

**Capston Design Project**

**PreFuture**

**Dynamic Presentation**

**Final Specification**

**담당교수 : 이석원 교수님**

**담당조교 : 고태형 조교**

**조 원 : 200620334 오희준**

**200620316 신대욱**

**200722160 박태선**

**200822166 박철룡**



**Table of Contents**

[1. Projects Overview 6](#_Toc327011341)

[1.1. Scope and Objectives 6](#_Toc327011342)

[1.1.1. Change about presentation tool - All-in-one 6](#_Toc327011343)

[1.1.2. Improvement about presentation environment - Anywhere 6](#_Toc327011344)

[1.1.3. Scope 6](#_Toc327011345)

[1.2. Supplementary Requirements 7](#_Toc327011346)

[1.2.1. Introduction 7](#_Toc327011347)

[1.2.2. Functionality 7](#_Toc327011348)

[1.2.3. Usability 7](#_Toc327011349)

[1.2.4. Human Factors 7](#_Toc327011350)

[1.2.5 Reliability 7](#_Toc327011351)

[1.2.6. Performance 8](#_Toc327011352)

[1.2.7. Supportability 8](#_Toc327011353)

[1.2.8. Implementation Constraints 8](#_Toc327011354)

[1.2.9. Interfaces 8](#_Toc327011355)

[2. Customer Requirements 8](#_Toc327011356)

[2.1. User requirements 9](#_Toc327011357)

[2.2. System requirements 10](#_Toc327011358)

[2.3. Domain requirements 11](#_Toc327011359)

[3. Requirements Analysis 11](#_Toc327011360)

[3.1. Structural Analysis 12](#_Toc327011361)

[3.1.1. Boundary 12](#_Toc327011362)

[3.1.2. Control 12](#_Toc327011363)

[3.1.3. Entity 13](#_Toc327011364)

[3.2. Deployment Model 14](#_Toc327011365)

[3.3. Behavioral Analysis 15](#_Toc327011366)

[3.4. Analysis Packages 25](#_Toc327011367)

[3.4.1. Application Specific Layer 26](#_Toc327011368)

[3.4.2. Application Generic Layer 26](#_Toc327011369)

[3.4.3. Middleware Layer 26](#_Toc327011370)

[3.4.4. Application System Software Layer 26](#_Toc327011371)

[4. Validation Criteria (metrics & measures) 26](#_Toc327011372)

[5. Use Case Realization Design 28](#_Toc327011373)

[5.1. Use Case 1 - Process Connection 28](#_Toc327011374)

[5.1.1. Use Case 1 – Realization 28](#_Toc327011375)

[5.1.2. Use Case 1 – Relationship 29](#_Toc327011376)

[5.2. Use Case 2 – Move mouse using touchpad 30](#_Toc327011377)

[5.2.1. Use Case 2 – Realization 30](#_Toc327011378)

[5.2.2. Use Case 2 – Relationship 31](#_Toc327011379)

[5.3. Use Case 3 – Click mouse using touchpad 32](#_Toc327011380)

[5.3.1. Use Case 3 – Realization 32](#_Toc327011381)

[5.3.2. Use Case 3 – Relationship 33](#_Toc327011382)

[5.4. Use Case 4 – Move mouse using motion 34](#_Toc327011383)

[5.4.1. Use Case 4 – Realization 34](#_Toc327011384)

[5.4.2. Use Case 4 – Relationship 35](#_Toc327011385)

[5.5. Use Case 5 – Control keyboard 36](#_Toc327011386)

[5.5.1. Use Case 5 – Realization 36](#_Toc327011387)

[5.5.2. Use Case 5 – Relationship 37](#_Toc327011388)

[5.6. Use Case 6 – Process record sound 38](#_Toc327011389)

[5.6.1. Use Case 6 – Realization 38](#_Toc327011390)

[5.6.2. Use Case 6 – Relationship 39](#_Toc327011391)

[5.7. Use Case 7 – Process play sound 40](#_Toc327011392)

[5.7.1. Use Case 7 – Realization 40](#_Toc327011393)

[5.7.2. Use Case 7 – Relationship 41](#_Toc327011394)

[5.8. Use Case 8 – Transfer voice to PC speaker 42](#_Toc327011395)

[5.8.1. Use Case 8 – Realization 42](#_Toc327011396)

[5.8.2. Use Case 8 – Relationship 43](#_Toc327011397)

[5.9. Use Case 9 – Transfer document 44](#_Toc327011398)

[5.9.1. Use Case 9 – Realization 44](#_Toc327011399)

[5.9.2. Use Case 9 – Relationship 45](#_Toc327011400)

[6. Subsystem Design 46](#_Toc327011401)

