. (2015, August 17). *B4J IDE released for Windows, with cross-platform app compilation*. Retrieved from macnn: http://www.macnn.com/articles/15/08/17/new.ide.aimed.visual.basic.net.coders.129991/

Mouse. (n.d.). Retrieved from Technopedia: https://www.techopedia.com/definition/2840/mouse

Oleg, R. (2017, March 27). *Remote for Mac*. Retrieved from Google Play: https://play.google.com/store/apps/details?id=com.remote.mac

Powell-Morse, A. (2016, November 23). *Rapid Application Development (RAD): What Is It And How Do You Use It?* Retrieved from Airbrake: https://airbrake.io/blog/sdlc/rapid-application-development

Rapid Application Development (RAD). (n.d.). Retrieved from Technopedia: https://www.techopedia.com/definition/3982/rapid-application-development-rad

Rapid application development. (2018, August 14). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Rapid_application_development

Rouse, M. (2013, December). *Mobile App Definition*. Retrieved from Tech Target: http://whatis.techtarget.com/definition/mobile-app

Rouse, M. (2016, July 29). *rapid application development (RAD)*. Retrieved from TechTarget: https://searchsoftwarequality.techtarget.com/definition/rapid-application-development

Rouse, M. (2018, October 12). *Android Studio*. Retrieved from TechTarget: https://searchmobilecomputing.techtarget.com/definition/Android-Studio

Seagrave, W. (2015, July 26). *B4A: Rapid Android App Development using BASIC*. Retrieved from Google Books: https://books.google.com.my/books?id=_A1ACgAAQBAJ&pg=PA104&lpg=PA104&dq=who+developed+b4a&source=bl&ots=ggJFiXkpjv&sig=ipvx3yAk_PaIKO2Z_QNruRjc4bI&hl=en&sa=X&ved=2ahUKEwiLvpncx8DeAhVEFHIKHWGLCM8Q6AEwDnoECAIQAQ#v=onepage&q&f=false

Sinicki, A. (2017, December 16). *Android SDK tutorial for beginners*. Retrieved from Android Authority: https://www.androidauthority.com/android-sdk-tutorial-beginners-634376/

Sinicky, A. (2016, March 10). *An introduction to Basic4Android – a simple tool for building powerful Android apps*. Retrieved from Android Authority: https://www.androidauthority.com/an-introduction-to-basic4android-678630/

Statista. (2016, June). Retrieved from Statista: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/

Technopedia. (n.d.). *Wireless Mouse*. Retrieved from Technopedia: https://www.techopedia.com/definition/16300/wireless-mouse

Thakur, D. (n.d.). *Rapid Application Development (RAD) Model and its Advantages and Disadvantages of RAD Model*. Retrieved from ECOMPUTER NOTES: http://ecomputernotes.com/software-engineering/rapid-application-development

Twinone. (2015, December 9). *Universal TV Remote*. Retrieved from Google Play: https://play.google.com/store/apps/details?id=org.twinone.irremote

Tyson, M. (2018, August 9). *Development Kit*. Retrieved from JavaWorld: https://www.javaworld.com/article/3296360/core-java/what-is-the-jdk-introduction-to-the-java-development-kit.html

Viswanathan, P. (2017, November 13). *What's a Mobile App?* Retrieved from Lifewire: https://www.lifewire.com/what-is-a-mobile-application-2373354

