Remote Presentation System, Android Based

By

Huah Kok Yu

A PROPOSAL

SUBMITTED TO

University Tunku Abdul Rahman

In partial fulfilment of the requirements

for the Degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology

(Perak Campus)

JAN 2014

DECLARATION OF ORIGINALITY

I declare that this report entitled "**Remote Presentation System, Android Based**" is my own word except as cited in the references. This report has not been accepted for my degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ________________________

Name : ________________________

Date : ________________________

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my project supervisor, Dr. Alex Ooi Boon Yaik, who had given my invaluable guidance, advice, and patience during completing the project. Dr. Alex had guided me on choosing the project title, and teaches me in knowing how to deal with problem solving throughout the process of the project. And, an acknowledgments to all my course mates for sharing ideas and valuable feedbacks that help me through the project.

To my family and friends, who had given me unconditional supports, encouragement and love, and standing by my side during hard times, thank you.

ABSTRACTS

This purpose of the project is to develop an application software to assist presenter during the presentation, also named as Remote Presentation System, Android Based. The proposed application software is aimed to improve the experience of a stage presentation. It enable the presenter to have a better control on the process flow of the presentation, thus, letting the audience to enjoy more. The application software is designed to run on Android based smart phones, where the presenter is able to use the smart phone to control everything that happens in the presentation. The application software is built with rich graphics, where is easy for the user to learn.

The application software come in the package of two parts, one part is client part, where the presenter's smart phone is required to equip with. Another part is server side, where need to run in the computer where all the presentations materials stores. These two parts of the application software will communicate with each other via local network.

The methodology used to develop this application software is incremental method, where all the user requirements are arranged in priority and is developed accordingly. Several existing solution had been reviewed, and the weakness and strength for the specific application is being analysed. This is to make to make sure that the developed application software does not meet the limitation of the previous work, and does stand out among the strength of the existing solution.

TABLE OF CONTENTS

List of Tables

List of Figures

Chapter 1

1.1 Project Background

Presentation is the practice of showing and explaining the content of a topic to audiences. No matter in what field a person is involve, when come to selling an idea, explaining a topic or to share a certain information, presentation is needed. (Dictionary.com, 2012)

In traditional presentation, most presentation is carry out by a presenter presenting a topic to room of audiences through slides. The information is organised in the form of slides and control by the presenter through the computer. This way of traditional presentation is able convey information, but is in a very limited way, in such the presentation slides can only be controlled from computer, and anything else are required to go through the computer, such as controlling the executing hyperlink, or showing another source of information such as a pdf. Thus, this way of presentation often limit the movement of the presenter.

When the presenter need to change the slide, presenter need to pause the presentation, and then walk to the computer to change the slides. This always been a critical problem to presenter, as it slow down the overall process of the presentation, and often disrupt the presenter presentation flow. This does not happens only to the change of slides, but anything that require the computer to control.

Another common problem to the presenter is the pointer to the slides. For most on the time, presenter will personally walk to the projected area to point on the specific spot, just to show the audience the where about of discussion. For some case, the presenter will use stick or a laser pointer. There are several gadgets that is able to help out the presenter on the pointer problems, but anyhow, the pointer is only available on the slide show surface, without any solid functionality others than pointing.

Although there are still several gadgets which is able to help the presenters to control the flow of the presentations such as software application system, laser pointer, microphone and so on, but there are times, where the presenters are not sure how the venue is like and what technical support is provided, until before they are going to make the presentation. It is not practical for the presenters to carry every gadgets along with them everywhere, and realise that there are any supports for their gadgets they bring along.

With the technologies we have today, we could do so much more to make improvement on the traditional presentation. The proposed application, an application which is able to bring out all the functionality of each presentations gadget and combine it into one hand. The application software runs on Android base smart phone, which turn the smart phone into a presentation device which is distantly manipulate the slides and control the most critical part of the presentation using the smart phone.

The application software is able to assist the presenter to deliver a better quality of presentation, by improving the flow the presentation, and providing better control on the presentation. Thus improve overall experience of presentation.

1.2 Problem Statement

In a traditional presentation, the presenter rely on the computer to manipulate the slides. This often arise challenges to the presenter, as the presenter need to get back to the computer every time the presenter need to change the slide. So, there will always be a pause on the flow of presentation when the presenter are about to change the slides and the movement of the presenter is also being limited by the computer as the presenter need to stick to the computer to control the slides.

- Presenter is bind to the computer for the needs of changing slides.
- Presenter is bind to the computer for the use of cursor pointer.

For the presenter to further explain or elaborate the slides, normally the presenter will do it on the whiteboard for visual support, but it is hard for presenters to knows what is prepared for them before they are in the venue. Thus, most of the time explanation can only be done verbally.

- Presenter is unable to provide visual explanation other than what is on the slide.
- Wired microphone bind the movement of the presenter.

1.3 Project Scope

The software application is develop to run on Windows operating system on desktop or laptop computer, and Android base smartphone. It will pair the smart phone with the computer, then use the smart phone remotely control the computer. The connection is via Wi-Fi connection.

The scope of this project involve the Android base smart phone remotely to control the flow of slides on the paired computer, write or draw on the slides in the slide by detecting the drawing the in mobile phone, and remotely control the cursor on the computer to do so. And finally play the slide on the projector to provide visual explanation to the audience.

Last but not least, is to enable the smart phone to transmit audio, the presenter speech, to the computer then play it on the speaker for better audio support.

The server must be equip with java platform, and presentation software is preferable using Microsoft PowerPoint for best performance.

1.4 Objectives

i. To develop a system that is able to remotely control the slide shows through mobile phones.
   - Remotely control the slides from mobile instead of computer. The presenter is able to switch the current slide to next or previous slide through the presenter mobile device.
   - Use the mobile device can remotely control the cursor movement, and mouse click using smartphone as a touchpad.

ii. To develop a system that is able to improve the visual and audio support for the presentation through mobile phones.
   - Presenter is able to draw or write on the slides through the mobile device to provide visual explanations to the audiences.
   - Use the mobile device as a microphone by connecting the smartphone to the computer speaker.

1.5 Contribution

With the proposed application software, the experience of presentation for the presenter and audiences will never be the same again. The idea of making full use of the technologies, and put it into one hand, and improve the way of traditional presentation.

The application software is able to smoothen the presentation flow by eliminate the mini pause on the presentation, and improve the way the presenter can explain the slides. This can be done by providing extra visual support to the audience by writing or drawing on the slides, opening extra source of information to show audiences. It also provide the functionality of acting as a pointer on the slides and as a microphone for the presenter. This application software is able to provide what the presenter needs in one single application, in one single device.

The application portability is high, as the server can run in any computer that equips with Java. The user can transfer the executable jar file to the computer, and start running the applicati

Chapter 2

2.1 Literature Review

There are several presentation application system been develop to help and solve the problem with delivering a better quality of presentation. Most of these system application comes in single gadget, with only one functionality. For example, for enhancing the audio of the presentation, microphone is used. Microphone itself also undergo evolution from wired microphones to wireless microphones, and to overhead microphones or lavalier microphones (a.k.a. clip mic or personal mic).

The microphone served a great purpose on enhancing the audio of the presenter, and did a great job by filtering noises and preserved the clarity of the presenter voice. But the price on a good microphone will be a main concern for the presenter. There are also several occasion where the venue is unable to guarantee the present of a working microphone to the presenter.

For the most common problems of presenter, rely on the computer to control the slides or pointer, has been limiting the body language of the presenter and waste of time when the presenter had to go to the computer to change slide or use the computer's mouse. The introduced solution to this problem is slideshow control device, one of the example, Logitech Professional Presenter R800. The presentation device have 30m range wireless connectivity to the presenter computer, and the presenter is able to use the device to play the next or previous slides, and the device also provide a laser pointer so the presenter is able to point to the specific part of the slides.

These hardware been helpful in presentations, but it is still troublesome for presenters to bring all these hardware to everywhere they are going to do a presentation, as some of the hardware only serve certain purpose.

Hence, LogicInMind had introduced Slideshow Remote™ for PowerPoint. This application software is designed running on iOS, for remotely control the slideshows in the computer. This application paired with computer via Wi-Fi or Bluetooth, and is able to display next or previous slide, jump from slides, and is able control the cursor using the mobile device. Soon after Android been launch, there are several application been released for slideshow controlling purpose, such as i-Click PowerPoint Remote, PowerPoint Keynote Remote Pro. These application also serve the purpose of remotely control the slides using mobile device, which include the next and back function. Where all the functionality provided are MS PowerPoint based, which only works on PowerPoint.

There are others software like Unified Remote, which serve the purpose of remotely control the computer using Wi-Fi. This software provide extra features other than controlling slideshows. It also provide the remote control on the computer settings and file explorer on computer

The functionality provided by some of the application is useful for the presenters, as the application enable the presenter to remotely control the computer from a distant. And some gadgets are able to serve the functionality to the presenter, but in a much separated way. Each of the gadgets mostly serve only a certain purpose, which did not cover all the aspect of presentation.

2.2 Fact Findings

2.2.1 Unified Remote (play.google.com/store, 2013)

Android-PC remote, an application that is able to turn Android device into a Wi-Fi or Bluetooth universal remote control for paired Windows PC. Control user favorite programs, mouse, and keyboard.

Features

- Basic Input (mouse and keyboard)
- Media (access to Windows Media Center / Windows Media Player / Windows Photo Viewer / Winamp / iTunes)
- File Manager (access file directory, and create folders)
- Slide Show (control the animation of the slideshow, and make notes on the slideshow)
- Browser supports (Chrome / Firefox / Opera)
- Custom Remotes

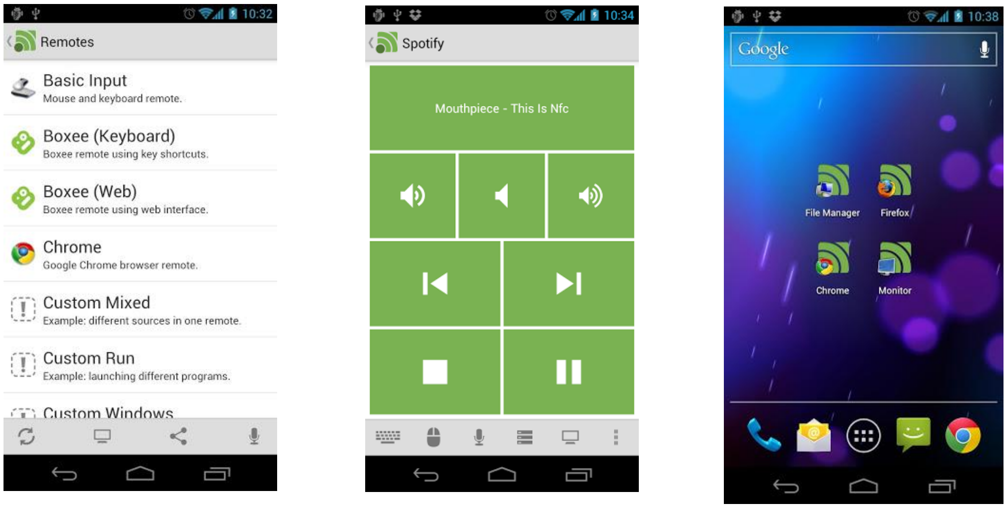
Latest version update: June 19, 2013

System Requirement: Android 1.5 and up

Microsoft Windows

Price: USD 3.73

Figure 2.1 show the application screenshot



10:32

Remotes

Basic Input
Mouse and keyboard remote.

Boxee (Keyboard)
Boxee remote using key shortcuts.

Boxee (Web)
Boxee remote using web interface.

Chrome
Google Chrome browser remote.

Custom Mixed
Example: different sources in one remote.

Custom Run
Example: launching different programs.

Custom Windows

Figure 2.1 Application Screenshot

2.2.2 i-Clickr PowerPoint Remote (play.google.com/store, 2013)

An application that turn Android-powered device as PowerPoint / OpenOffice / Keynote remote clicker. It allow user to control presentation directly from phone using Wi-Fi.

Features

- Complete control on the PowerPoint slides including animations.
- Help keep track of the presentation time.
- View slides and slide notes on the handset.

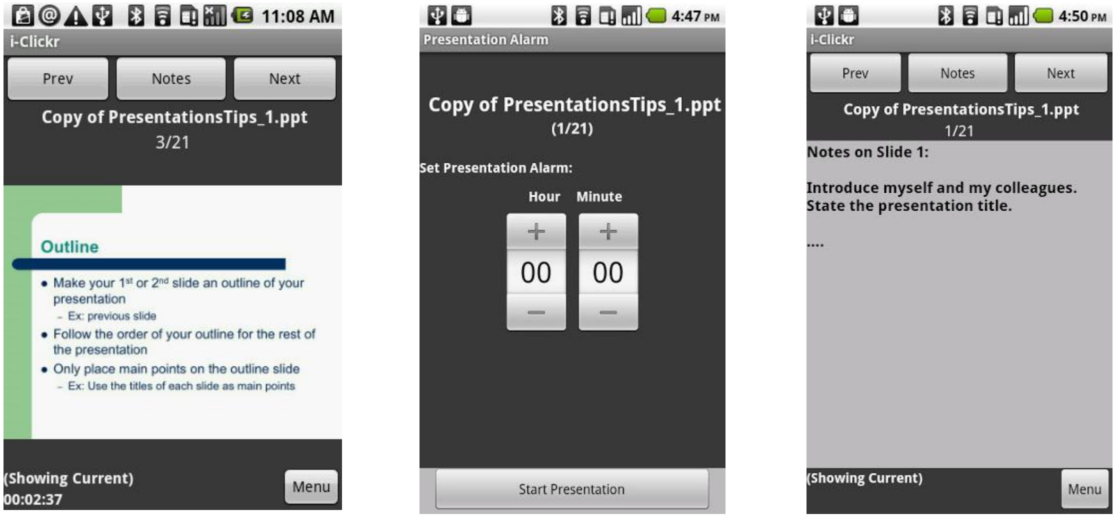
Latest Update: May 28, 2013

System Requirement: Android 2.3.3 and up

Microsoft Window XP and later / Mac OS X10.7+, with PowerPoint 2000 to 2010

Price: USD 9.99

Figure 2.2 show the application screenshot



Copy of PresentationsTips_1.ppt
3/21

Outline

- Make your 1st or 2nd slide an outline of your presentation
  - Ex: previous slide
- Follow the order of your outline for the rest of the presentation
- Only place main points on the outline slide
  - Ex: Use the titles of each slide as main points

(Showing Current)
00:02:37

Copy of PresentationsTips_1.ppt
(1/21)

Set Presentation Alarm:

Hour: 00 Minute: 00

Copy of PresentationsTips_1.ppt
1/21

Notes on Slide 1:

Introduce myself and my colleagues.
State the presentation title.

....

(Showing Current)

Figure 2.2 Application Screenshot

2.2.3 PowerPoint Keynote Remote (play.google.com/store, 2013)

An application that provide control to PowerPoint and Keynote presentation via Bluetooth and Wi-Fi using Android.