[6.1. Action Manager Subsystem 46](#_Toc327011402)

[6.2. Sound Action Management 47](#_Toc327011403)

[6.3. Transfer Management 48](#_Toc327011404)

[6.4. Mouse Action Manager 49](#_Toc327011405)

[6.5. Keyboard Action Management 50](#_Toc327011406)

[6.6. PC Connection Management 51](#_Toc327011407)

[6.7. Device Connection Management 52](#_Toc327011408)

[7. Human Interfaces 53](#_Toc327011409)

[7.1. Client 53](#_Toc327011410)

[7.1.1. Main 53](#_Toc327011411)

[7.1.2. Connection Popup 53](#_Toc327011412)

[7.1.3. File List 54](#_Toc327011413)

[7.1.4. File selection popup 54](#_Toc327011414)

[7.2. PC 55](#_Toc327011415)

[7.2.1. PC Status 55](#_Toc327011416)

[8. Use-Case Scenarios 56](#_Toc327011417)

[8.1. Process Connecting 56](#_Toc327011418)

[8.2. Move mouse using touchpad 56](#_Toc327011419)

[8.3. Click mouse using touchpad 56](#_Toc327011420)

[8.4. Move mouse pointer using motion 56](#_Toc327011421)

[8.5. Control keyboard 57](#_Toc327011422)

[8.6. Process Record Sound 57](#_Toc327011423)

[8.7. Process Play Sound 57](#_Toc327011424)

[8.8. Transfer voice to PC speaker 57](#_Toc327011425)

[8.9. Transfer document 58](#_Toc327011426)

[9. System/Data Dependencies & Requirements 58](#_Toc327011427)

[9.1. Software System 58](#_Toc327011428)

[9.2. Hardware System 58](#_Toc327011429)

[9.3. Data Dependencies & requirements 58](#_Toc327011430)

[10. Testing Plan & Results 59](#_Toc327011431)

[10.1 Introduction 59](#_Toc327011432)

[10.2 Goals and Objectives 59](#_Toc327011433)

[10.2.1 Goals 59](#_Toc327011434)

[10.2.2. Statement of scope 59](#_Toc327011435)

[10.3. Major constraints 60](#_Toc327011436)

[10.4 Test Plan 60](#_Toc327011437)

[10.4.1. Software to be tested 60](#_Toc327011438)

[10.4.2. Testing Strategy 60](#_Toc327011439)

[10.4.3. Testing resources and staffing 61](#_Toc327011440)

[10.5. Test metrics 61](#_Toc327011441)

[10.6. Testing tools and environment 61](#_Toc327011442)

[10.7. Test Schedule 61](#_Toc327011443)

[10.8. Unit Test 62](#_Toc327011444)

[10.8.1. Testing procedure: 62](#_Toc327011445)

[10.8.2. DeviceConnectionManager Test 62](#_Toc327011446)

[10.8.3. ConnectionActivity Test 63](#_Toc327011447)

[10.8.4. MousePointerbyTouchpadActivity Test 63](#_Toc327011448)

[10.8.5. MouseClickActivity Test 63](#_Toc327011449)

[10.8.6. MousePointerbyMotionActivity Test 63](#_Toc327011450)

[10.8.7. KeyboardActionActivity Test 63](#_Toc327011451)

[10.8.8. SoundRecordActivity Test 63](#_Toc327011452)

[10.8.9. RecordedSoundPlayActivity Test 63](#_Toc327011453)

[10.8.10. SoundTransferActivity Test 64](#_Toc327011454)

[10.8.11. DeviceSoundActionManager Test 64](#_Toc327011455)

[10.8.12. DocumentTransferActivity Test 64](#_Toc327011456)

[10.8.13. PCConnectionManager Test 64](#_Toc327011457)

[10.8.14. MouseActionManager Test 64](#_Toc327011458)

[10.8.15. KeyboardActionManager Test 65](#_Toc327011459)

[10.8.16. SoundActionManager Test 65](#_Toc327011460)

[10.8.17. TransferManager Test 65](#_Toc327011461)

[10.9. Integration / System Tests 65](#_Toc327011462)

[11. Appendices 80](#_Toc327011463)

[11.1. Project Status 80](#_Toc327011464)

**Table of Figure**

[Figure 1: Project Scope 8](#_Toc327022128)

[Figure 2: Use Case Diagram 11](#_Toc327022129)

[Figure 3: Project Structure 14](#_Toc327022130)

[Figure 4: Deployment Model 16](#_Toc327022131)

[Figure 5: Behavior Analysis 17](#_Toc327022132)

[Figure 6: Use Case 1. Connecting Process 18](#_Toc327022133)

[Figure 7: Use Case 2. Move Mouse by Touchpad 19](#_Toc327022134)