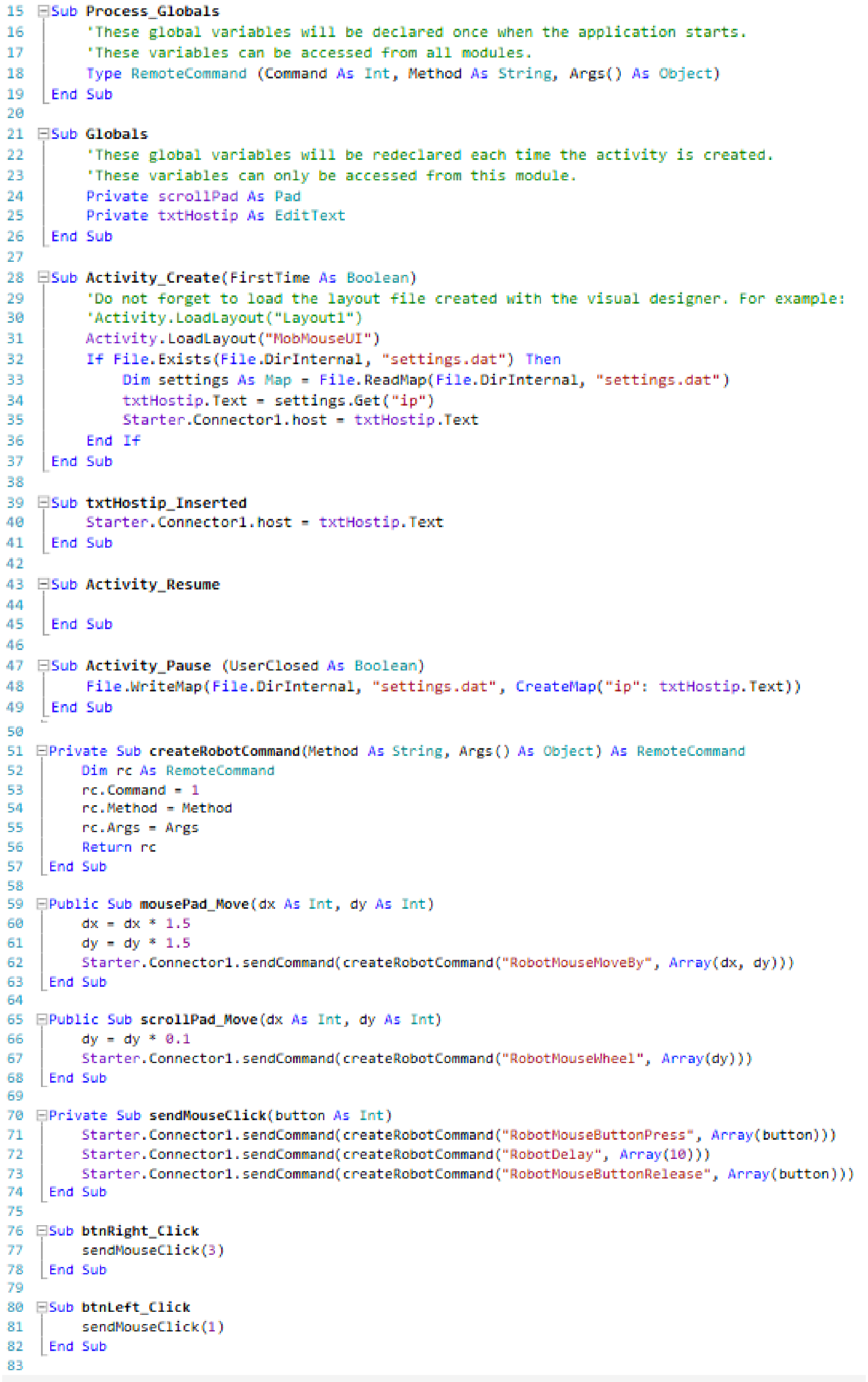
What is an Android mobile phone? (2018, March 2). Retrieved from uSwitch: https://www.uswitch.com/mobiles/guides/android-phones-tablets-and-updates-explained/

What is B4J. (n.d.). Retrieved from Wordpress: https://majgblog.wordpress.com/what-is-b4j/

What is RAD model- advantages, disadvantages and when to use it? (n.d.). Retrieved from TRY QA: http://tryqa.com/what-is-rad-model-advantages-disadvantages-and-when-to-use-it/