Features

- Complete control on the flow of slides including next, back, go to specific slide.
- Preview slides on the phone
- Laser pointer and annotation
- Video and audio support for the slide previewed on the phone

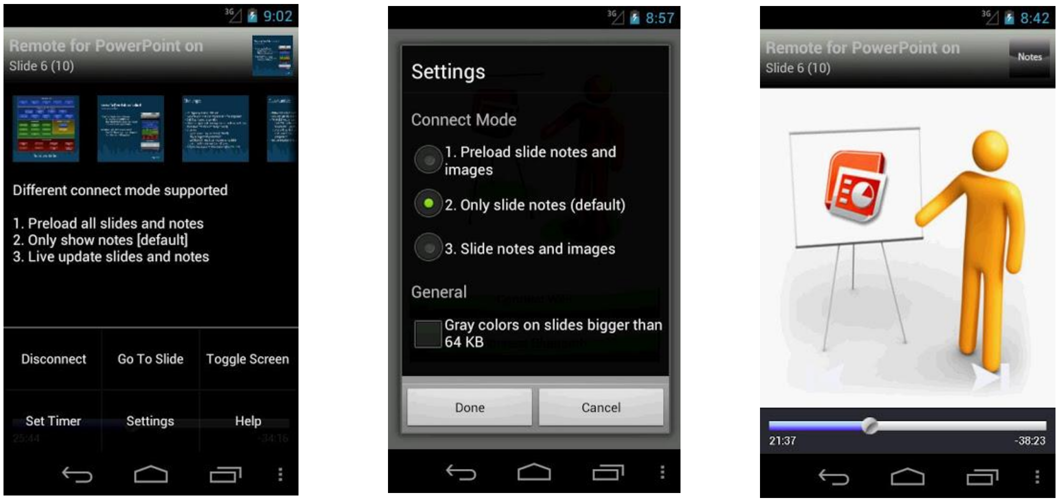
Latest Update: June 23, 2013

System Requirement: Android 2.0.1 and up

Microsoft Windows XP and later / Mac OS X 10.7+, with any version of PowerPoint

Price: USD 5.96

Figure 2.3 shows the application screenshot.



Remote for PowerPoint on
Slide 6 (10)

Different connect mode supported

1. Preload all slides and notes
2. Only show notes [default]
3. Live update slides and notes

Settings

Connect Mode

○ 1. Preload slide notes and images

● 2. Only slide notes (default)

○ 3. Slide notes and images

General

□ Gray colors on slides bigger than 64 KB

Remote for PowerPoint on
Slide 6 (10)

Figure 2.3 Application Screenshot

2.2.4 Slideshow Remote™ for PowerPoint (itunes.apple.com, 2012)

An application the designed for both iPhone and iPad, a remote control for Windows PowerPoint and mobile viewer. The application can control the user PC, transfer files with iTune (Mac and Windows), open presentation attached to emails, view the slides, display on HDTV and share via AirPlay.

Features

- View current and next slide, jump from slide to slide, and browse slides in thumbnail view.
- View or edit presenter's notes
- Timer to acknowledge presenter elapsed time.
- AirPlay sharing, to share the slides with audience.
- View on Apple TV.
- Control mouse cursors and customize the cursor

Latest Update: Nov 09, 2012

System Requirement: iOS 3.1.2 or later

Microsoft Windows XP or later, with PowerPoint 2003 or later.

Price: 4.99

Figure 2.4 shows the application screenshot.

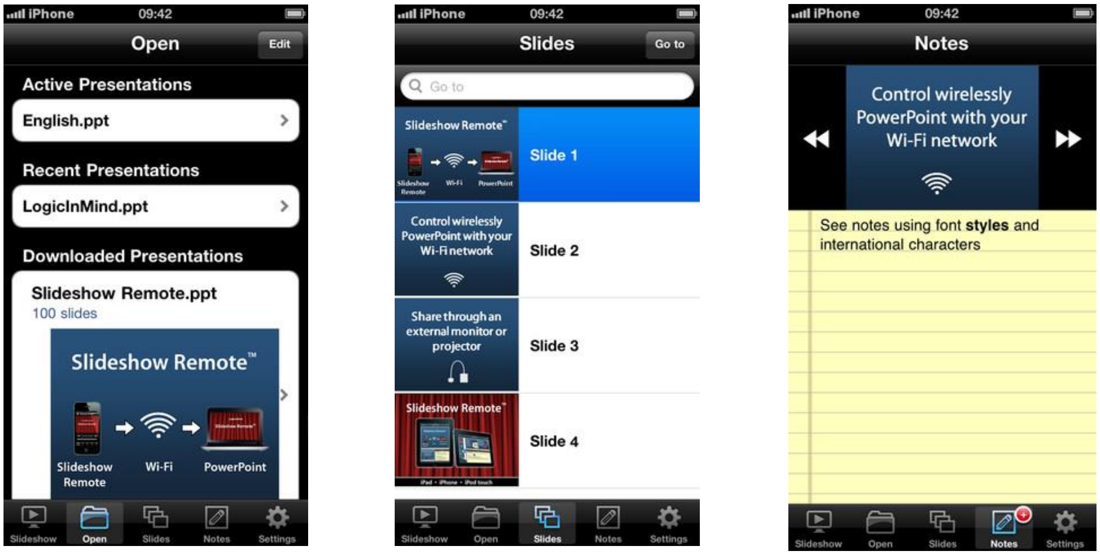


Figure 2.4 Application Screenshot

2.2.5 PG185TQG Condenser Lavalier Microphone (shure.co.uk.com, 2013)

A visually discreet electret condenser lavalier microphone, suitable for spoken word and presentation. A low visibility with high-quality professional audio support, clear sound for speech and vocal application.

Featured components

- Transmitter
- Receiver

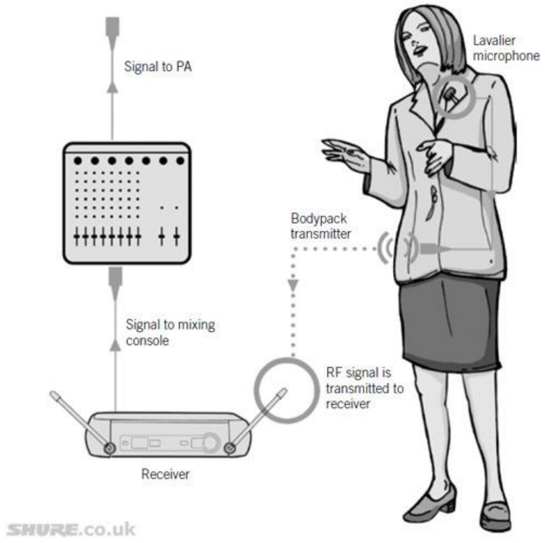


Figure 2.5 Microphone Architecture

Price: USD 100.1

2.2.6 Logitech Professional Presenter R800 (logitech.com, 2011)

A wireless presenter, with laser pointer and intuitive slideshow controls which helps presenter make high-impact presentations.

Features

- Laser pointer, a powerful and easy to see green laser.
- Intuitive slideshow controls, jump from first slide to last slide, next slide and previous slide.
- Timer, to keep track of time, can alert presenter with a glance or vibrating alert.

System Requirements: Windows XP or later

USB port, plug-and-play receiver

Price: USD 79.99

2.4 Critical Remarks

2.4.1 Evaluation Criteria

Table 2.1 shows the evaluation of existing application or devices used in presentation.

	Control slides	**Write/Draw on the slides**	**Act as pointer**	**Act as microphone**	**Cost (USD)**
Slideshow Remote™	✔	✔	✔		4.99
PowerPoint Keynote Remote	✔		✔		5.96
i-Clickr ProwerPoint Remote	✔				9.99
Unified Remote	✔	✔			3.73
Lavalier Microphones				✔	100.1
Logitech Professional Presenter R800	✔		✔		79.99
Proposed Application Software	✔	✔	✔	✔	

Table 2.1

2.4.2 Criteria Description

Table 2.2 show the elaboration of the term used in evaluation criteria.

Criteria	Elaboration
Control slides	The presenter is able to control the flow of the slides, such as play the next slide, previous slide, jump from slide to slide.
Write or draw on the slides	The presenter is able to draw or write something on the slide and display on the projector to give extra visual support to audience.
Act as pointer	The device is able to act as a pointer to the slides, by using laser light to point on the screen in a distance instead of using a cursor.
Act as microphone	The device is able to act as a microphone by transmitting the presenter sounds to the speaker to enhance the volume.
Cost	The cost for the product in USD currency.

Table 2.2

Chapter 3

3.1 Methodology and Tools

3.1.1 Methodology

The software development methodology that will be used to develop the application system is incremental development.

The iterative and incremental development divide the system functionality into portions, and the each portion of the functionality is given with priority, then system portion with highest priority will be develop first.

The development model is show in Figure 3.1.

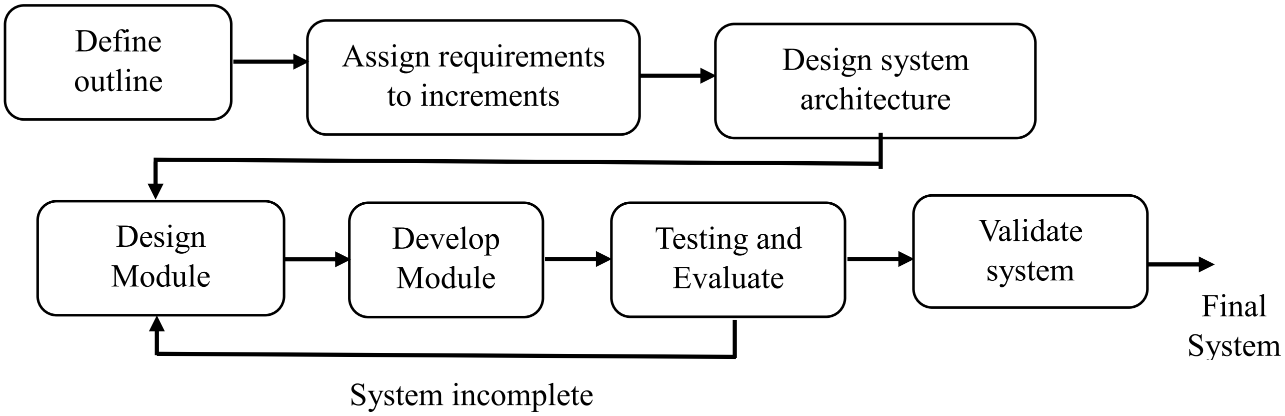


Figure 3.1 (Robabdul.com, 2005)

Define Outline:

To develop an application for mobile that are able to assist presenter during presentation.

Assign requirement to increment:

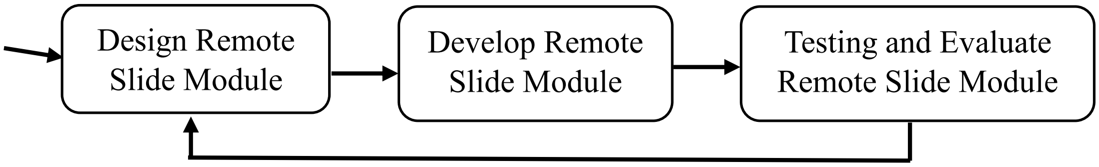
1. Remotely control the slides from mobile instead of computer.
2. Remotely control the computer cursor from mobile.
3. Write or draw on the mobile and display on the screen.
4. Use the mobile as a microphone to enhance the volume.

Design System architecture:

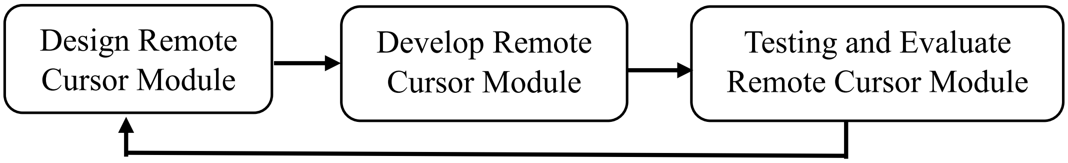
Design the use case diagram, activity diagram, class diagram, sequence diagram, and the architecture design of the system.

Develop system, validate increment, and integrate increment:

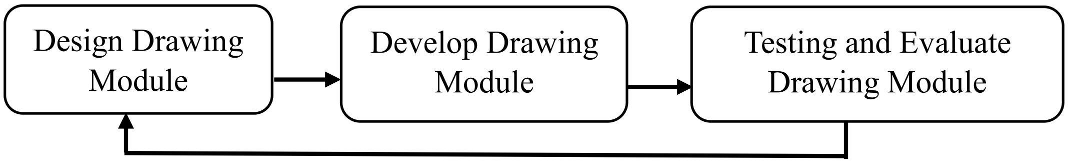
1. First loop: Remotely control the slides from mobile instead of computer.


Design Remote Slide Module
Develop Remote Slide Module
Testing and Evaluate Remote Slide Module

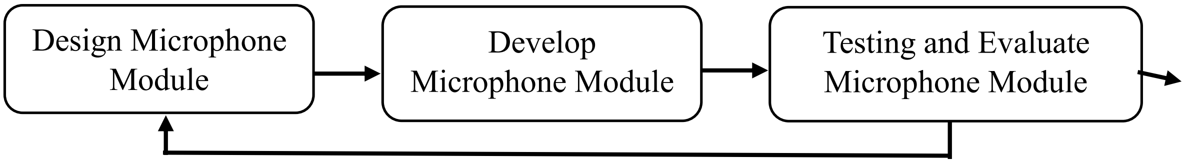
2. Second loop: Remotely control the computer cursor from mobile.


Design Remote Cursor Module
Develop Remote Cursor Module
Testing and Evaluate Remote Cursor Module

3. Third loop: Write or draw on the mobile and display on the screen.



4. Last loop: Use the mobile as a microphone to enhance the volume.


Design Microphone Module
Develop Microphone Module
Testing and Evaluate Microphone Module

Validate System:

Test each module are integrating as a whole system. Information passing between client and server are stable and workable.

Final system:

Test case is created and used to test with the system in the real environment.

3.1.2 Technologies Involved

Devices

Server, Notebook/Desktop computer (Windows) with java installed.

Mobile Phones (Android)

Router (Wi-Fi, Internet)

Projector, speaker

Computer Language

Java Development Tools: Developing Android application for smart phone, server application for computers.

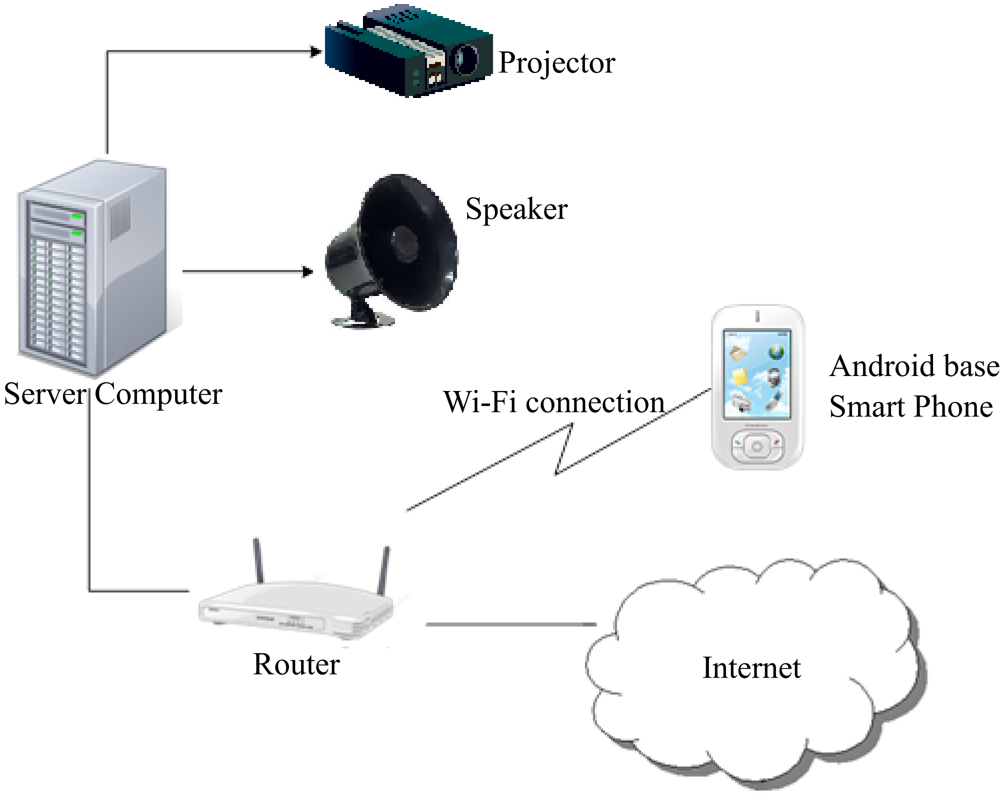


Figure 3.2 shows the technologies that involve in the system.