[Figure 8: Use Case 3. Click Mouse 20](#_Toc327022135)

[Figure 9: Use Case 4. Move Mouse by Motion 21](#_Toc327022136)

[Figure 10: Use Case 5. Keyboard Control 22](#_Toc327022137)

[Figure 11: Use Case 6. Record Sound 23](#_Toc327022138)

[Figure 12: Use Case 7. Play Recorded 24](#_Toc327022139)

[Figure 13: Use Case 8. Transfer Voice 25](#_Toc327022140)

[Figure 14: Use Case 8. Transfer Document 27](#_Toc327022141)

[Figure 15: System Architecture 28](file:///C:\Users\xephysis\Desktop\Dropbox\Capston%20Project\종설%20문서\6.%20Final%20Report%20Doc\Final%20Specification.docx#_Toc327022142)

[Figure 16: Use Case 1 – Realization Diagram 31](#_Toc327022143)

[Figure 17: Use Case 1 – Relationship Diagram 32](#_Toc327022144)

[Figure 18: Use Case 2 – Realization Diagram 33](#_Toc327022145)

[Figure 19: Use Case 2 – Relationship Diagram 34](#_Toc327022146)

[Figure 20: Use Case 3 – Realization Diagram 35](#_Toc327022147)

[Figure 21: Use Case 3 – Relationship Diagram 36](#_Toc327022148)

[Figure 22: Use Case 4 – Realization Diagram 37](#_Toc327022149)

[Figure 23: Use Case 4 – Relationship Diagram 38](#_Toc327022150)

[Figure 24: Use Case 5 – Realization Diagram 39](#_Toc327022151)

[Figure 25: Use Case 5 – Relationship Diagram 40](#_Toc327022152)

[Figure 26: Use Case 6 – Realization Diagram 41](#_Toc327022153)

[Figure 27: Use Case 6 – Relationship Diagram 42](#_Toc327022154)

[Figure 28: Use Case 7 – Realization Diagram 43](#_Toc327022155)

[Figure 29: Use Case 7 – Relationship Diagram 44](#_Toc327022156)

[Figure 30: Use Case 8 – Realization Diagram 45](#_Toc327022157)

[Figure 31: Use Case 8 – Relationship Diagram 46](#_Toc327022158)

[Figure 32: Use Case 9 – Realization Diagram 47](#_Toc327022159)

[Figure 33: Use Case 9 – Relationship Diagram 48](#_Toc327022160)

[Figure 34: Action Manager Subsystem Class Diagram 49](#_Toc327022161)

[Figure 35: Sound Action Management Class Diagram 50](#_Toc327022162)

[Figure 36: Transfer Management Class Diagram 51](#_Toc327022163)

[Figure 37: Mouse Action Manager Class Diagram 52](#_Toc327022164)

[Figure 38: Keyboard Action Management Class Diagram 53](#_Toc327022165)

[Figure 39: PC Connection Management Class Diagram 54](#_Toc327022166)

[Figure 40: Device Connection Management Class Diagram 55](#_Toc327022167)

[Figure 41: Main Interface 56](#_Toc327022168)

[Figure 42: Connection Popup Interface 57](#_Toc327022169)

[Figure 43: File List Interface 58](#_Toc327022170)

[Figure 44. File Selection Popup Interface 59](#_Toc327022171)

[Figure 45: Record Voice Interface 60](#_Toc327022172)

[Figure 46: Transmit Voice to PC interface 61](#_Toc327022173)

[Figure 47: Keyboard, Mouse Button Action Interface 62](#_Toc327022174)

[Figure 48: Mouse Pointing Interface 63](#_Toc327022175)

[Figure 49: Test Result Table 88](#_Toc327022176)

[Figure 50: Project Status 88](#_Toc327022177)

# 1. Projects Overview

<Dynamic Presentation> aimed to provide easily an All-in-one technology to user for presentation when environment for presentation is not prepared in advance or a lack of equipment. A number of mike and remote-control application, based on Android platform, already developed. Because of some problems, for example delay, noise or UI without user experience, inconveniences often arise. Dynamic presentation can improves problems that existing system have and provides various and useful functions for user.

## 1.1. Scope and Objectives

### 1.1.1. Change about presentation tool - All-in-one

Dynamic Presentation provides just one system integrating mike, keyboard, pointing using Gyroscope sensor and virtual storage. This can replace existing mouse for presentation, control keyboard and mouse action and use documents with file management easily. The service containing mobility and convenience can be developed for presentation.

### 1.1.2. Improvement about presentation environment - Anywhere

Dynamic Presentation operates in Wireless network environment like 3G and Wi-Fi. Everywhere wireless network worked user can this anywhere.