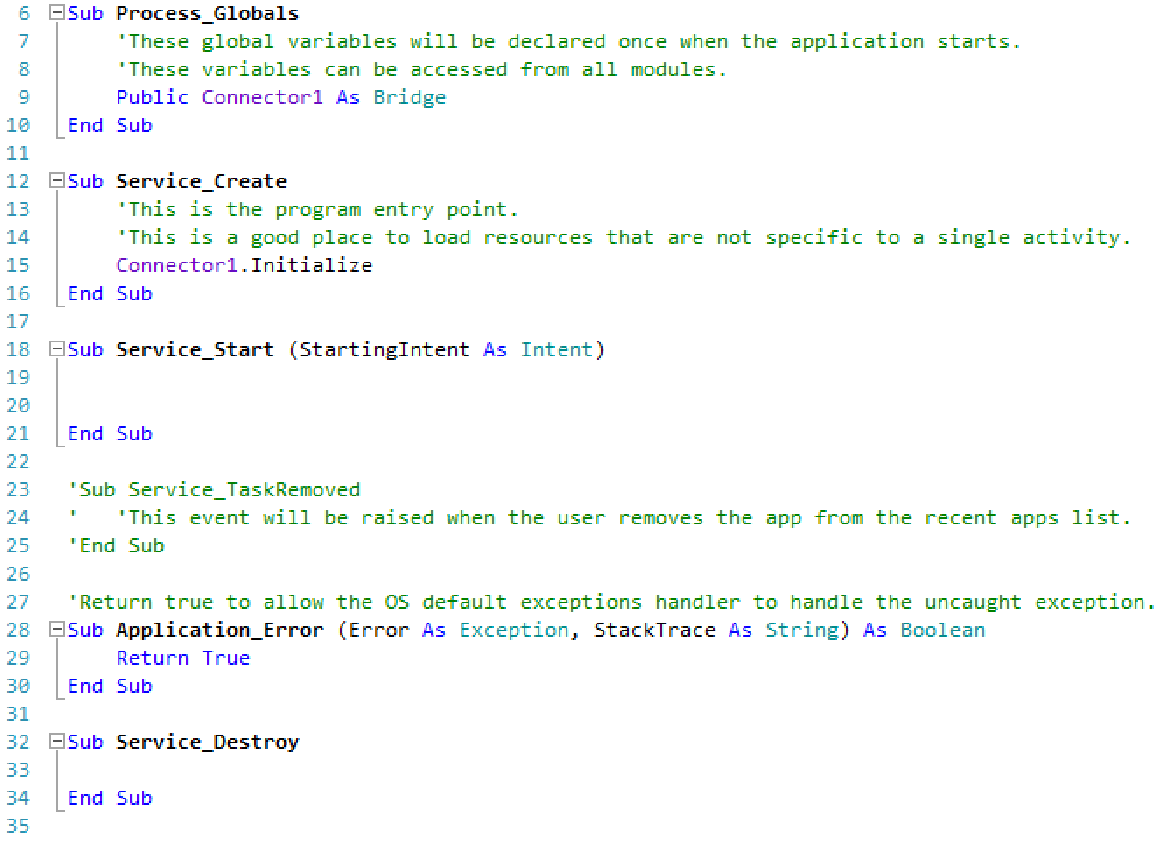
APPENDIX A

MobMouse Main Form (B4A)