3.1.3 Architecture Design

Figure 3.3 shows the architecture design for the proposed system, Android Base Remote Presentation System.

Android Base Remote Presentation System							
Phone (Android)				Computer (Windows)			
Slide Control	Cursor Control	Drawing Tools	Microphone	Power Point	Cursor	Projector Driver	Speaker Driver

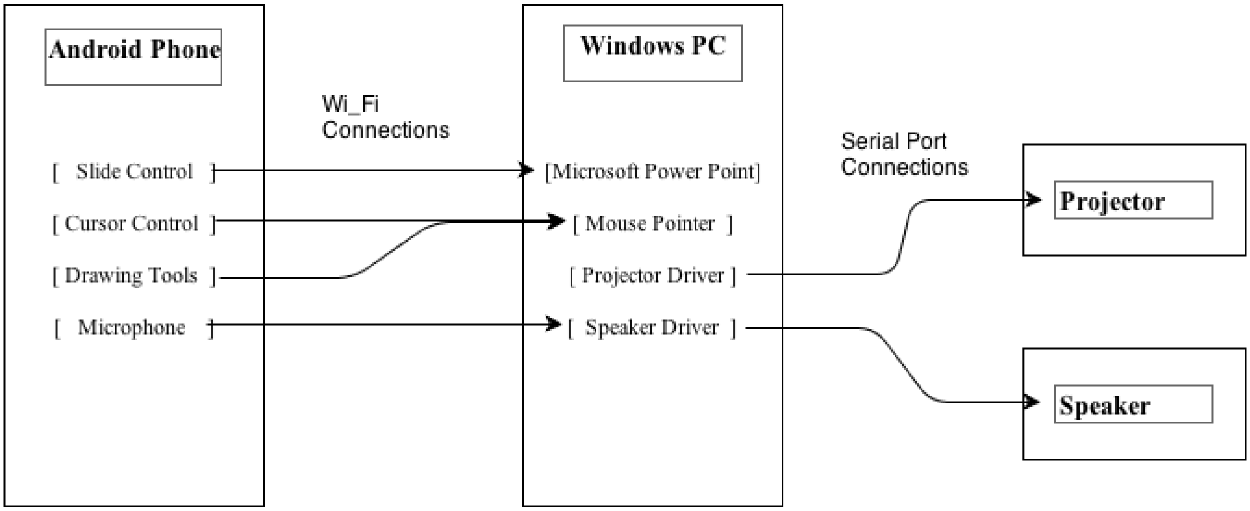


Figure 3.3 Architecture Design

The architecture consist of both machine, the smart phone and the computer. The architecture shows how the class in the application software interact with the computer during the presentation.

The computer is installed with all the software such as Microsoft PowerPoint, java and others required software driver. The proposed system enable the Android phone to remotely control the computer setting, and use the function provided.

3.2 Unified Modeling Language

3.2.1 Use Case Diagram

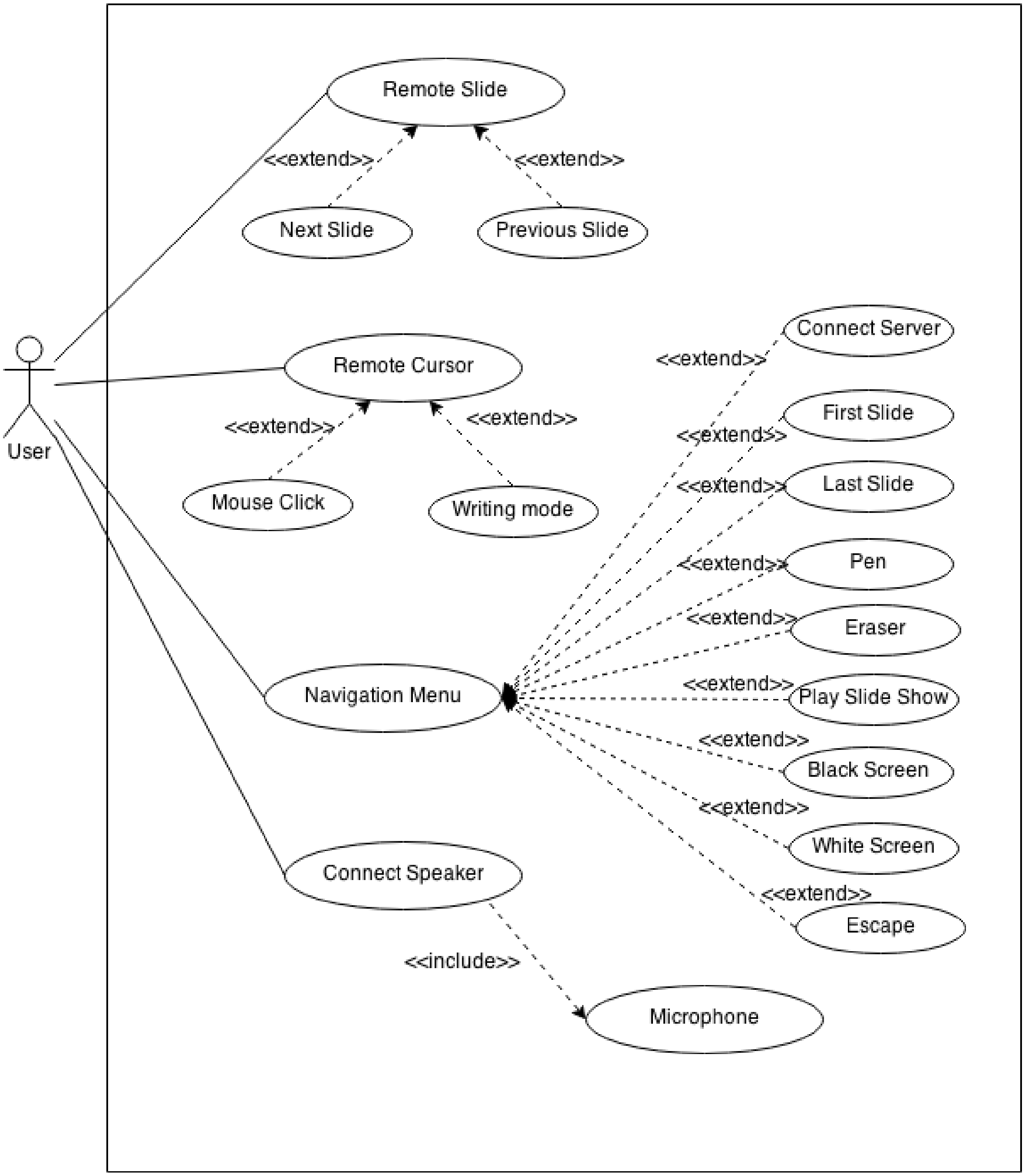


Figure 3.4 Use Case Diagram

3.2.2 Use Case Description

Remote slide use case description

Use case Name : Remote Slide	ID: 01	Importance Level : High
Primary Actor: User (Presenter)	Use case Type: Detail, Essential	
Stakeholders and interests: Audience – requests to play previous or next slides. Presenter – wants to show previous or next slide.		
Brief Description : This use case describes how presenter change the slide remotely		
Trigger: Presenter require to change the slides.		
Type : External		
Relationships : Association : Include : Extend: Next slide, previous slide. Generalization :		
Normal flow of Events : 1. The presenter wanted to change the slide. 2. The presenter touch the arrow buttons on the phone to play next or previous slide. 3. The phone sends command to server computer. 4. The computer changes the slides. 5. The projector display the new slide.		
Sub-Flows: Not applicable		
Alternate/Exceptional Flows: Not Applicable		

Remote cursor use case description

<table>
<tr><td colspan="2">Use case Name : Remote Cursor</td><td>ID: 02</td><td>Importance Level : High</td></tr>
<tr><td>Primary Actor: Presenter</td><td colspan="3">Use case Type: Detail, Essential</td></tr>
<tr><td colspan="4">Stakeholders and interests:
Audience – requests to further explain the slides visually.
Presenter – wants to show explain visually to the audience, such as writing equation, drafting a graph, run hyperlink, open external file.</td></tr>
<tr><td colspan="4">Brief Description :
This use case describes how presenter write or draw on the slide to provide visual support to the explanations.</td></tr>
<tr><td colspan="4">Trigger: Presenter require to provide extra visual explanation.</td></tr>
<tr><td colspan="4">Type : External</td></tr>
<tr><td colspan="4">Relationships :
Association :
Include :
Extend : Mouse click, Writing mode
Generalization :</td></tr>
<tr><td colspan="4">Normal flow of Events :
1. The presenter wanted to provide extra visual explanation.
2. The presenter switch the pointer to writing tools from navigation menu (refer ID:03 – Navigation Menu)
3. Presenter write or draw on the phone.
4. The phone send information to the server to process written information on the phone.
5. The computer show the result on the slide.
6. Projector display the slides.</td></tr>
<tr><td colspan="4">Sub-Flows: Not applicable</td></tr>
<tr><td colspan="4">Alternate/Exceptional Flows:
a) controlling the cursor on the computer</td></tr>
</table>

1.1 Presenter uses the pointer to remotely run hyperlink in the slides.

b) controlling the cursor on the computer

1.1 Presenter uses the pointer to remotely execute a third party programs or files as extra information for audiences.

a) Writing mode on the phone

2.1 Presenter switch on the writing mode.

2.2 The phone detect the drawing on the phone as a writing mode (not as a cursor)

b) Tools : pen

2.1 Presenter switch the pointer to pen.

2.2 Pen tools will collect information on the phone screen and send information as a drawing or writing to the server.

c) Tools : eraser

2.1 Presenter switch the pointer to eraser.

2.2 Eraser tools will erase the writing information by pen on the slides.

Navigation menu use case description

Use case Name : Navigation Menu	ID: 03	Importance Level : Medium
Primary Actor: Presenter	Use case Type: Detail, Essential	
Stakeholders and interests: Presenter – wants to execute hotkey for the slides controls, which include navigating the writing tools, slide control, screen swap.		
Brief Description : This use case describes how presenter navigating the slides tools and function via smart phone. (Navigation on works on MS PowerPoint)		
Trigger: Presenter require to perform certain action during presentation such as drawing or writing, swap slides, or navigate to other pages of slide.		
Type : External		
Relationships : Association : Include : Extend: Connect server, first slide, last slide, pen, eraser, play slide show, black screen, white screen, escape. Generalization :		
Normal flow of Events : 1. The presenter wanted to perform certain task in presentation. 2. Presenter select navigation option. 3. Phone sends instructions to the server to execute.		
Sub-Flows: Not applicable		
Alternate/Exceptional Flows: 2.1 Presenter choose connect to server 2.2 Presenter enter IP address of the server		

Connect speaker use case description

Use case Name : Connect Speaker	ID: 04	Importance Level : High
Primary Actor: Presenter	Use case Type: Detail, Essential	
Stakeholders and interests: Audience – requests presenter to use microphone Presenter – wants to use phone as a microphone.		
Brief Description : This use case describes how presenter can louder his or her volume only by using smart phone.		
Trigger: Presenter wants to louder the speaking volume.		
Type : External		
Relationships : Association : Include : Microphone Extend : Generalization :		
Normal flow of Events : 1. The presenter wanted to louder his or her speech volume. 2. Presenter connect the smart phone to server. 3. Presenter talks directly in the smart phone, and the audio is collect and transfer to the server. 4. Server plays the speech to the speaker connected to the computer.		
Sub-Flows: Not applicable		
Alternate/Exceptional Flows: Not Applicable		