### 1.1.3. Scope

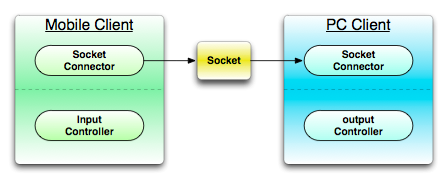


Figure 1: Project Scope

#### 1.1.3.1. Mobile Client

- Mobile Graphic User Interface

- Communication for connecting PC

- Mike, Recording

- Cloud service

- Keyboard action control

- Mouse action control

#### 1.1.3.2. PC Client

- Communication for connecting mobile device

- Output voice streaming

- Cloud service

- Keyboard action control

- Mouse action control

## 1.2. Supplementary Requirements

### 1.2.1. Introduction

This document includes all requirements that aren’t extracted when making Use case.

### 1.2.2. Functionality

- Logging and Error Handling  
- Every errors have to be saved in storage.  
- Pluggable Rule  
- System’s functions can be specialized using optional regulation that will be invoked at the point generating various scenario of Use Case.  
- Security  
- All user must be authorized.

### 1.2.3. Usability

- document  
- A manual that defined way to use is needed.  
- A help mode that explained each functions is needed.  
- A problem-solving document containing problems and solution is needed.

### 1.2.4. Human Factors

- Customer can see Dynamic Presentation system using proper sized screen. So, the letters have to consist of suitable color, size and font.  
- Customer wants to use Dynamic Presentation quickly. Then he/she has not a correct understanding. Thus, it is a critical factor to prevent error generating and use system easily.

### 1.2.5 Reliability

- Recoverability  
- If system has problem with external problem, alternative is prepared in advance in order to handle that immediately.  
- Responsibility  
- The system sensor events (touch, acceleration and gravity) will respond to more than 95%.Because the value is to passed exactly.  
- A value of sensor that user inputs have to respond within 0.1 seconds.

### 1.2.6. Performance

- Speed  
- As be said in the Human Factors section, customer want to use Dynamic Presentation quickly. Therefore, delay has to take 0.5 second at most when system is used.

### 1.2.7. Supportability

- Internationality  
- System provides an English service for foreigner who has no ability in a Korean.

### 1.2.8. Implementation Constraints

• System have to be implemented using Java technical solution. It is anticipated to give a guarantee of portability and scalability.  
• System have to be implemented using Android SDK 2.3.

### 1.2.9. Interfaces

- Hardware Interfaces  
- Smartphone(Android SDK 2.3.3, Android API Ver.10)  
- Laptop or Desktop

- Software Interfaces  
- System have to provide an intuitive UI for customer to use system easily.

# 2. Customer Requirements

This section contains the architecture view of your use-case model.  
It may include natural language or (and) use-case diagram(s), actor descriptions and prioritized use-case descriptions.

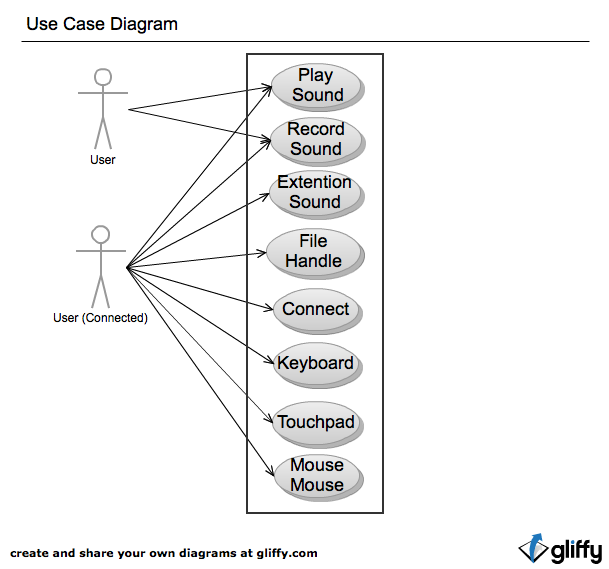


Figure 2: Use Case Diagram

## 2.1. User requirements

- The system should provide users with connectivity features.  
- The system should be able to enter the required information to connect to user.

- The system should be telling wrong, if the user's connection information is wrong.  
- The system will be allowed use some features, even though system is not connected.  
- The system provides disconnect feature for the user.  
- The system provides the feature to store connected information.  
- The system provides the feature to record user's sound.  
 - The system should provide user's voice recording and allows user to enter information in the file.

- The system should allow storing user’s recorded sound.

- The system should allow managing files stored.