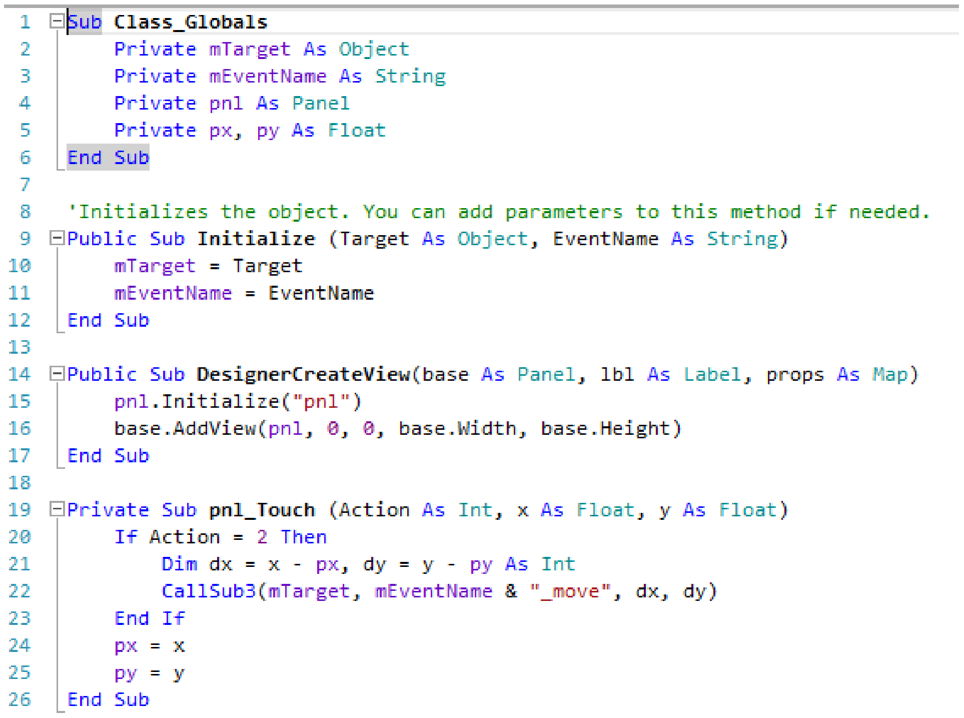
```
Sub Process_Globals
    'These global variables will be declared once when the application starts.
    'These variables can be accessed from all modules.
    Type RemoteCommand (Command As Int, Method As String, Args() As Object)
End Sub

Sub Globals
    'These global variables will be redeclared each time the activity is created.
    'These variables can only be accessed from this module.
    Private scrollPad As Pad
    Private txtHostip As EditText
End Sub

Sub Activity_Create(FirstTime As Boolean)
    'Do not forget to load the layout file created with the visual designer. For example:
    'Activity.LoadLayout("Layout1")
    Activity.LoadLayout("MobMouseUI")
    If File.Exists(File.DirInternal, "settings.dat") Then
        Dim settings As Map = File.ReadMap(File.DirInternal, "settings.dat")
        txtHostip.Text = settings.Get("ip")
        Starter.Connector1.host = txtHostip.Text
    End If
End Sub

Sub txtHostip_Inserted
    Starter.Connector1.host = txtHostip.Text
End Sub

Sub Activity_Resume

End Sub

Sub Activity_Pause (UserClosed As Boolean)
    File.WriteMap(File.DirInternal, "settings.dat", CreateMap("ip": txtHostip.Text))
End Sub

Private Sub createRobotCommand(Method As String, Args() As Object) As RemoteCommand
    Dim rc As RemoteCommand
    rc.Command = 1
    rc.Method = Method
    rc.Args = Args
    Return rc
End Sub

Public Sub mousePad_Move(dx As Int, dy As Int)
    dx = dx * 1.5
    dy = dy * 1.5
    Starter.Connector1.sendCommand(createRobotCommand("RobotMouseMoveBy", Array(dx, dy)))
End Sub

Public Sub scrollPad_Move(dx As Int, dy As Int)
    dy = dy * 0.1
    Starter.Connector1.sendCommand(createRobotCommand("RobotMouseWheel", Array(dy)))
End Sub

Private Sub sendMouseClick(button As Int)
    Starter.Connector1.sendCommand(createRobotCommand("RobotMouseButtonPress", Array(button)))
    Starter.Connector1.sendCommand(createRobotCommand("RobotDelay", Array(10)))
    Starter.Connector1.sendCommand(createRobotCommand("RobotMouseButtonRelease", Array(button)))
End Sub

Sub btnRight_Click
    sendMouseClick(3)
End Sub

Sub btnLeft_Click
    sendMouseClick(1)
End Sub
```

MobMouse Starter (B4A)