3.2.3 Activity Diagram

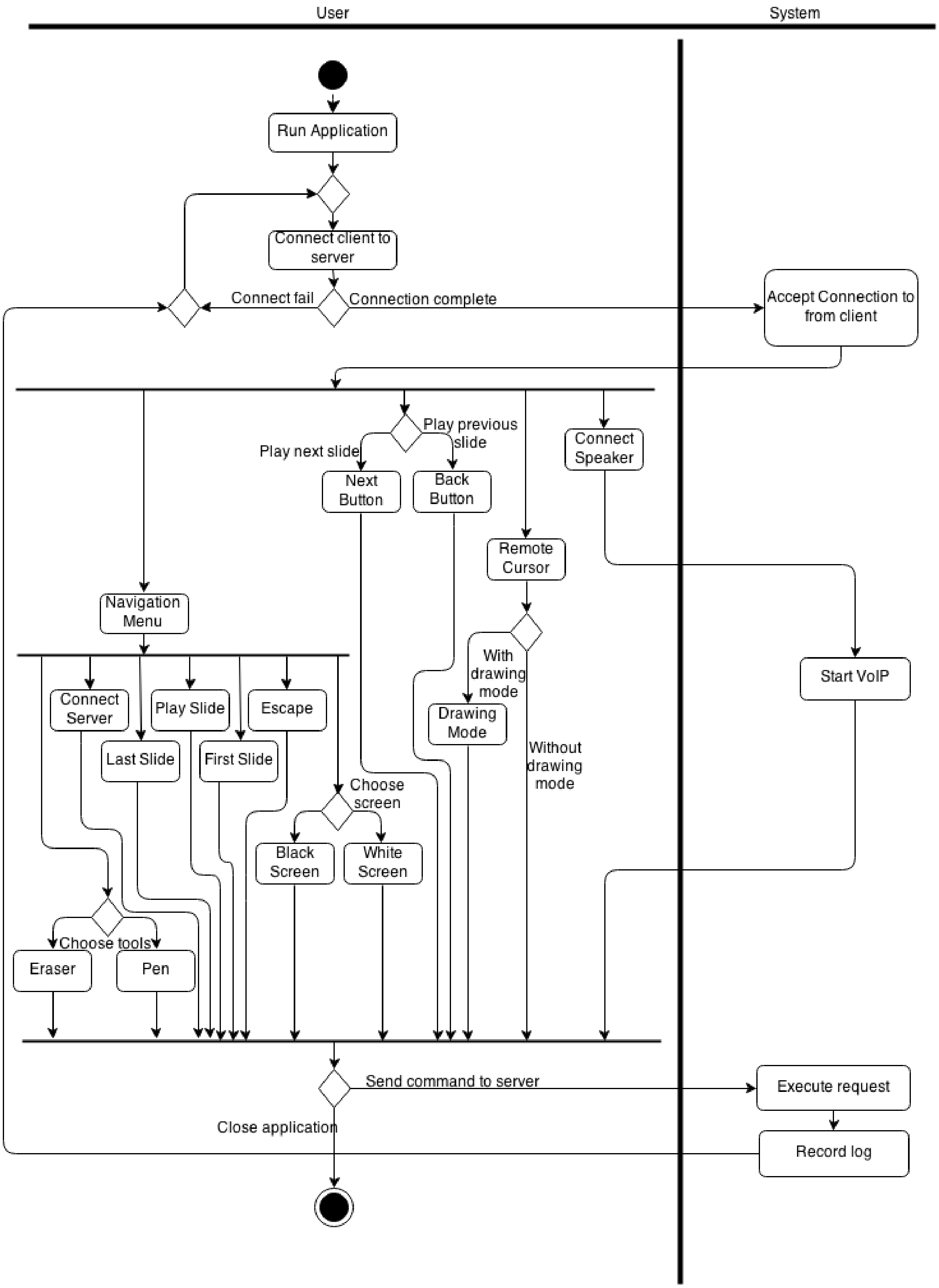


Figure 3.5 Activity Diagram

3.2.4 Structural Diagram

Class Diagram

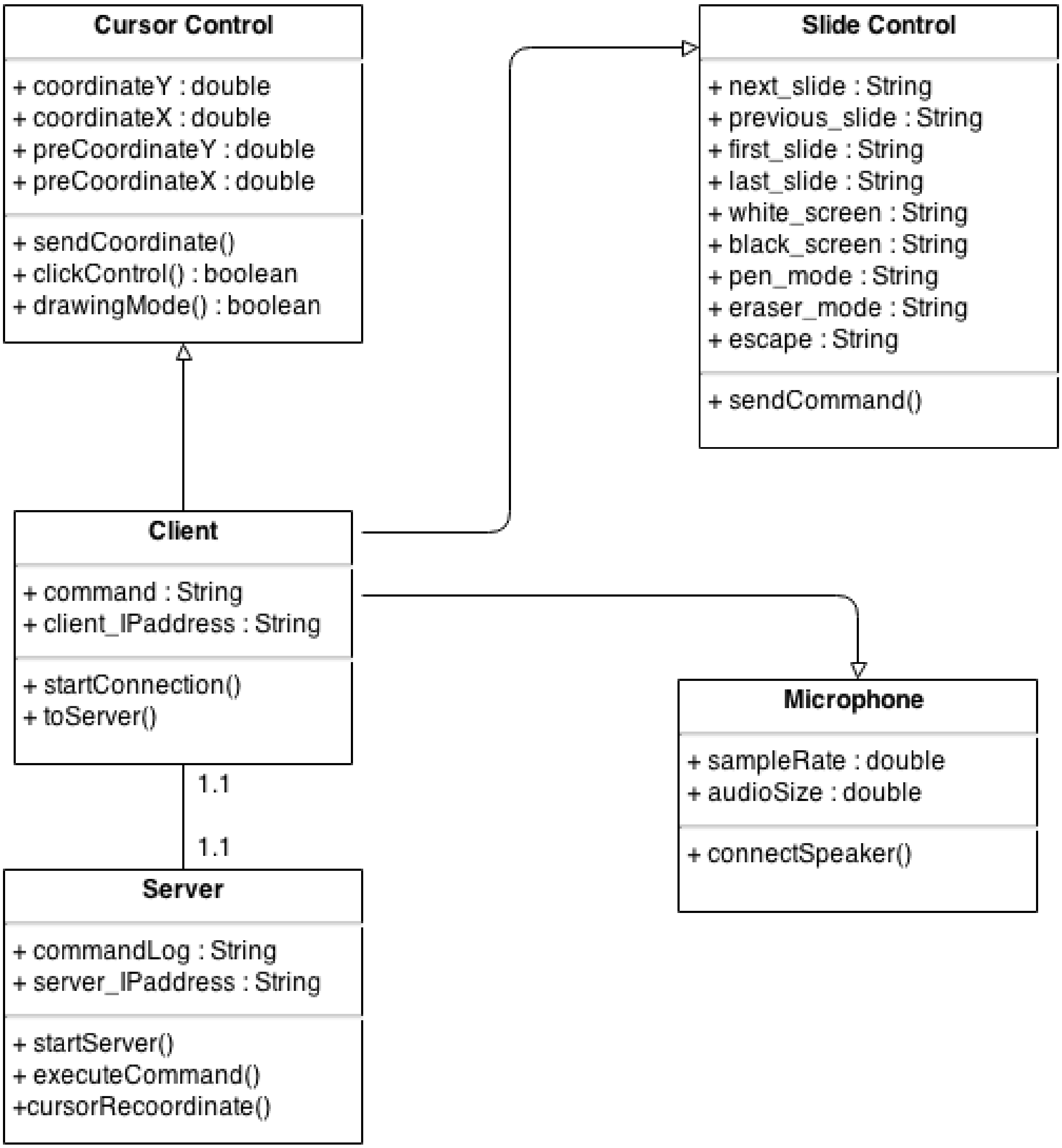


Figure 3.6 Class Diagram

3.2.5 Behavioural Diagram
State Machine Diagram

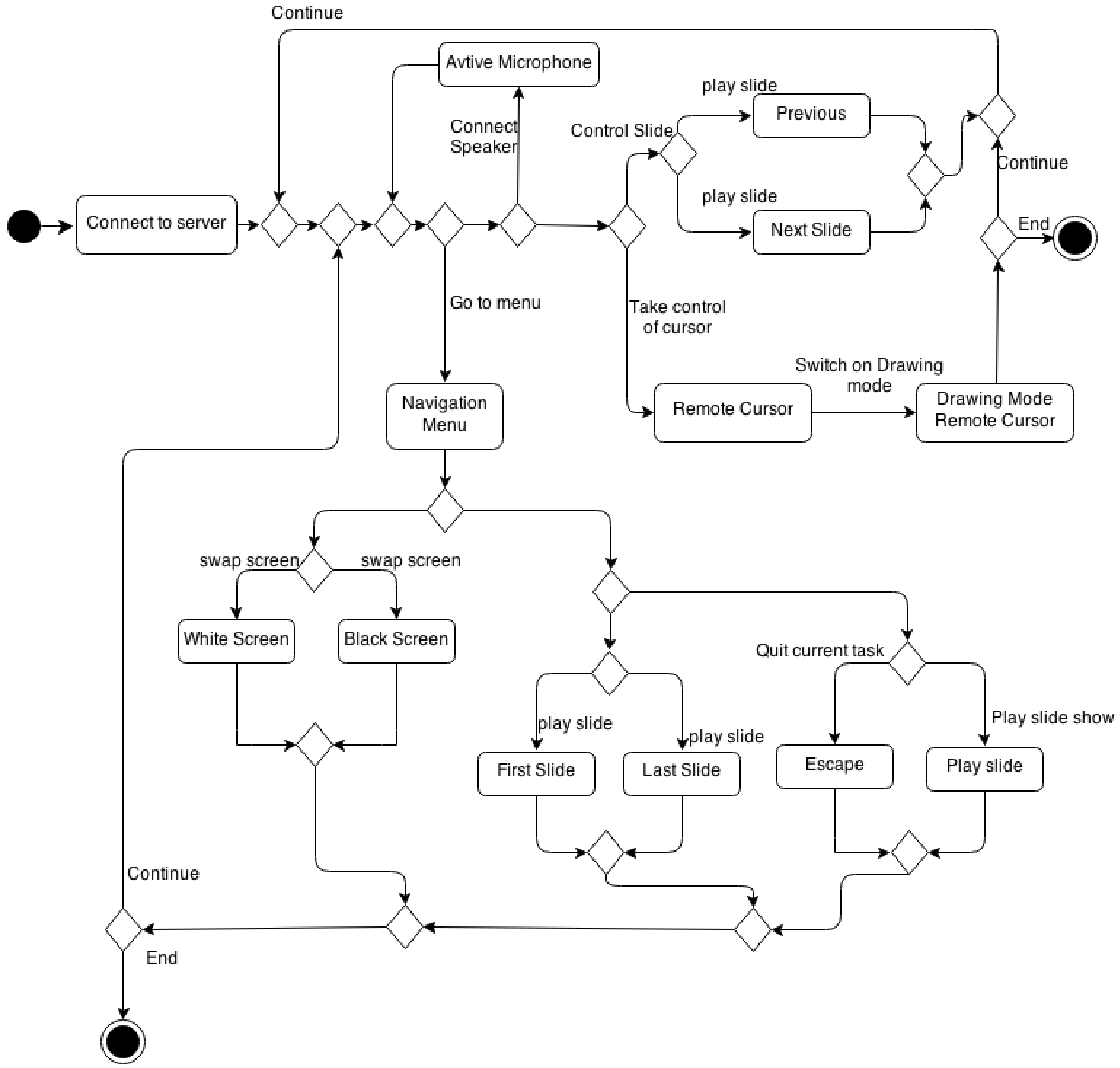


Figure 3.7 State Machine Diagram

Communication Diagram

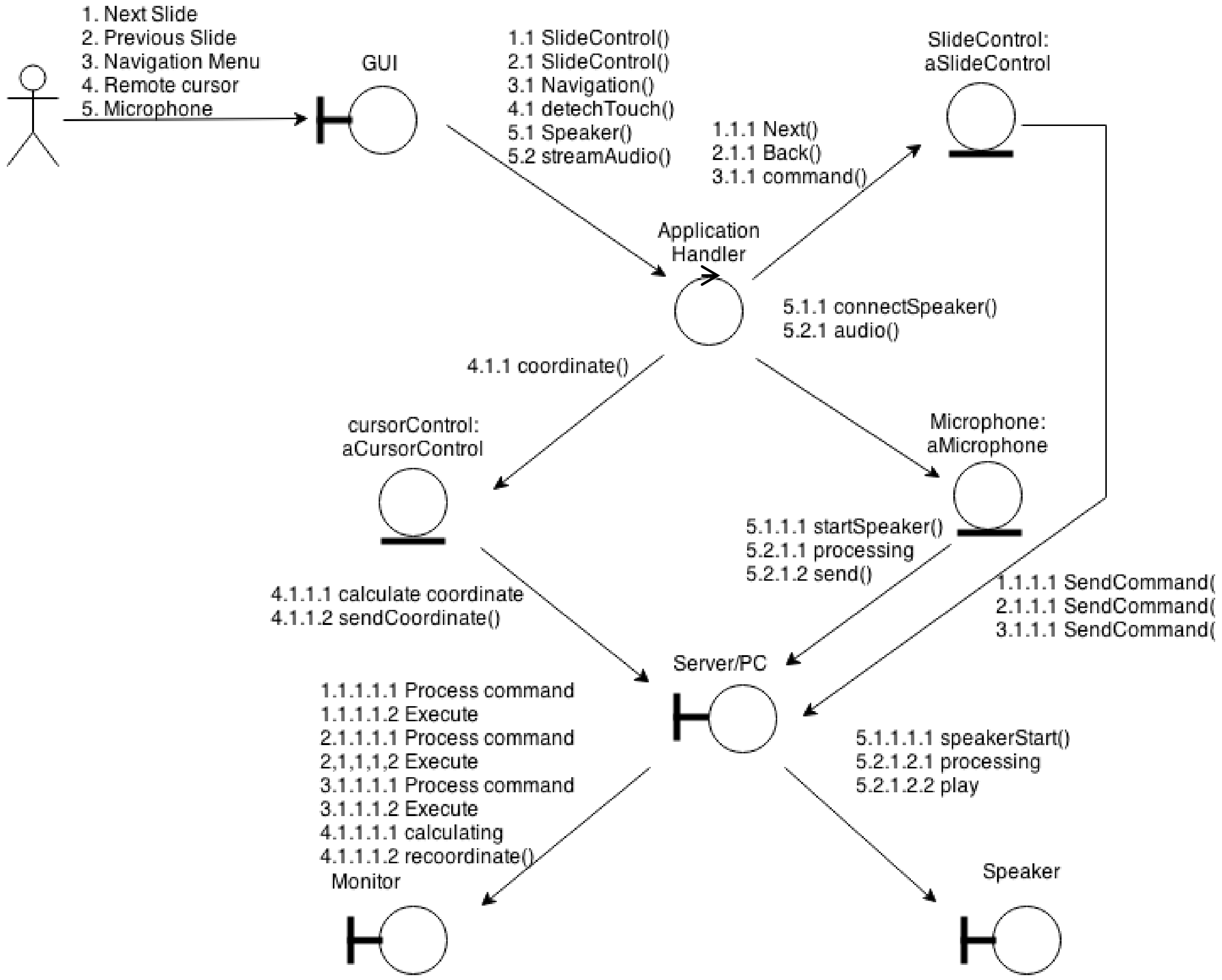


Figure 3.8 Communication Diagram

Sequence Diagram

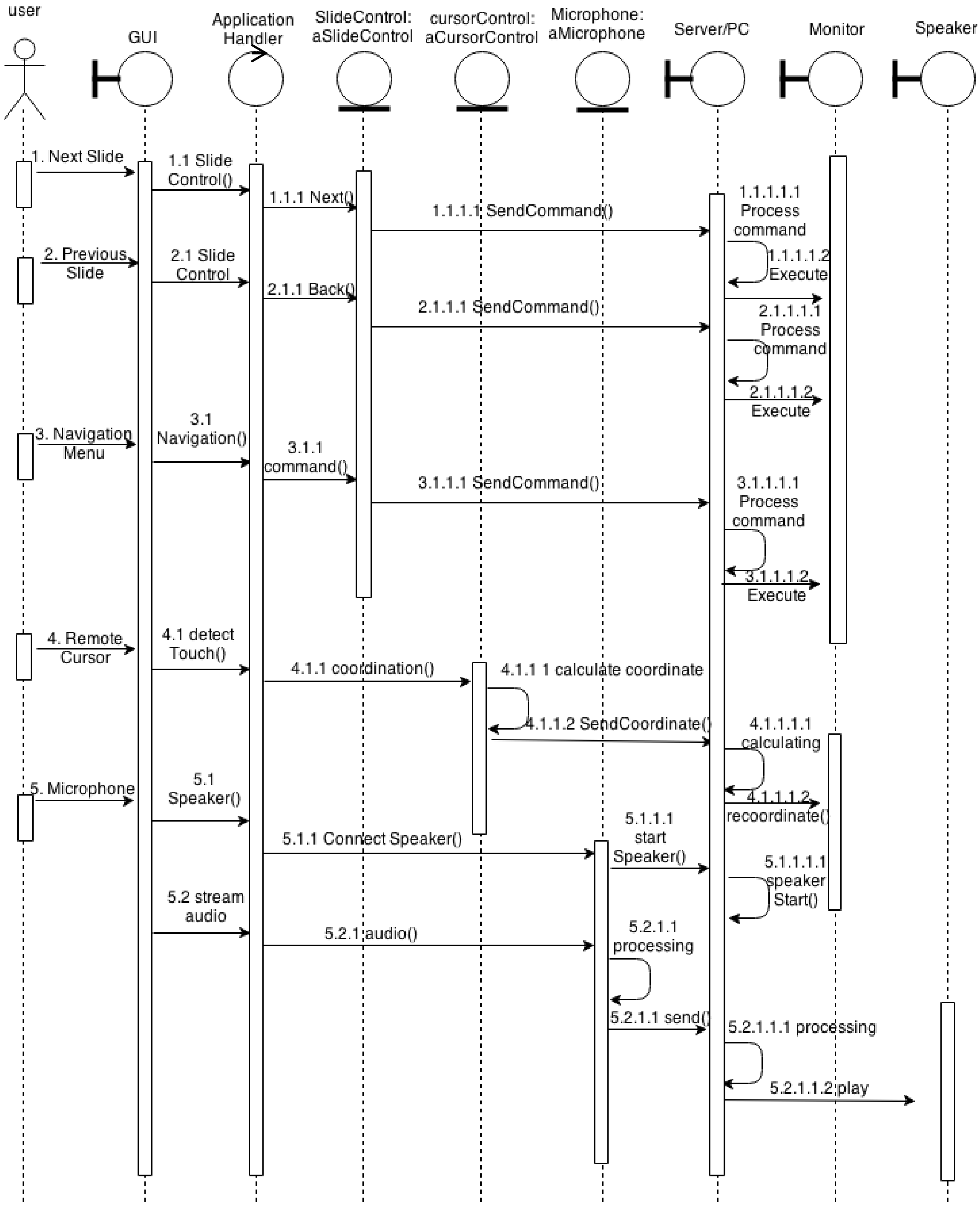


Figure 3.9 Sequence Diagram

3.3 Timeline

3.3.1 Gantt chart

Figure 3.10 show the Gantt chart for the proposed system, each period consist of two weeks.