- The system provides the feature to transfer user’s voice to PC  
 - The system should be prevented delays when transfer sound.  
- The system provides user's mouse pointer control.  
- The system provides user's mouse click control.  
- The system provides user's motion mouse pointer control.  
- The system provides user's keyboard control.  
- The system provides the feature to save files.  
 - The system should show the details of the file.  
- The system provides the feature to transfer files.

- The system should show the list of stored files.

## 2.2. System requirements

- The system provides a connection between PC to device.

- The system might be able to input data needed by user in order to connect between PC to device

- When it failed to connect, a notice will be shown.

- User might be able to using system partially without connection.

- The system provides a result depends on connection.

- System provides the feature to connected user as follows.

- PC in the sound transmission.

- The system activate the feature to transmit a sound to PC

- Mouse control

- The system activate the feature to transmit a movement of mouse controlled by user to PC

- File transmission

- The system provides the feature to transmit user selected file to PC

- The system provides a function to do not connect user as follows.

- Recording

- The system saves a voice in the device.

- Showing file list

- The system provides the feature to show a file in the device.

- The system provides the feature to user wanted connect to system as follows.

- Trying to connect to the system asks to enter the eigenvalues ​​of the PC.

- The user must enter the correct value that does not deviate from the type of eigenvalues at prompt.

- The system provides disconnect feature to the PC.

- The system provides that feature disconnection information to user that want to disconnect.

- The system provides feature to record sound..

- When user wants record the system provides enabled recording option feature to user.

- The system provides feature that transmit for user voice.

- When user want transmit the voice to PC the system provides feature after enabled option.

- The device must be requiring input the voice through the microphone.

- The system provides mouse control feature.

- The pointer that out of range should not move.

- It is prevented that multiple input pointing.

- The system activate feature to users who want to click.

-The mouse provides the ability to control points that user's movement.

- It should handle user's move within the area shall be controlled in system.

- The system provides keyboard control.

-The user should not require process that consist multiple actions in time.

- The system provides file transmission.

- The system provides detail information of file for using to user.

- The user activate transfer files menu.

## 2.3. Domain requirements

- When the system error occurs, the system records error cause and the time information.  
- When system creates a voice file in system storage should not be exceeded more than create a file.  
- The system can handle only allows handling for the file type.  
- The system file processing and file processing is handled separately about other details.  
- The system is used Two-dimensional version of the labeling system (e.g. 1.0).

# 3. Requirements Analysis

## 3.1. Structural Analysis

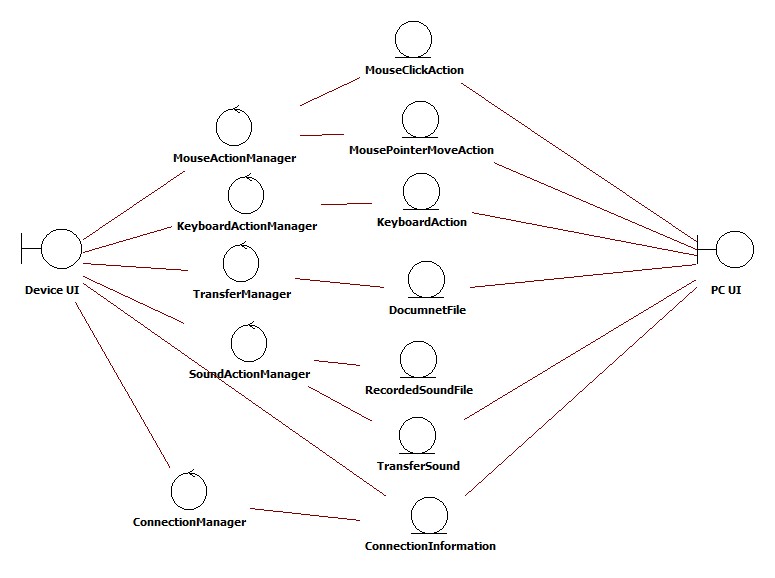


Figure 3: Project Structure

### 3.1.1. Boundary

- Device UI provides the user interface, the main interaction between the user and the system. Used to initiate recording sounds, controlling mouse and keyboard actions, file transference to PC. Call the ConnectionManager to connect device and pc.  
- PC UI shows the ConnectionInformation that use when user wants to start system.

### 3.1.2. Control

- MouseActionManager is the controller that user can move mouse pointer of computer and do click action remotely.   
- KeyboardActionManager provides function that can use keyboard remotely.  
- SoundActionManager is the controller that records sound and plays recorded sound file.  
- ConnectionManager is the controller that used in the system-initiation step before another manamerstarts.  
- TrasferManager is the controller that sends document and sound files from device to pc.