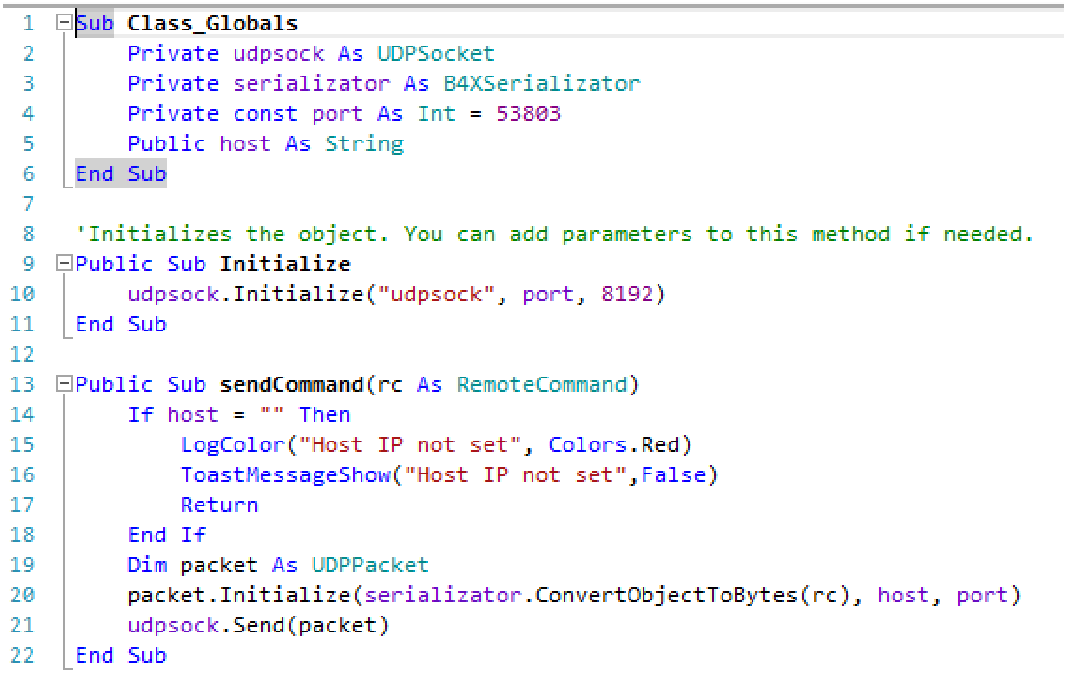
```
Sub Process_Globals
    'These global variables will be declared once when the application starts.
    'These variables can be accessed from all modules.
    Public Connector1 As Bridge
End Sub

Sub Service_Create
    'This is the program entry point.
    'This is a good place to load resources that are not specific to a single activity.
    Connector1.Initialize
End Sub

Sub Service_Start (StartingIntent As Intent)


End Sub

'Sub Service_TaskRemoved
'    'This event will be raised when the user removes the app from the recent apps list.
'End Sub

'Return true to allow the OS default exceptions handler to handle the uncaught exception.
Sub Application_Error (Error As Exception, StackTrace As String) As Boolean
    Return True
End Sub

Sub Service_Destroy

End Sub
```

MobMouse Pad (B4A)



```
Sub Class_Globals
    Private mTarget As Object
    Private mEventName As String
    Private pnl As Panel
    Private px, py As Float
End Sub

'Initializes the object. You can add parameters to this method if needed.
Public Sub Initialize (Target As Object, EventName As String)
    mTarget = Target
    mEventName = EventName
End Sub

Public Sub DesignerCreateView(base As Panel, lbl As Label, props As Map)
    pnl.Initialize("pnl")
    base.AddView(pnl, 0, 0, base.Width, base.Height)
End Sub

Private Sub pnl_Touch (Action As Int, x As Float, y As Float)
    If Action = 2 Then
        Dim dx = x - px, dy = y - py As Int
        CallSub3(mTarget, mEventName & "_move", dx, dy)
    End If
    px = x
    py = y
End Sub
```