ID	Activity	Period	1	2	3	4	5	6	7	8	9	10	11	12
1	Gather Equipment	2	■	■										
2	Define Outline	3		■	■	■								
3	Assign Requirements	2			■	■								
4	Design System	2			■	■								
5	Coding	6				■	■	■	■	■	■			
6	Testing	6					■	■	■	■	■	■		
7	Integration	6						■	■	■	■	■	■	
8	Compilation	2										■	■	
9	Documentation	1											■	■

Figure 3.10

*The incremental stage is in period 5 to11, with the Activity ID 5, 6, 7.

3.3.2 Planning

The first stage of the planning is to define the outline of the project, and what impact of the project. After the outline being defined, objectives are develop and problem statements are formed. Previous work are being reviewed to justify any solution being developed.

Next, user requirement is to be stated down, with priority. The priority is used to define which functionality is to develop first. The system architecture is build, to define the structure, behaviour, and the overview of the system.

The system start module by module according to the priority stated in user requirement, as in incremental development. The each module system the treated as a prototype, and tested individually. After all module is developed, only the validating of the system as a whole will be done. The system is test on the working environment with the test case prepared.

The documentation report on application software is produced after the system being developed.

3.4 Requirement Specification

3.4.1 User Requirement

Table 3.1 show the user requirements for the proposed application software. The priority of the requirement is sorted from highest to lowest ascending.

ID Number	Requirement
01	The system should allow the user to remotely control the slides.
02	The system should allow the user to remotely control the computer cursor.
03	The system should allow the user to write or draw on the mobile.
04	The system should allow the user to use the mobile as a microphone.

Table 3.1

3.4.2 System Requirement

Table 3.2 show the system requirement according to the user requirement from table 3.1.

User Requirement ID Number	Description
01	The system should allow the user to play next slide. The system should allow the user to play previous slide. The system should allow the user to play first slide. The system should allow the user to play last slide. The system should allow the user to view slide in thumbnail.
02	The system should allow the user to move computer cursor on mobile screen. The system should allow the user to use mouse click on mobile.
03	The system should allow the user to write on mobile screen. The system should allow the user to use drawing tools, pen and eraser. The system should allow the user to swap screen, black and white screen.
04	The system should allow the user to speak on the mobile. The system should play the user’s speech on computer’s speaker.

Table 3.2

3.4.3Verification Plan

Table 3.3 show the design and verification plan for the proposed application software.

Requirement ID	Verification
01	Able to remotely control slides
02	Able to remotely control the computer cursor.
03	Able to write or draw on the mobile.
04	Able to use the mobile as a microphone.

Table 3.3

Test case

Test Case 1

Name of the Module : Connection				
ID	**Scenario**	**Expected Result**	**Actual Result**	**Pass / Fail**
C1	Enter correct IP address	Connected to server.	Connected to server.	Pass
C2	Enter wrong IP address	Not connected, ask for reconnection.	Not connected, manually reconnect.	Pass
C3	Disconnected, and connect back to server.	Connected to server.	Connected to server.	Pass
C4	Scan the correct QR code.	Connected to server.	Connected to server.	Pass
C5	Scan the wrong QR code.	Not connected	Not connected	Pass

Table 3.4

Test Case 2

Name of the Module : Remote Slide				
ID	**Scenario**	**Expected Result**	**Actual Result**	**Pass / Fail**
RS1	Press → button	Computer play next slide.	Computer play next slide.	Pass
RS2	Press ← button	Computer play previous slide.	Computer play previous slide.	Pass
RS3	Press ↑ button	Computer play previous slide.	Computer play previous slide.	Pass
RS4	Press ↓ button	Computer play next slide.	Computer play next slide.	Pass
RS5	Play first slide	Computer play first slide.	Computer play first slide.	Pass
RS6	Play last slide	Computer play last slide.	Computer play last slide.	Pass
RS7	Switch to white screen	Computer turn on white screen.	Computer turn on white screen.	Pass
RS8	Switch to black screen	Computer turn on black screen.	Computer turn on black screen.	Pass
RS9	Escape function	Back to previous ongoing activity.	Back to previous ongoing activity.	Pass

Table 3.5

Test Case 3

Name of the Module : Remote Cursor				
ID	**Scenario**	**Expected Result**	**Actual Result**	**Pass / Fail**
RC1	Touch and hold from left to right	Cursor move from left to right	Cursor move from left to right	Pass
RC2	Touch and hold from right to left	Cursor move from right to left	Cursor move from right to left	Pass

RC3	Touch and hold from top to bottom	Cursor move from top to bottom	Cursor move from top to bottom	Pass
RC4	Touch and hold from bottom to top	Cursor move from bottom to top	Cursor move from bottom to top	Pass
RC5	Touch and hold and circle clockwise	Cursor move clockwise	Cursor move clockwise	Pass
RC6	Touch and hold and circle anti clockwise	Cursor move anti clockwise	Cursor move anti clockwise	Pass

Table 3.6

Test Case 4

Name of the Module : Drawing				
ID	**Scenario**	**Expected Result**	**Actual Result**	**Pass / Fail**
D1	Check writing mode	User is able to write anything on the mobile, and shown in the slide.	User is able to write anything on the mobile, and shown in the slide.	Pass
D2	Uncheck writing mode	User control the cursor as from a touch pad.	User control the cursor as from a touch pad.	Pass
D3	Write “TESTING” on the phone screen with writing mode.	“TESTING” shown on the slide.	“TESTING” shown on the slide.	Pass
D4	Write “TESTING” on the phone screen with writing mode, and erase “ING”.	“TEST” shown on the slide.	“TEST” shown on the slide.	Pass

Table 3.7

Test Case 5

Name of the Module : Microphone				
ID	**Scenario**	**Expected Result**	**Actual Result**	**Pass / Fail**
M1	Connect speaker	Computer speaker is able to stream speech	Computer speaker is able to stream speech	Pass
M2	Disconnect speaker	Computer speaker is disconnected from phone	Computer speaker is disconnected from phone	Pass

Table 3.8

Test Case 6

Name of the Module : Module integration				
ID	**Scenario**	**Expected Result**	**Actual Result**	**Pass / Fail**
TC1	Check drawing mode, play next slide, play previous slide.	Play next slide, then previous slide.	Play next slide, then previous slide.	Pass
TC2	Uncheck drawing mode, play next slide, play previous slide.	Play next slide, then previous slide.	Play next slide, then previous slide.	Pass
TC3	Check drawing mode, connect speaker, play next slide, play previous slide.	Speaker play presenter speech, at the same time play next slide, then previous slide.	Speaker play presenter speech, at the same time play next slide, then previous slide.	Pass

TC4	Check drawing mode, connect speaker, uncheck drawing mode, play next slide, play previous slide.	Speaker play presenter speech, at the same time play next slide, then previous slide.	Speaker play presenter speech, at the same time play next slide, then previous slide.	Pass
TC5	Check drawing mode, connect speaker, and draw circle on the slide.	Speaker play presenter speech, circle shown in the slide.	Speaker play presenter speech, circle shown in the slide.	Pass
TC6	Check drawing mode, connect speaker, change black screen, draw circle on the black screen, and return black screen to slide show, play next slide.	Speaker play presenter speech, screen change to black screen, and show a circle, then black to slide show, and continue to next slide.	Speaker play presenter speech, screen change to black screen, and show a circle, then black to slide show, and continue to next slide.	Pass
TC7	Press back on the first slide	Nothing happen.	Nothing happen.	Pass
TC8	Press next on the last slide	Nothing happen.	Nothing happen.	Pass
TC9	Press connect server while connection is still on	Nothing happen.	Nothing happen.	Pass

Table 3.9

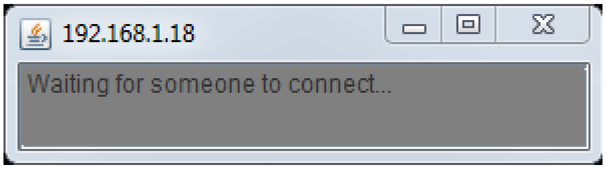
Chapter 4

4.1 System Manual

System requirements:

1. Microsoft PowerPoint installed on server side.
2. Java installed in server side. (Java Version 7)
3. Smart phone is running Android.
4. PresenterPartner.apk is installed in the smart phone.
5. Speaker and monitor is enabled and turned on.
6. Smart phone and computer is connected to the same local network.

Application software UI for server computer.

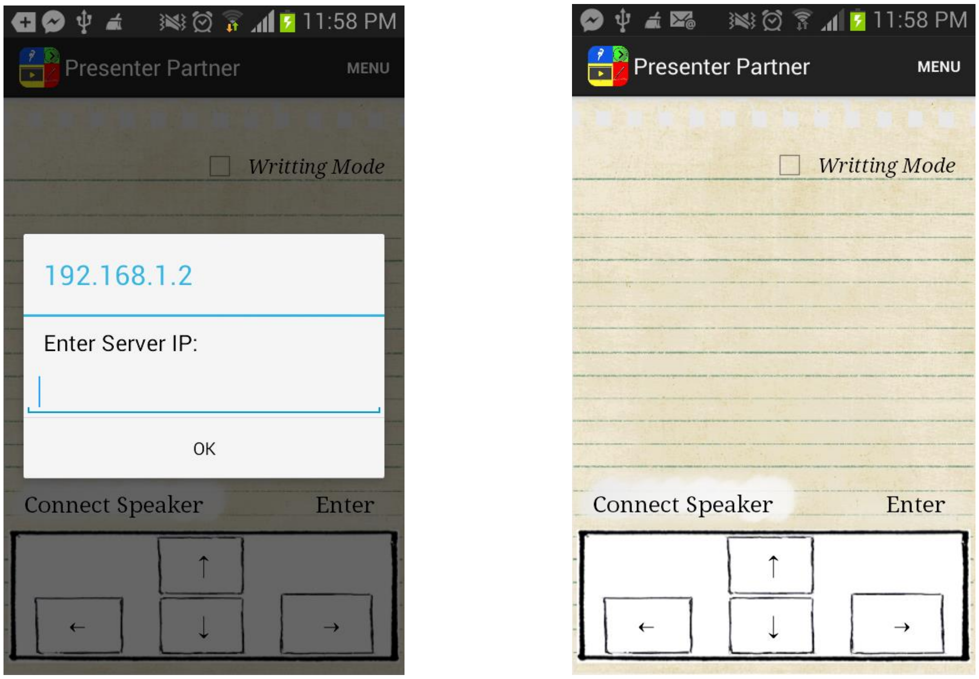


192.168.1.18

Waiting for someone to connect...

Figure 4.1 Server UI

Application software UI for Android Smartphone (client)



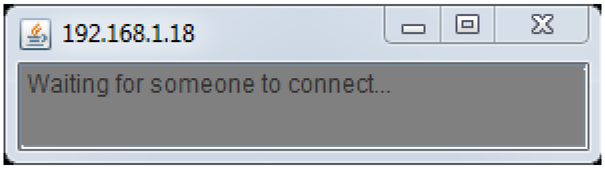
192.168.1.2

Enter Server IP:

Figure 4.2 Client UI

System setup:

1. Run PresenterPartner_server.jar on server computer. The IP address of the server will be encode into a QR code.

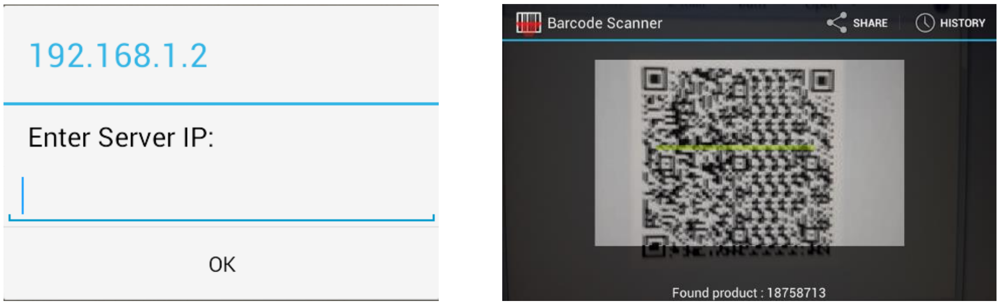


192.168.1.18

Waiting for someone to connect...

Figure 4.3 Server start up

2. While server is waiting for connection, execute PresenterPartner in the smart phone. Figure 4.4 shows the two alternative ways of connecting the smart phones with server.



192.168.1.2

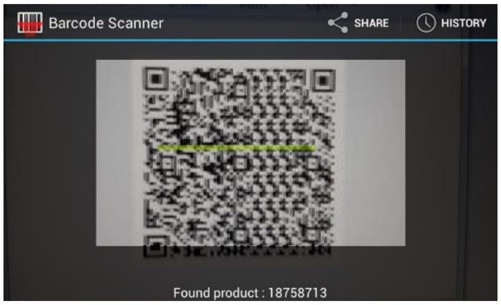
Enter Server IP:

OK

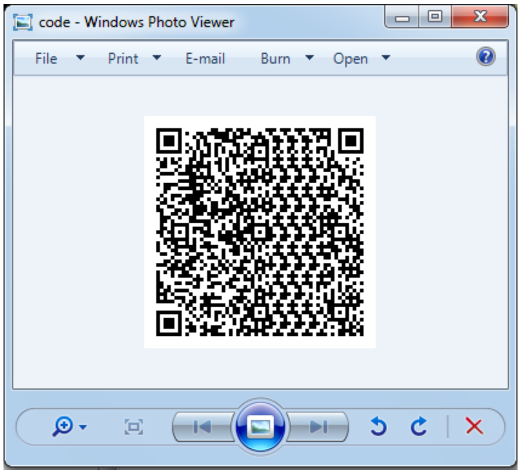
Found product : 18758713

Figure 4.4 Client start up

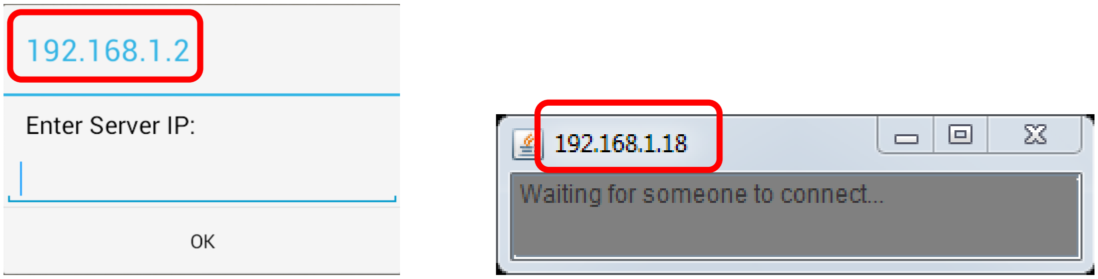
3. Smart phone will execute a QR decoder. The user is required either to scan the QR code, and the smart phone will get the IP address of sever and attempt the connection, or user can manually type in the IP address shown in the system log windows. The smart phone local IP is shown in the message, and the server IP is shown in the window title bar.