### 3.1.3. Entity

- MousePointerMoveAction  
- MouseClickAction  
- KeyboardAction  
- DocumnetFile  
- RecordedSoundFile  
- ConnectionInformation  
- TransferSound

## 3.2. Deployment Model

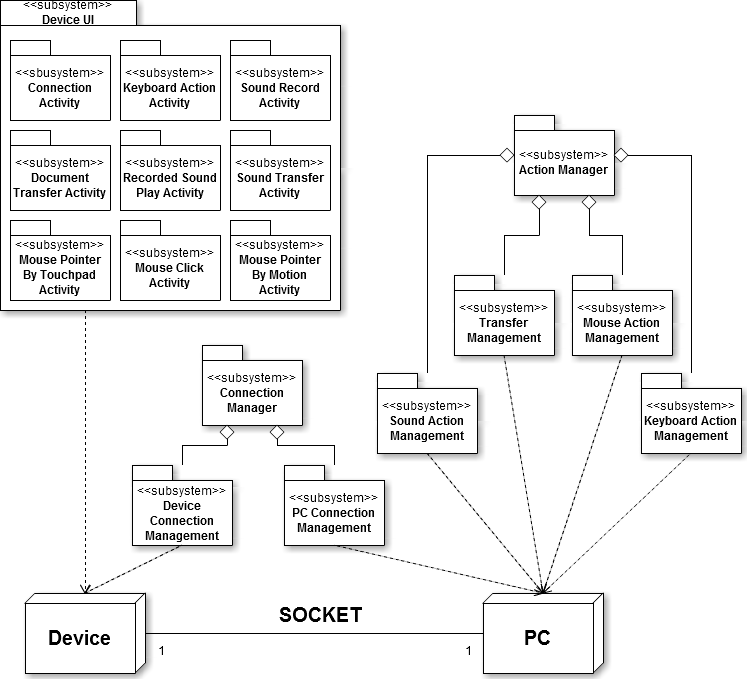


Figure 4: Deployment Model

## 3.3. Behavioral Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case | Actor(s) | Entry Condition | Exit Condition | Extends/includes/inherits |
| Process Connecting | User | User want to that PC connected to Device for Presentation. | Device is connected to a PC. | Process Connecting |
| Move mouse using touchpad | User | PC’s mouse cursor position is moved in the user desired direction. | PC’s mouse cursor position is moved in the user desired direction. | Process Connecting |
| Click mouse using touchpad | User | User wants that mouse clickhappened on PC. | Click event will happen on PC that user wanted. | Process Connecting |
| Move mouse pointer using motion | User | Users want to move the PC’s mouse pointer by motion. | PC’s mouse cursor position is moved in the user desired direction, when user moves device. | Process Connecting |
| Control keyboard | User | User want to that keyboard event occurs on PC. | PC keyboard input occurs that is user-selected keyboard inputted of a Device. | Process Connecting |
| Process Record Sound | User | User wants to record user’s voice on Device. | User's voice is recorded on the Device. | Process Connecting |
| Process Play Sound | User | User wants to hear recorded contents. | Voice that is recorded in device is outputted through the speakers on the Device. | Process Connecting |
| Transfer voice to PC speaker | User | User wants its voice through the speakers on the PC is played in real time. | If User says, voice through the speakers on the PC is played in real time. | Process Connecting |
| Transfer voice to PC speaker | User | User wants to use files in the PC from the Device. | UC 1. Should be progressed. | Process Connecting |

Figure 5: Behavior Analysis

|  |  |
| --- | --- |
| **Use Case 1 :** | Process Connecting |
| **Scope :** | Dynamic Presentation |
| **Level :** | user-goal |
| **Primary Actor:** | user |
| **Stakeholders and interests:** | User want to that PC connected to Device for Presentation. |
| **Preconditions:** | None. |
| **Success Guarantee :** | Device is connected to a PC. |
| **Main Success Scenario :** | 1. PC displays the information that can connect to PC.  2. Device makes user to input information that can connect to PC.  3. User enter the PC’s information to connect on Device  4. PC will compare to information with user’s input.  5. PC and Device notice the success connection. |
| **Extensions :** | 5a. If connection failed  1. Device notify to user that connection failed.  2. Device ask to user that user try to reconnection.  2.a If user want to try reconnection.  1. User enter the PC’s information to connect on Device.(Main #3.)  2.b If user want not to try reconnection.  1. System cancel the connecting action. |
| **Special Requirements:** | None. |
| **Frequency of Occurrence:** | Occurs only once at the initial operation. |
| **Open Issues:** | Before user enter the information directly, The system will show a list could be connected?  Is there any way other method of connection that proposed? |
|  | C:\Users\ajou\Desktop\Dropbox\Capston Project\종설 문서\Requirements Doc\Requirements Doc\AD_Connect.png |