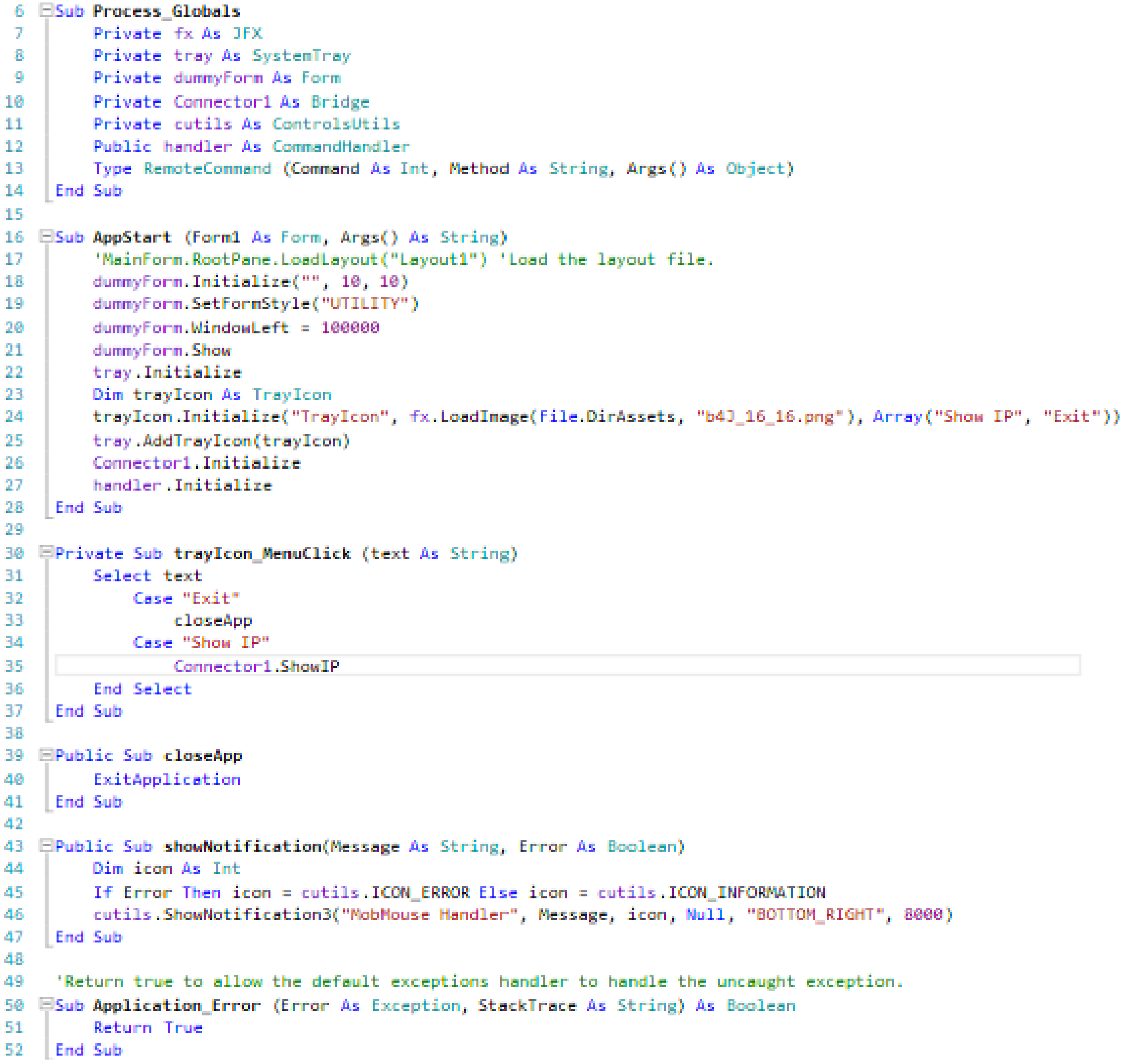
MobMouse Bridge (B4A)



```
Sub Class_Globals
    Private udpsock As UDPSocket
    Private serializator As B4XSerializator
    Private const port As Int = 53803
    Public host As String
End Sub

'Initializes the object. You can add parameters to this method if needed.
Public Sub Initialize
    udpsock.Initialize("udpsock", port, 8192)
End Sub

Public Sub sendCommand(rc As RemoteCommand)
    If host = "" Then
        LogColor("Host IP not set", Colors.Red)
        ToastMessageShow("Host IP not set",False)
        Return
    End If
    Dim packet As UDPPacket
    packet.Initialize(serializator.ConvertObjectToBytes(rc), host, port)
    udpsock.Send(packet)
End Sub
```