Found product : 18758713



code - Windows Photo Viewer



192.168.1.2

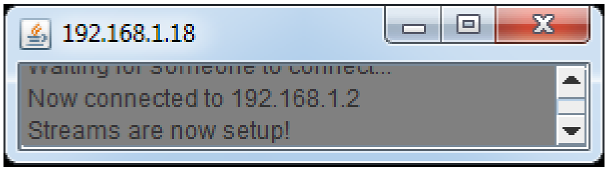
Enter Server IP:

192.168.1.18

```
Waiting for someone to connect...
```

Figure 4.5 Client and server IP addresses

4. After the user enter the correct server's IP address, or scan the correct QR code, the server will prompt connection successful message in server log, and the system is now ready to use.



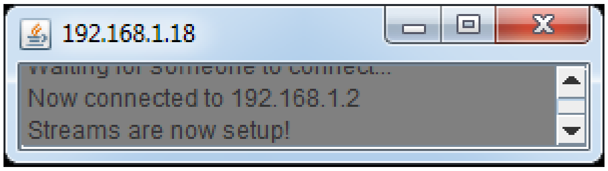
192.168.1.18

```
Waiting for someone to connect...
Now connected to 192.168.1.2
Streams are now setup!
```

Figure 4.6 Server log

System Functionality:

1. System log.



192.168.1.18

```
Waiting for someone to connect...
Now connected to 192.168.1.2
Streams are now setup!
```

Figure 4.7 Server Log

System log store all the command sent from the client side. User can check back the log for reference purpose.

2. Control buttons.

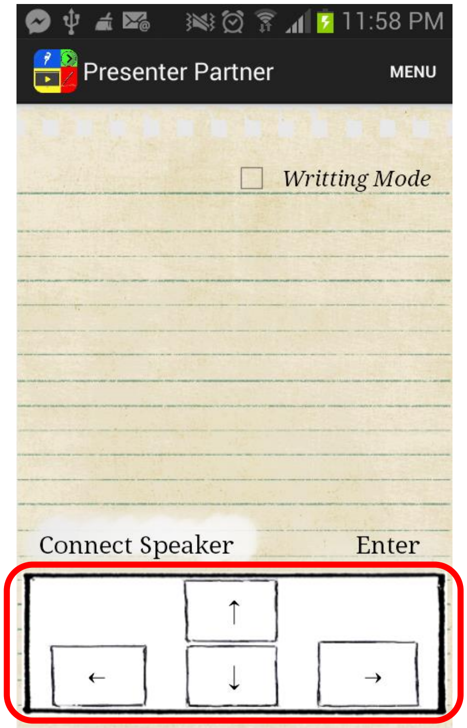


Figure 4.8 Back and next button

All the buttons are used to trigger the function of playing the next action of the slide or the previous animation of the slide. Either next animation or next slide. The back plays back the previous action of the slide show, either previous slide or previous animation.

3. Connect speaker button.

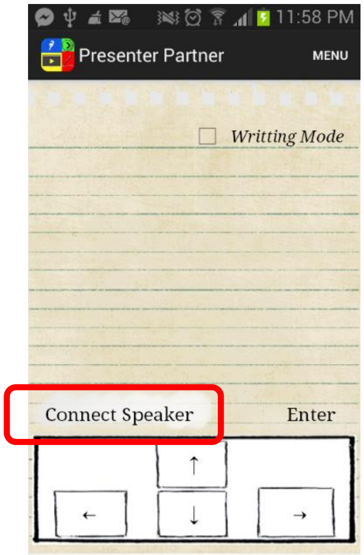


Figure 4.9 Connect speaker

Connect speaker button trigger the function of microphone. The function will active the audio input of the phone, and start collecting audio (the user speech), and send it to the server. The server will then play the audio through the speaker.

4. Drawing area, or remote cursor area.

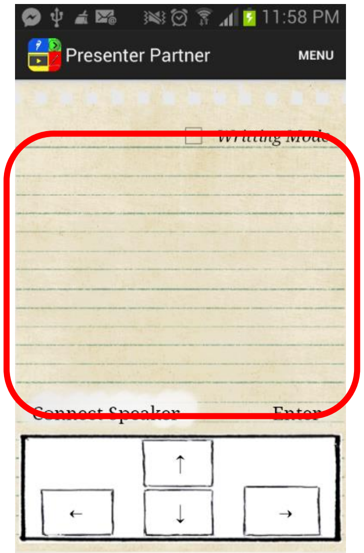


Figure 4.10 Remote area

The spaces provided is where the user can move around with the fingers, and the coordinate will be sends to the server and relocate the cursor position, or known as remote cursor.

5. Writing mode check box.

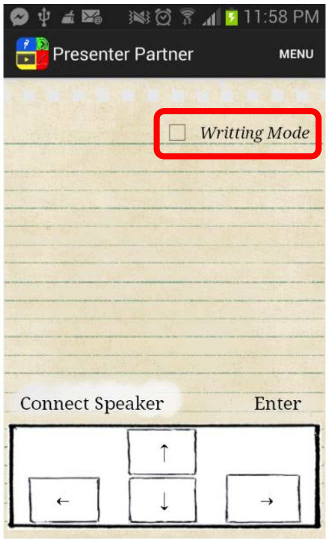


Figure 4.11 Writing mode and click

Writing mode function is for the user to active the writing mode, where the user can simply write anything on the remote cursor area (Function 4), and the writing will display on the slide, without readjusting the cursor position like using computer mouse.

To trigger the mouse left click functionality, tap two times on the remote cursor area. Tap two times to click, tap two times and hold to hold the click. For double click, tap four times.

6. Enter button

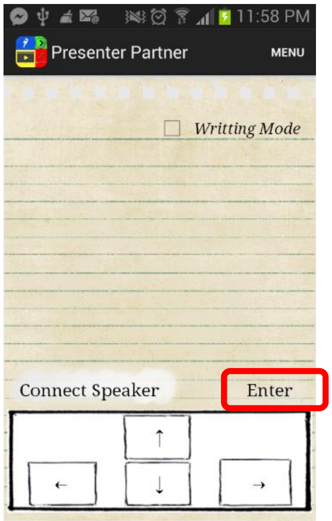


Figure 4.12 Writing mode and click

Triggers the function of Enter function from keyboard.

7. Simple navigation menu.

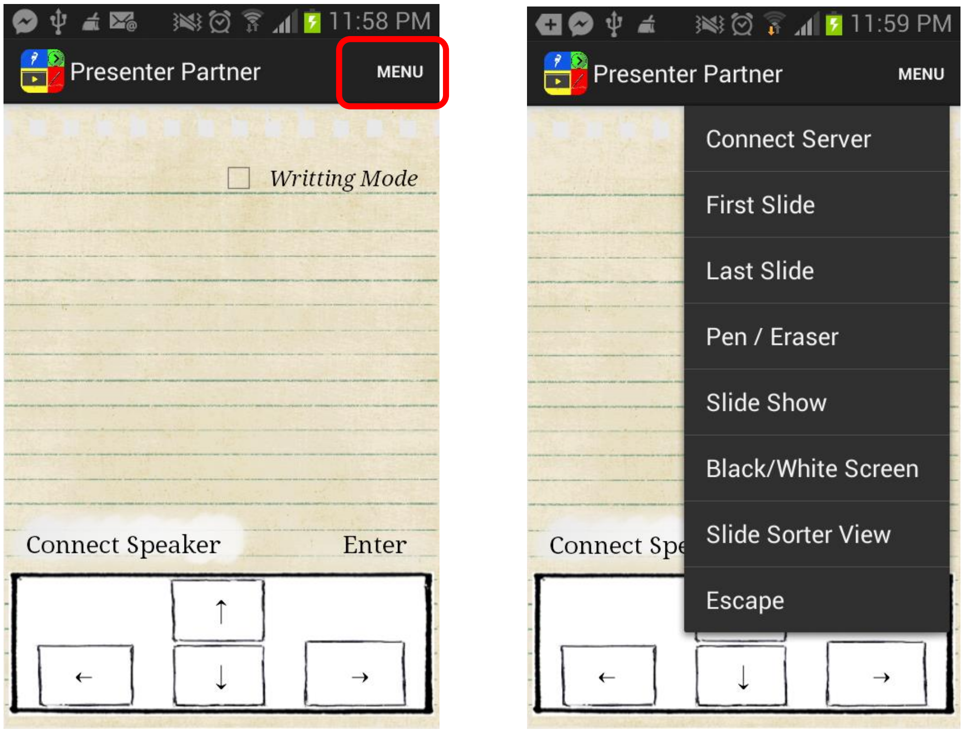


Figure 4.13 Navigation Menu

The menu provide several shortcut key to control with the slides, and some drawing tools.

i. Connection Server: It will prompt user to enter server IP address.

ii. First Slide: Jump to first slide of the slide show.

iii. Last Slide: Jump to last slide of the slide show.

iv. Pen / Eraser: Swap between pen tools and eraser tools.

v. Slide Show: Play slide full screen.

vi. Black/White Screen: Swap a black screen or a white screen into the slide shows.

vii. Slide Sorter View: View the slide show in thumbnails form.

viii. Escape: Back to previous stage.

4.2 System Review

1. QR code

Server	Client

……
……
String ipAddress = "";
try {
InetAddress addr =
InetAddress.getLocalHost();
ipAddress =
addr.getHostAddress();
} catch (UnknownHostException e) {
}
setTitle(""+ipAddress);
setVisible(true);
QRCode barcode = new QRCode();
barcode.setData(ipAddress);
 	
barcode.setDataMode(QRCode.M_AUTO);
barcode.setVersion(10);
barcode.setEcl(QRCode.ECL_M);
barcode.setProcessTilde(false);
 	
barcode.setUom(IBarcode.UOM_PIXEL);
barcode.setX(3f);
barcode.setLeftMargin(10f);
barcode.setRightMargin(10f);
barcode.setTopMargin(10f);
barcode.setBottomMargin(10f);
barcode.setResolution(72);
try {
 	
barcode.drawBarcode("code.gif");
} catch (Exception e1) {
// TODO Auto-generated
catch block
e1.printStackTrace();
}
……
……
	IntentIntegrator integrator = new IntentIntegrator(this);
	integrator.initiateScan();
	…..
	…..
	public void onActivityResult(int requestCode, int
	resultCode, Intent intent) {
	final IntentResult scanResult =
	IntentIntegrator.parseActivityResult(requestCode,
	resultCode, intent);
	if (scanResult != null) {
	m_objThreadClient = new Thread(new
	Runnable(){
	public void
	run(){
	try{
	
	clientSocket = new
	Socket(scanResult.getContents(),6789);
	
	hostIP = scanResult.getContents();
	oos = new
	ObjectOutputStream(clientSocket.getOutputStream());
	Message serverMessage = Message.obtain();
	
	ObjectInputStream ois = new
	ObjectInputStream(clientSocket.getInputStream());
	
	String strMessage = (String)ois.readObject();
	serverMsg = strMessage;
	serverMessage.obj=strMessage;
	mHandler.sendMessage(serverMessage);
	}catch(Exception e){
	e.printStackTrace();
	}
	}});
	
	Toast.makeText(this, scanResult.toString(),
	Toast.LENGTH_LONG).show();
	
	m_objThreadClient.start();
	}
	}
	…….
	……

The server's code segments show how server generate a QR code image using QRCode library. The IP address of the server is then encode into the QR code.

Client's segment shows how the client initiate the intent using zxing library. Zxing is a QR code decoder software. The library decode the QR code from the server, and will store the content (server's IP address) into a string, and start connection to the server.

2. Starting Connection

Server	Client

……
public void startRunning(){
try{
server = new ServerSocket(6789, 100);
while(true){
try{
waitForConnection();
setupStreams();
whileChatting();
}catch(EOFException eofException){
 	
showMessage("\n Server ended the connection! ");
}finally{
closeCrap();
}
}
}catch(IOException ioException){
 	
ioException.printStackTrace();
}
}
……
……
 	
private void waitForConnection() throws IOException{
showMessage(" Waiting for someone to connect... \n");
connection = server.accept();
showMessage(" Now connected to "+connection.getInetAddress().getHostName());
}
……
……
 	
private void setupStreams() throws IOException{
output = new ObjectOutputStream(connection.getOutputStream());
output.flush();
input = new ObjectInputStream(connection.getInputStream());
showMessage("\n Streams are now setup!");
}
……
……
 	
private void whileChatting() throws IOException{
…….
…….//process command area
}
…….
……
	……
	public void Start(View view){
	m_objThreadClient = new Thread(new Runnable(){
	public void run(){
	try{
	
	clientSocket = new Socket(ipAddress,6789);
	oos = new ObjectOutputStream(clientSocket.getOutputStream());
	
	Message serverMessage = Message.obtain();
	
	ObjectInputStream ois = new ObjectInputStream(clientSocket.getInputStream());
	
	String strMessage = (String)ois.readObject();
	serverMessage.obj=strMessage;
	oos.writeObject("Connected");
	
	mHandler.sendMessage(serverMessage);
	
	}catch(Exception e){
	e.printStackTrace();
	}
	}});
	m_objThreadClient.start();
	}
	……
	…...

The server's code segments show how the server start up, and wait for connection from client.

Client's code segments show how the program take in server's IP input either by QR decoder or manually entered by user, then start the connection to the server.

3. Remote Button

Server

```
……
……
private void whileChatting() throws IOException{
……
do{
try{
            message = (String) input.readObject();

if(message.equals("CLIENT - NEXT")){
           try {
                 Robot r=new Robot();
                 r.keyPress(KeyEvent.VK_DOWN);
                 r.keyRelease(KeyEvent.VK_DOWN);
            }catch (AWTException ae) {
                    //ae.printStackTrace();
}}

else if(message.equals("CLIENT - BACK")){
                 try {
                    Robot r=new Robot();
                    r.keyPress(KeyEvent.VK_UP);
                    r.keyRelease(KeyEvent.VK_UP);
                 } catch (AWTException ae) {
                    //ae.printStackTrace();
}}

else if(message.equals("CLIENT - LEFT")){
                 try {
                    Robot r=new Robot();
                    r.keyPress(KeyEvent.VK_LEFT);
                    r.keyRelease(KeyEvent.VK_LEFT);
                 } catch (AWTException ae) {
                    //ae.printStackTrace();
}}

else if(message.equals("CLIENT - RIGHT")){
                 try {
                    Robot r=new Robot();
                    r.keyPress(KeyEvent.VK_RIGHT);
                    r.keyRelease(KeyEvent.VK_RIGHT);
                 } catch (AWTException ae) {
                    //ae.printStackTrace();
}}

else if(message.equals("CLIENT - ENTER")){
                 try {
                    Robot r=new Robot();
                    r.keyPress(KeyEvent.VK_ENTER);
                    r.keyRelease(KeyEvent.VK_ENTER);
                 } catch (AWTException ae) {
                    //ae.printStackTrace();
}}
……
……
}catch(ClassNotFoundException
classNotFoundException){
   showMessage("\n I don't know what the user sent! ");
}
```

Client

```
……
……
public void back(View view){
                          Button question = (Button)
findViewById(R.id.btn_back);
       question.setOnClickListener(new
OnClickListener(){

       public void onClick(View v){
            try {
                 oos.writeObject("CLIENT - BACK");
            } catch (IOException e) {
                  e.printStackTrace();
            }
       }
       });
}

public void next(View view){
                          Button question = (Button)
findViewById(R.id.btn_next);
       question.setOnClickListener(new
OnClickListener(){

       public void onClick(View v){
            try {
                  oos.writeObject("CLIENT - NEXT");
            } catch (IOException e) {
                  e.printStackTrace();
            }
       }
       });
}

public void left(View view){
                          Button question = (Button)
findViewById(R.id.btn_left);
       question.setOnClickListener(new
OnClickListener(){

       public void onClick(View v){
            try {
                  oos.writeObject("CLIENT - LEFT");
            } catch (IOException e) {
                  e.printStackTrace();
            }
       }
       });
}

public void right(View view){
                          Button question = (Button)
findViewById(R.id.btn_right);
       question.setOnClickListener(new
OnClickListener(){

       public void onClick(View v){
            try {
}while(!message.equals("CLIENT - END"));
}
```

```
                    oos.writeObject("CLIENT - RIGHT");
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        });
}

public void enter(View view){
                        Button question = (Button)
findViewById(R.id.btn_enter);
        question.setOnClickListener(new
OnClickListener(){

        public void onClick(View v){
                try {
                    oos.writeObject("CLIENT - ENTER");
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        });
}
……
……
```

The server's code segments show how the server process the command sent from the client side.