Figure 6: Use Case 1. Connecting Process

|  |  |
| --- | --- |
| **Use Case 2 :** | Move mouse using touchpad |
| **Scope :** | Dynamic Presentation |
| **Level :** | user-goal |
| **Primary Actor:** | user |
| **Stakeholders and interests:** | Users want to move the PC’s mouse pointer. |
| **Preconditions:** | UC 1. Should be progressed. |
| **Success Guarantee :** | PC’s mouse cursor position is moved in the user desired direction. |
| **Main Success Scenario :** | 1. Device shows the User to move the mouse by touchpad.  2. User places its finger on Device’s screen, and then it moves in the desired direction.  3. PC shows the movement of the mouse pointer in user's desired direction. |
| **Extensions :** | \*a. If system failed.  1. If between PC and Device connection is lost.  1. Device found out that disconnected from PC.  2. Device attempts to reconnect based on information used in UC.1  2.a If external problem occurs that prevent connection.  1. System notifies the user for problem by Device.  2. If system terminated.  1. User restarts system for resolving problem. |
| **Special Requirements:** | User should be able to adjust mouse sensitivity.  Move the mouse pointer should be executed immediately |
| **Frequency of Occurrence:** | Frequently occurs. |
| **Open Issues:** | System as the difference between the sizes of the screen by any modifications needed?  How strong is user's sensitivity to feel comfortable? |
|  | C:\Users\ajou\Desktop\Dropbox\Capston Project\종설 문서\Requirements Doc\Requirements Doc\AD_Touchpad.png |

Figure 7: Use Case 2. Move Mouse by Touchpad

|  |  |
| --- | --- |
| **Use Case 3 :** | Click mouse using touchpad |
| **Scope :** | Dynamic Presentation |
| **Level :** | user-goal |
| **Primary Actor:** | user |
| **Stakeholders and interests:** | User wants that mouse clickhappened on PC. |
| **Preconditions:** | UC 1. Should be progressed. |
| **Success Guarantee :** | Click event will happen on PC that user wanted. |
| **Main Success Scenario :** | 1. Device shows the User to click the mouse.  2. User select mouse click on Device.  3. Device notifies that user’s action is normally processed. |
| **Extensions :** | \*a. If system failed.  1. If between PC and Device connection is lost.  1. Device found out that disconnected from PC.  2. Device attempts to reconnect based on information used in UC.1  2.a If external problem occurs that prevent connection.  1. System notifies the user for problem by Device.  2. If system terminated.  1. User restarts system for resolving problem.  2.a If user select mouse left click.  1. PC generates mouse left click event.  2.b If user select mouse right click.  1. PC generates mouse right click event. |
| **Special Requirements:** | Click the mouse on PC should be executed immediately.  Device uses sound and vibration for notify that user’s action is normally processed. |
| **Frequency of Occurrence:** | Frequently occurs. |
| **Open Issues:** | None. |
|  | C:\Users\ajou\Desktop\Dropbox\Capston Project\종설 문서\Requirements Doc\Requirements Doc\AD_MouseClick.png |

Figure 8: Use Case 3. Click Mouse

|  |  |
| --- | --- |
| **Use Case 4 :** | Move mouse pointer using motion |
| **Scope :** | Dynamic Presentation |
| **Level :** | user-goal |
| **Primary Actor:** | User |
| **Stakeholders and interests:** | Users want to move the PC’s mouse pointer by motion. |
| **Preconditions:** | 1. UC 1. Should be progressed.  2. Device must operate acceleration sensor. |
| **Success Guarantee :** | PC’s mouse cursor position is moved in the user desired direction, when user moves device. |
| **Main Success Scenario :** | 1. Device shows the user to move the mouse by motion.  2. User moves Device in the direction.  3. PC shows the movement of the mouse pointer in user's desired direction. |
| **Extensions :** | \*a. If system failed.  1. If between PC and Device connection is lost.  1. Device found out that disconnected from PC.  2. Device attempts to reconnect based on information used in UC.1  2.a If external problem occurs that prevent connection.  1. System notifies the user for problem by Device.  2. If system terminated.  1. User restarts system for resolving problem. |
| **Special Requirements:** | The mouse pointer moves by the motion must be executed immediately. |
| **Frequency of Occurrence:** | Frequently occurs. |
| **Open Issues:** | How strong is user's sensitivity to feel comfortable? |
|  | C:\Users\ajou\Desktop\Dropbox\Capston Project\종설 문서\Requirements Doc\Requirements Doc\AD_MotionMouse.png |