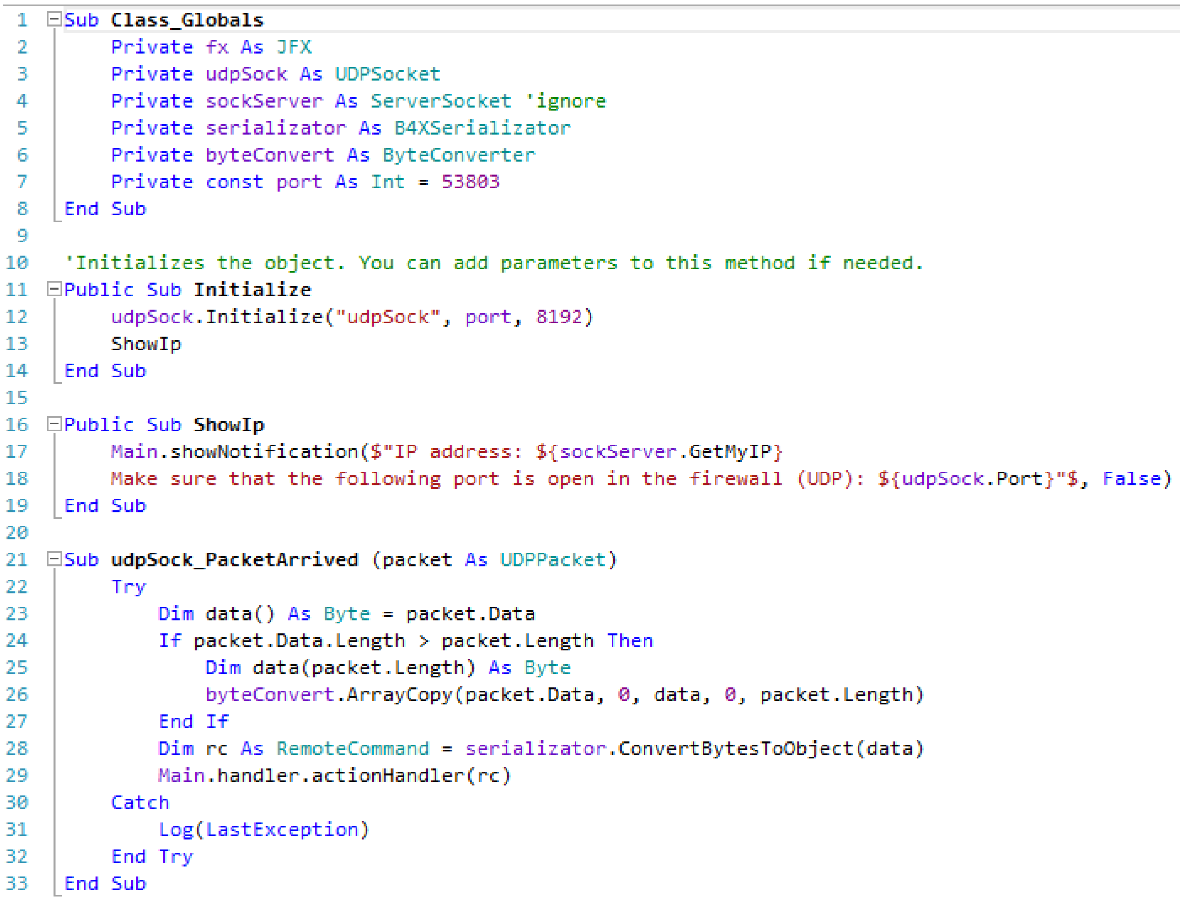
APPENDIX B

MobMouseHandler Main Form (B4J)