The five main function of controlling the slide, which include the four arrows button in the keyboard, and most often used button, enter.

Table 4.1 shows the five different command shown in the above code segments:

Command sent	Server process
i. CLIENT – BACK	Keyboard non-number pad up arrow key input
ii. CLIENT – NEXT	Keyboard non-number pad down arrow key input
iii. CLIENT – LEFT	Keyboard non-number pad left arrow key input
iv. CLIENT – RIGHT	Keyboard non-number pad right arrow key input
v. CLIENT – ENTER	Keyboard enter input

Table 4.1

4. Remote Cursor

Server

```
……
……
private void whileChatting() throws IOException{
……
do{
try{
            message = (String) input.readObject();
…….
…….
else{
int posY = 0;
int posX = 0;
String position = message;
if (position.contains(":")) {
            String [] parts = position.split(":");
            String part1 = parts[0];
            String part2 = parts[1];
            posX = Integer.parseInt(part1);
            posY = Integer.parseInt(part2);
}else{
throw new IllegalArgumentException("String " + position
+ " does not contain :");
}
try{
            Robot r = new Robot();
            x=x+posX;y=y+posY;
            r.mouseMove(x, y);
            repaint();
} catch (AWTException ae) {
            //ae.printStackTrace();
}}
catch(ClassNotFoundException
classNotFoundException){
   showMessage("\n I don't know what the user sent! ");
}
}while(!message.equals("CLIENT - END"));
}
……
……
```

Client

```
int count=0,count2=0;
double curX=0, preX=0, curY=0, preY=0,
x1=0,y1=0,x3,y3;
double curX2,curY2;
boolean firstTouch = false;
long time;
…….
…….
@Override
public boolean onTouchEvent(MotionEvent event) {
            ……..
            …….
preX = event.getX();
preY = event.getY();

if(event.getAction() == MotionEvent.ACTION_MOVE){
            final int historySize = event.getHistorySize();
            final int pointerCount =
event.getPointerCount();
            double posX, posY;
            for (int h = 0; h < historySize; h++) {
                for (int p = 0; p < pointerCount; p++) {
                        curX = event.getX();
                        curY = event.getY();
                        posX = event.getHistoricalX(p,h);
                        posY = event.getHistoricalY(p,h);
                        x1 = curX-posX;
                        y1 = curY-posY;
                        try {
                             int x2 = (int) x1;
                             int y2 = (int) y1;
                             oos.writeObject(""+x2+":"+y2);
                        }catch (IOException e) {
                        e.printStackTrace();
                        }
            }}
}
if(event.getAction()== MotionEvent.ACTION_UP){
            try {
            oos.writeObject("CLIENT - Unclick");
            } catch (IOException e) {
            e.printStackTrace();
            }
            }
}
…….
…….
```

The server’s code segments show how the server break down the coordinates sent by client into X-axis and Y-axis, and store into two different integers. These two integers will be used to relocate the cursor. The result will be received in every few milliseconds, hence the relocation of the cursor happen very often (every time the new result arrive).

The client's code segments show how the client detect the hands movement on the phone screen, by X-axis and Y-axis. The coordinates is then calculate by using current coordinate minus the previous coordinate to get the distant between the two points (previous and current coordinates). The calculation happen in every few milliseconds, and the result is send to the server, to relocate the cursor of the computer.

The frequency of how many times the calculation takes place indicates how smooth will the remote cursor be.

While the server receive the result (+/- distance), the server will get the current cursor position and add in the latest result to move the cursor (+/- X-axis and Y-axis).

5. Clicking Function

Server	Client

……
……
private void whileChatting() throws IOException{
……
do{
try{
message = (String) input.readObject();
…….
…….
else if(message.equals("CLIENT - Click")){
try {
Robot r=new Robot();
r.mousePress(InputEvent.BUTTON1_MASK);
} catch (AWTException ae) {
//ae.printStackTrace();
}}
 	
else if(message.equals("CLIENT - Unclick")){
try {
Robot r=new Robot();
r.mouseRelease(InputEvent.BUTTON1_MASK);
} catch (AWTException ae) {
//ae.printStackTrace();
}}
…….
…….
}catch(ClassNotFoundException
classNotFoundException){
showMessage("\n I don't know what the user sent! ");
}
}while(!message.equals("CLIENT - END"));
}
……
……
	@Override
	public boolean onTouchEvent(MotionEvent event) {
	while(true){
	…….
	…….
	if(event.getAction() == MotionEvent.ACTION_DOWN) {
	if(firstTouch && (System.currentTimeMillis() -
	time) <= 300) {
	try {
	oos.writeObject("CLIENT - Click");
	} catch (IOException e) {
	e.printStackTrace();
	}
	firstTouch = false;
	} else {
	firstTouch = true;
	time = System.currentTimeMillis();
	return false;
	}
	…….
	…….
	if(event.getAction()== MotionEvent.ACTION_UP){
	try {
	oos.writeObject("CLIENT - Unclick");
	} catch (IOException e) {
	e.printStackTrace();
	}
	}
	}
	……
	…….

The code segments show how the click function works.

The client first touch on the screen will initiate a time counter, and if the user touches another time in the screen within 0.3 second, it will initiate click command.

Table 4.2 shows the two different command shown in the above code segments:

Command sent	Server process
i. CLIENT – Click	Hold mouse left click
ii. CLIENT – Unclick	Release mouse left click

Table 4.2

6. Writing mode

Server

```
……
……
private void whileChatting() throws IOException{
……
do{
try{
            message = (String) input.readObject();
…….
…….
else{
int posY = 0;
int posX = 0;
String position = message;
if (position.contains(":")) {
            String [] parts = position.split(":");
            String part1 = parts[0];
            String part2 = parts[1];
            posX = Integer.parseInt(part1);
            posY = Integer.parseInt(part2);
}else{
throw new IllegalArgumentException("String " + position
+ " does not contain :");
}
try{
            Robot r = new Robot();
            x=x+posX;y=y+posY;
            r.mouseMove(x, y);
            repaint();
} catch (AWTException ae) {
            //ae.printStackTrace();
}}
catch(ClassNotFoundException
classNotFoundException){
   showMessage("\n I don't know what the user sent! ");
}
}while(!message.equals("CLIENT - END"));
}
…….
……
```

Client

```
@Override
public boolean onTouchEvent(MotionEvent event) {
…….
…….
if(event.getAction() == MotionEvent.ACTION_MOVE){
            final int historySize = event.getHistorySize();
            final int pointerCount =
event.getPointerCount();
            double posX, posY;
            for (int h = 0; h < historySize; h++) {
                for (int p = 0; p < pointerCount; p++) {
                        curX = event.getX();
                        curY = event.getY();
                        posX = event.getHistoricalX(p,h);
                        posY = event.getHistoricalY(p,h);
                        x1 = curX-posX;
                        y1 = curY-posY;
            try {
                int x2 = (int) x1;
                int y2 = (int) y1;
                oos.writeObject(""+x2+":"+y2);
            } catch (IOException e) {
                        e.printStackTrace();
            }}
      }
}
……
……
```

The information sent is same as remote cursor function (refer 4. Remote Cursor). But the calculation is different.

When the user release the touch on the phone, the calculation of distant does not reset back to zero, but continues. This allow the program able to count the distant on the user's next following touch.

This formula allow the user to directly write on the phone like writing on the paper, there is no worries of adjusting the cursor to the correct position before start writing.

7. Speaker

<table>
<tr><th>Server</th><th>Client</th></tr>
<tr><td>

```
……
static int sampleRate = 32000;
static int dataSize = 2024;
……
public static void main(String args[]) throws Exception {
            Server speakerServer = new Server();
            speakerServer.setDefaultCloseOperation
(JFrame.EXIT_ON_CLOSE);


   DatagramSocket serverSocket = new
DatagramSocket(port);

   /**
    * Formula for lag = (byte_size/sample_rate)*2
    * Byte size 9728 will produce ~ 0.45 seconds of lag.
Voice slightly broken.
    * Byte size 1400 will produce ~ 0.06 seconds of lag.
Voice extremely broken.
    * Byte size 4000 will produce ~ 0.18 seconds of lag.
Voice slightly more broken then 9728.
    */

   byte[] receiveData = new byte[dataSize];

   format = new AudioFormat(sampleRate, 16, 1, true,
false);
   dataLineInfo = new
DataLine.Info(SourceDataLine.class, format);
   sourceDataLine = (SourceDataLine)
AudioSystem.getLine(dataLineInfo);
   sourceDataLine.open(format);
   sourceDataLine.start();

   FloatControl volumeControl = (FloatControl)
sourceDataLine.getControl
   (FloatControl.Type.MASTER_GAIN);
   volumeControl.setValue(1.00f);

   DatagramPacket receivePacket = new
DatagramPacket(receiveData,
          receiveData.length);
   ByteArrayInputStream baiss = new
ByteArrayInputStream(
          receivePacket.getData());
   while (status == true) {
      serverSocket.receive(receivePacket);
      ais = new AudioInputStream(baiss, format,
receivePacket.getLength());
      toSpeaker(receivePacket.getData());
   }
   sourceDataLine.drain();
   sourceDataLine.close();
}

public static void toSpeaker(byte soundbytes[]) {
   try {
```

</td><td>

```
private int sampleRate = 32000;
private int channelConfig =
AudioFormat.CHANNEL_IN_MONO;
private int audioFormat =
AudioFormat.ENCODING_PCM_16BIT;
int minBufSize =
AudioRecord.getMinBufferSize(sampleRate,
channelConfig, audioFormat);
private boolean status = true;
……
…...
public void startStreaming() {
              streamThread = new Thread(new
Runnable() {

            @Override
            public void run() {
            try {
            DatagramSocket socket = new
DatagramSocket();
            Log.d("VS", "Socket Created");

            byte[] buffer = new byte[minBufSize];

            Log.d("VS","Buffer created of size " +
minBufSize);
            DatagramPacket packet;

            final InetAddress destination =
InetAddress.getByName(hostIP);
            Log.d("VS", "Address retrieved");

            recorder = new
AudioRecord(MediaRecorder.AudioSource.
MIC,sampleRate,channelConfig,audioFormat,
minBufSize*10);
            Log.d("VS", "Recorder initialized");

            recorder.startRecording();

             while(status == true) {
                   //reading data from MIC into buffer
                   minBufSize = recorder.read(buffer, 0,
buffer.length);

                   //putting buffer in the packet
                   packet = new DatagramPacket
(buffer,buffer.length,destination,port);

                   socket.send(packet);
                   System.out.println("MinBufferSize: "
+minBufSize);
                    }
            } catch(UnknownHostException e) {
            Log.e("VS", "UnknownHostException");
             } catch (IOException e) {
                          e.printStackTrace();
                          Log.e("VS", "IOException");
```

</td></tr>
</table>

```
        sourceDataLine.write(soundbytes, 0,
soundbytes.length);
    } catch (Exception e) {
        System.out.println("Not working in speakers...");
        e.printStackTrace();
    }
}}
```

```
                }
              }
          });
          streamThread.start();
}
…….
…….
```

The code segments show how client start connect to the server's speaker server. The phone will start collecting voice into a datagram, and send to server to process and play it on speaker.

The specification of the UDP:

i. Sample rate: 32000
ii. Datagram size: 2024 Bytes
iii. Audio channel: Mono
iv. Encoding format: 16 bit

8. Navigation Menu

Server	Client

……
……
private void whileChatting() throws IOException{
……
do{
try{
message = (String) input.readObject();
……
……
else if(message.equals("CLIENT - First Slide")){
try {
Robot r=new Robot();
r.keyPress(KeyEvent.VK_HOME);
r.keyRelease(KeyEvent.VK_HOME);
} catch (AWTException ae) {
//ae.printStackTrace();
}}
 	
else if(message.equals("CLIENT - Last Slide")){
try {
Robot r=new Robot();
r.keyPress(KeyEvent.VK_END);
r.keyRelease(KeyEvent.VK_END);
} catch (AWTException ae) {
//ae.printStackTrace();
}}
 	
else if(message.equals("CLIENT - Pen")){
try {
Robot r=new Robot();
r.keyPress(KeyEvent.VK_CONTROL);
r.keyPress(KeyEvent.VK_P);
r.keyRelease(KeyEvent.VK_CONTROL);
r.keyRelease(KeyEvent.VK_P);
} catch (AWTException ae) {
//ae.printStackTrace();
}}
 	
else if(message.equals("CLIENT - Eraser")){
try {
Robot r=new Robot();
r.keyPress(KeyEvent.VK_CONTROL);
r.keyPress(KeyEvent.VK_E);
r.keyRelease(KeyEvent.VK_CONTROL);
r.keyRelease(KeyEvent.VK_E);
} catch (AWTException ae) {
//ae.printStackTrace();
}}
 	
else if(message.equals("CLIENT - Play Slide")){
try {
Robot r=new Robot();
r.keyPress(KeyEvent.VK_F5);
r.keyRelease(KeyEvent.VK_F5);
} catch (AWTException ae) {
//ae.printStackTrace();
}}
	……
	……
	@Override
	public boolean onOptionsItemSelected(MenuItem item) {
	// Handle item selection
	switch (item.getItemId()) {
	case R.id.connect_server:
	final EditText serverIP2 = new EditText(this);
	
	AlertDialog alertDialog = new AlertDialog.Builder(this).create();
	
	alertDialog.setTitle(Utils.getIPAddress(true));
	alertDialog.setMessage("Enter Server IP:");
	alertDialog.setView(serverIP2);
	
	alertDialog.setButton("OK", new DialogInterface.OnClickListener() {
	
	public void onClick(DialogInterface dialog, int which) {
	final String serverIPinput = serverIP2.getEditableText().toString();
	
	m_objThreadClient = new Thread(new Runnable(){
	public void run(){
	try{
	clientSocket = new Socket(serverIPinput,6789);
	hostIP = serverIPinput;
	oos = new ObjectOutputStream(clientSocket.getOutputStream());
	
	Message serverMessage = Message.obtain();
	
	ObjectInputStream ois = new ObjectInputStream(clientSocket.getInputStream());
	
	String strMessage = (String)ois.readObject();
	serverMsg = strMessage;
	serverMessage.obj=strMessage;
	mHandler.sendMessage(serverMessage);
	