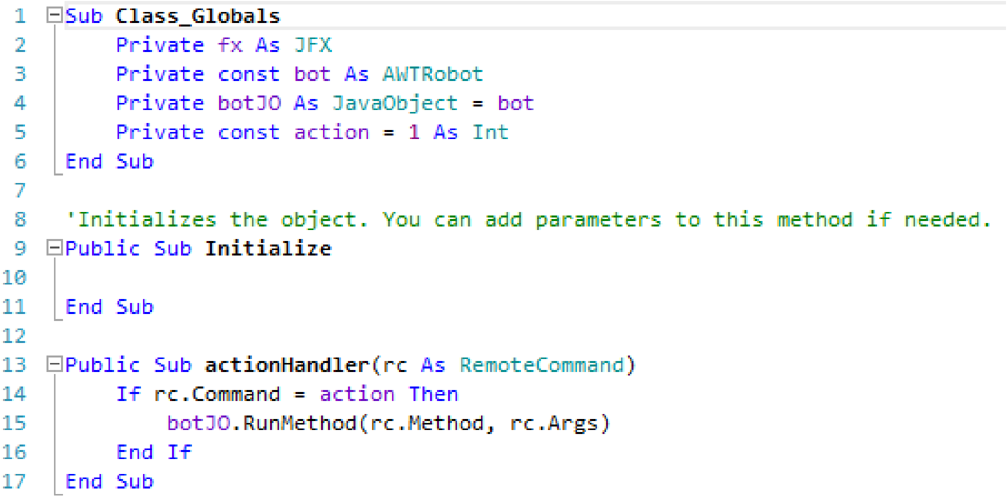
```
Sub Process_Globals
    Private fx As JFX
    Private tray As SystemTray
    Private dummyForm As Form
    Private Connector1 As Bridge
    Private cutils As ControlsUtils
    Public handler As CommandHandler
    Type RemoteCommand (Command As Int, Method As String, Args() As Object)
End Sub

Sub AppStart (Form1 As Form, Args() As String)
    'MainForm.RootPane.LoadLayout("Layout1") 'Load the layout file.
    dummyForm.Initialize("", 10, 10)
    dummyForm.SetFormStyle("UTILITY")
    dummyForm.WindowLeft = 100000
    dummyForm.Show
    tray.Initialize
    Dim trayIcon As TrayIcon
    trayIcon.Initialize("TrayIcon", fx.LoadImage(File.DirAssets, "b4J_16_16.png"), Array("Show IP", "Exit"))
    tray.AddTrayIcon(trayIcon)
    Connector1.Initialize
    handler.Initialize
End Sub

Private Sub trayIcon_MenuClick (text As String)
    Select text
        Case "Exit"
            closeApp
        Case "Show IP"
            Connector1.ShowIP
    End Select
End Sub

Public Sub closeApp
    ExitApplication
End Sub

Public Sub showNotification(Message As String, Error As Boolean)
    Dim icon As Int
    If Error Then icon = cutils.ICON_ERROR Else icon = cutils.ICON_INFORMATION
    cutils.ShowNotification3("MobMouse Handler", Message, icon, Null, "BOTTOM_RIGHT", 8000)
End Sub

'Return true to allow the default exceptions handler to handle the uncaught exception.
Sub Application_Error (Error As Exception, StackTrace As String) As Boolean
    Return True
End Sub
```

MobMouseHandler Bridge (B4J)



```
Sub Class_Globals
    Private fx As JFX
    Private udpSock As UDPSocket
    Private sockServer As ServerSocket 'ignore
    Private serializator As B4XSerializator
    Private byteConvert As ByteConverter
    Private const port As Int = 53803
End Sub

'Initializes the object. You can add parameters to this method if needed.
Public Sub Initialize
    udpSock.Initialize("udpSock", port, 8192)
    ShowIp
End Sub

Public Sub ShowIp
    Main.showNotification($"IP address: ${sockServer.GetMyIP}
    Make sure that the following port is open in the firewall (UDP): ${udpSock.Port}"$, False)
End Sub

Sub udpSock_PacketArrived (packet As UDPPacket)
    Try
        Dim data() As Byte = packet.Data
        If packet.Data.Length > packet.Length Then
            Dim data(packet.Length) As Byte
            byteConvert.ArrayCopy(packet.Data, 0, data, 0, packet.Length)
        End If
        Dim rc As RemoteCommand = serializator.ConvertBytesToObject(data)
        Main.handler.actionHandler(rc)
    Catch
        Log(LastException)
    End Try
End Sub
```

MobMouseHandler Command Handler (B4J)



```
Sub Class_Globals
    Private fx As JFX
    Private const bot As AWTRobot
    Private botJO As JavaObject = bot
    Private const action = 1 As Int
End Sub

'Initializes the object. You can add parameters to this method if needed.
Public Sub Initialize

End Sub

Public Sub actionHandler(rc As RemoteCommand)
    If rc.Command = action Then
        botJO.RunMethod(rc.Method, rc.Args)
    End If
End Sub
```

APPENDIX C

	NAME	DATE
Tested & Verified by: ___________________ Developer		
Tested by: ___________________ Client		
Tested by: ___________________ Client		
Tested by: ___________________ Client		
Tested by: ___________________ Client		
Tested by: ___________________ Client		