	}catch(Exception e){
	e.printStackTrace();
	}
	}});
	m_objThreadClient.start();
	});
	alertDialog.show();
	return true;
	
	case R.id.first_slide:
	try {
	oos.writeObject("CLIENT - First Slide");
	} catch (IOException e) {

```
else if(message.equals("CLIENT - Black Screen")){
try {
            Robot r=new Robot();
             r.keyPress(KeyEvent.VK_B);
             r.keyRelease(KeyEvent.VK_B);
} catch (AWTException ae) {
//ae.printStackTrace();
}}

else if(message.equals("CLIENT - White Screen")){
try {
                Robot r=new Robot();
                 r.keyPress(KeyEvent.VK_W);
                 r.keyRelease(KeyEvent.VK_W);
} catch (AWTException ae) {
//ae.printStackTrace();
}}

else if(message.equals("CLIENT - Slide Sorter")){
try {
             Robot r=new Robot();
             r.keyPress(KeyEvent.VK_ALT);
             r.keyPress(KeyEvent.VK_V);
             r.keyRelease(KeyEvent.VK_ALT);
             r.keyRelease(KeyEvent.VK_V);
             r.keyPress(KeyEvent.VK_D);
             r.keyRelease(KeyEvent.VK_D);
} catch (AWTException ae) {
  //ae.printStackTrace();
}}

else if(message.equals("CLIENT - Escape")){
try {
              Robot r=new Robot();
              r.keyPress(KeyEvent.VK_ESCAPE);
              r.keyRelease(KeyEvent.VK_ESCAPE);
} catch (AWTException ae) {
//ae.printStackTrace();
}}
……
……
}catch(ClassNotFoundException
classNotFoundException){
    showMessage("\n I don't know what the user sent! ");
}
}while(!message.equals("CLIENT - END"));
}
```

```
e.printStackTrace();
}
return true;

case R.id.last_slide:
try {
oos.writeObject("CLIENT – Last Slide");
} catch (IOException e) {
e.printStackTrace();
}
return true;

case R.id.pen_eraser:
try {
        if(penCount == 0){
oos.writeObject("CLIENT - Pen");
penCount=1;
}else{
oos.writeObject("CLIENT - Eraser");
penCount=0;
}
} catch (IOException e) {
e.printStackTrace();
}
return true;

case R.id.slide_show:
try {
oos.writeObject("CLIENT – Play Slide");
} catch (IOException e) {
      e.printStackTrace();
}
return true;

case R.id.switch_focus:
try {
if(screenCounter == 0){
oos.writeObject("CLIENT - Black Screen");
screenCounter =1;
}else{
oos.writeObject("CLIENT - White Screen");
screenCounter = 0;
}} catch (IOException e) {
       e.printStackTrace();
}
return true;

case R.id.slide_sorter_view:
try {
oos.writeObject("CLIENT - Slide Sorter");
} catch (IOException e) {
      e.printStackTrace();
}
return true;

case R.id.escape:
try {
oos.writeObject("CLIENT - Escape");
} catch (IOException e) {
       e.printStackTrace();
}
return true;
```

	default:
	return super.onOptionsItemSelected(item);
	}
	}
	……
	……

The code segments show how the navigation menu is able to integrate with the tools applicable in Microsoft PowerPoints.

Table 4.3 shows the five different command shown in the above code segments:

Command sent	Server process
i. CLIENT – First Slide	Keyboard “home” button input
ii. CLIENT – Last Slide	Keyboard “end” button input
iii. CLIENT – Pen	Keyboard “ctrl” button and “P” character button input
iv. CLIENT – Eraser	Keyboard “ctrl” button and “E” character button input
v. CLIENT – Play Slide	Keyboard “F5” button input
vi. CLIENT – Black Screen	Keyboard “B” character button input
vii. CLIENT – White Screen	Keyboard “W” character button input
viii. CLIENT – Slide Sorter	Keyboard “alt” button and “V” character button, then “D” character button input
ix. CLIENT – Escape	Keyboard “esc” character button input

Table 4.3

The first option of menu “Connect Server” will initiate an intent to prompt user to enter server IP address, where the input will be used to attempt connection with the server.

Chapter 5

5.1 Future Works

The application software still can be further improve. First and foremost, the functions provided can be further improve, such as the remote cursor and the microphone functionality.

The remote cursor provided had the delay around 0.3 seconds when the finger move around the phone. The delay is caused by the algorithm which uses the current position and historical position of the finger, and send the calculation to the server the relocate the cursor. The calculation on the historical position takes time, which causes the remote cursor had delay.

As for the phone as microphone, it uses user datagram protocol (UDP), as streaming in real time is the main concern. This caused the audio (datagram) stream to the server might had slight broken, and a little delay caused by the network latency.

Other than improving what is already in hands, several module could be added to the application to make the application even more interesting. The application can make use of the sensor phone provided. For example, the application is able to use the accelerometer from android phones to switch between slides, or remotely control the cursor using the phone's sensor.

Another function which would make the presentation even more interesting is the sharing of information from the presenter to the audience. For example, there are times when the audience is unable to see the slide, or the audience want some information from the presenter, the presenter is able to use the application to take a screenshot of the slide and push the screenshot to the audience phone. This way, the audience is able to keep information wanted, and have a clearer view on the slides. The same function can also be used to share images or video from youtube for extra information for the audiences.

5.2 Conclusions

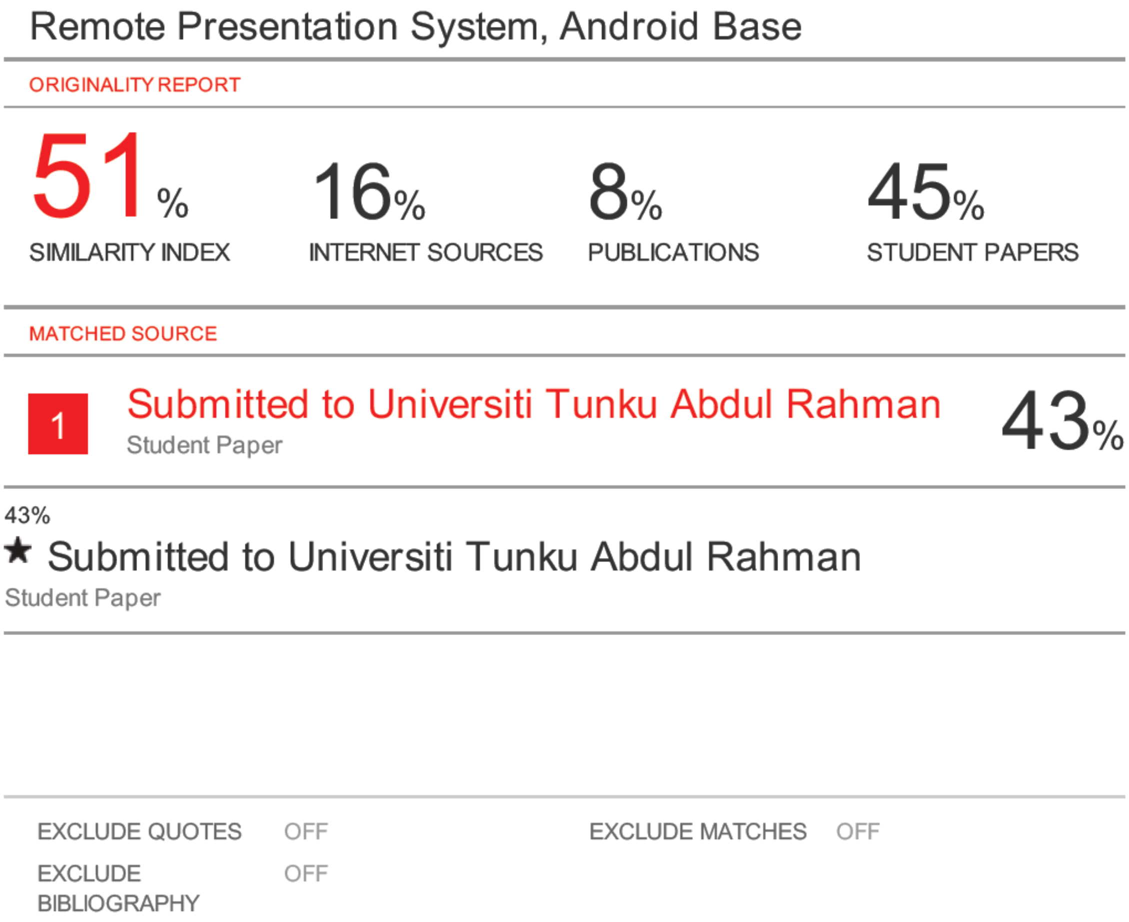
The value of a product or idea is not solely rests on the quality of it, but also how well the product or idea can be presented to the audience. There are times where the presenter is able to present well, but was limited by the environment, such the audience sitting on the corner is unable to see the slide from the projector clearly. It would be such a waste that if the presentation went wrong because of the environmental factor.

With a single gadget on hand, the mobile phone, the presenter can do so much more on the presentation. By pairing the presenter’s mobile with the computer, the presenter is able to remotely control the computer, and give a better quality of presentation on audio and visual guild. Hence providing a more convincing result on how well audience understand the presented product or idea.

The proposed system is able to make new changes to traditional presentation, bring the traditional presentation to a whole new level.

Bibliography

1. Dictionary.com (2012) *the definition of presentation*. [online] Available at: http://dictionary.reference.com/browse/presentation [Accessed: 1 June 2013].
2. Logitech.com (2001) *Logitech - Professional Presenter R800*. [online] Available at: http://www.logitech.com/en-us/for-business/products/presentation/professional-presenter-r800-business [Accessed: 15 June 2013].
3. Robabdul.com (2005) *Evolutionary Development Model, Waterfall Model and IPO Model*. [online] Available at: http://www.robabdul.com/Data-Management-System-Software-Development-Cycle.asp [Accessed: 20 June 2013].
4. App Store. 2010. *Slideshow Remote™ for PowerPoint*. [online] Available at: https://itunes.apple.com/us/app/slideshow-remote-for-powerpoint/id328556878?mt=8 [Accessed: 23 June 2013].
5. Play.google.com. 1999. *i-Clickr PowerPoint Remote - Android Apps on Google Play*. [online] Available at: https://play.google.com/store/apps/details?id=com.senstic.iclickrpaid2&feature=related_apps [Accessed: 20 June 2013].
6. Play.google.com. 1999. *PowerPoint Keynote Remote Pro - Android Apps on Google Play*. [online] Available at: https://play.google.com/store/apps/details?id=com.pptremotecontrol.android.presenter.fullversion&feature=related_apps [Accessed: 20 June 2013].
7. Play.google.com. 1999. *Unified Remote - Android Apps on Google Play*. [online] Available at: https://play.google.com/store/apps/details?id=com.Relmtech.Remote&hl=en [Accessed: 28 June 2013].
8. Shure.co.uk. n.d.. *Lavalier Microphones*. [online] Available at: http://www.shure.co.uk/products/microphones/lavalier_microphones [Accessed: 25 June 2013].



Remote Presentation System, Android Base

ORIGINALITY REPORT

51%	16%	8%	45%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

MATCHED SOURCE

1	Submitted to Universiti Tunku Abdul Rahman Student Paper	43%

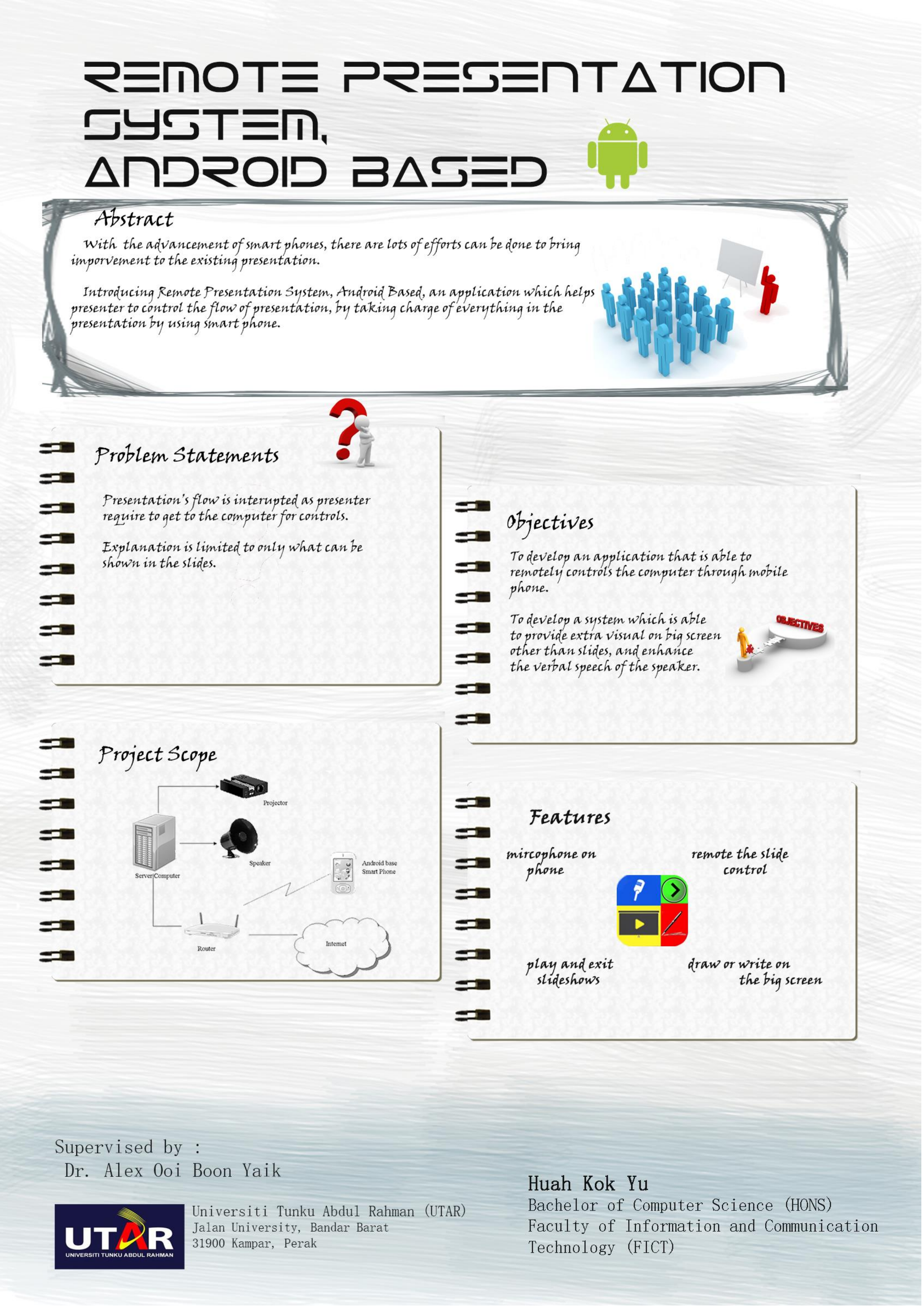
43%

★ Submitted to Universiti Tunku Abdul Rahman

Student Paper

EXCLUDE QUOTES	OFF	EXCLUDE MATCHES	OFF
EXCLUDE BIBLIOGRAPHY	OFF		

*The “Student Paper – Submitted to Universiti Tunku Abdul Rahman” is the previous work of my own.



REMOTE PRESENTATION SYSTEM, ANDROID BASED

Abstract

With the advancement of smart phones, there are lots of efforts can be done to bring imporvement to the existing presentation.

Introducing Remote Presentation System, Android Based, an application which helps presenter to control the flow of presentation, by taking charge of everything in the presentation by using smart phone.

Problem Statements

Presentation's flow is interupted as presenter require to get to the computer for controls.

Explanation is limited to only what can be shown in the slides.

Objectives

To develop an application that is able to remotely controls the computer through mobile phone.

To develop a system which is able to provide extra visual on big screen other than slides, and enhance the verbal speech of the speaker.

Project Scope

Features

mircophone on phone

remote the slide control

play and exit slideshows

draw or write on the big screen

Supervised by :
Dr. Alex Ooi Boon Yaik

Universiti Tunku Abdul Rahman (UTAR)
Jalan University, Bandar Barat
31900 Kampar, Perak

Huah Kok Yu
Bachelor of Computer Science (HONS)
Faculty of Information and Communication Technology (FICT)