Figure 9: Use Case 4. Move Mouse by Motion

|  |  |
| --- | --- |
| **Use Case 5 :** | Control keyboard |
| **Scope :** | Dynamic Presentation |
| **Level :** | user-goal |
| **Primary Actor:** | User |
| **Stakeholders and interests:** | User want to that keyboard event occurs on PC. |
| **Preconditions:** | UC 1. Should be progressed. |
| **Success Guarantee :** | PC keyboard input occurs that is user-selected keyboard inputted of a Device. |
| **Main Success Scenario :** | 1. Device shows the user a choice of possible keyboard actions.  2. User selects keyboard action on the Device.  3. Device notifies that user’s action is normally processed.  4. PC executes user-selected keyboard action. |
| **Extensions :** | \*a. If system failed.  1. If between PC and Device connection is lost.  1. Device found out that disconnected from PC.  2. Device attempts to reconnect based on information used in UC.1  2.a If external problem occurs that prevent connection.  1. System notifies the user for problem by Device.  2. If system terminated.  1. User restarts system for resolving problem. |
| **Special Requirements:** | Device uses sound and vibration for notify that user’s action is normally processed.  Keyboard operation must be executed immediately |
| **Frequency of Occurrence:** | Frequently occurs. |
| **Open Issues:** | None. |
|  | C:\Users\ajou\Desktop\Dropbox\Capston Project\종설 문서\Requirements Doc\Requirements Doc\AD_Keyboard.png |

Figure 10: Use Case 5. Keyboard Control

|  |  |
| --- | --- |
| **Use Case 6 :** | Process Record Sound |
| **Scope :** | Dynamic Presentation |
| **Level :** | user-goal |
| **Primary Actor:** | User |
| **Stakeholders and interests:** | User wants to record user’s voice on Device. |
| **Preconditions:** | Device has an audio input device. |
| **Success Guarantee :** | User's voice is recorded on the Device. |
| **Main Success Scenario :** | 1. Device makes user to choose recording voice.  2. User should select the voice recording.  3. Device notifies to recording starts.  4. User should enter the voice on the Device.  5. User stop the speaking. |
| **Extensions :** | \*a. If system failed.  1. If between PC and Device connection is lost.  1. Device found out that disconnected from PC.  2. Device attempts to reconnect based on information used in UC.1  2.a If external problem occurs that prevent connection.  1. System notifies the user for problem by Device.  2. If system terminated.  1. User restarts system for resolving problem.  5.a If user want to save recording voice.  1. User unselect recording voice.  2. User saves the recorded voice.  5.b If user want to continue recording voice.  1. User re-select voice recording. |
| **Special Requirements:** | Device must have space to store the voice in.  User-recorded voice of the original speech and should not be a big difference.  User should be able to control the volume that sound stored in device. |
| **Frequency of Occurrence:** | Sometimes. |
| **Open Issues:** | None. |
|  | C:\Users\ajou\Desktop\Dropbox\Capston Project\종설 문서\Requirements Doc\Requirements Doc\AD_RecordSound.png |

Figure 11: Use Case 6. Record Sound

|  |  |
| --- | --- |
| **Use Case 7 :** | Process Play Sound |
| **Scope :** | Dynamic Presentation |
| **Level :** | user-goal |
| **Primary Actor:** | User |
| **Stakeholders and interests:** | User wants to hear recorded contents. |
| **Preconditions:** | None. |
| **Success Guarantee :** | Voice that is recorded in device is outputted through the speakers on the Device. |
| **Main Success Scenario :** | 1. Device makes user to choose play recorded contents.  2. User chooses to play the recorded voice.  3. Device shows a list of the recorded voice to the user.  4. User selects recording the voice in list.  5. System makes the user to listen the selected voice. |
| **Extensions :** | \*a. If system failed.  1. If between PC and Device connection is lost.  1. Device found out that disconnected from PC.  2. Device attempts to reconnect based on information used in UC.1  2.a If external problem occurs that prevent connection.  1. System notifies the user for problem by Device.  2. If system terminated.  1. User restarts system for resolving problem.  3.a If device not have a recorded voice.  1. Device notifies that there is a recorded voice.  5.a If system fails play the recorded voice.  1. Device notifies to user of problem. |
| **Special Requirements:** | User shall be able to adjust the volume. |
| **Frequency of Occurrence:** | None. |
| **Open Issues:** | Only playback of recorded voice will provide on Device?  The concept of sharing will need to be added? |
|  | C:\Users\ajou\Desktop\Dropbox\Capston Project\종설 문서\Requirements Doc\Requirements Doc\AD_PlaySound.png |

Figure 12: Use Case 7. Play Recorded

|  |  |
| --- | --- |
| **Use Case 8 :** | Transfer voice to PC speaker |
| **Scope :** | Dynamic Presentation |
| **Level :** | user-goal |
| **Primary Actor:** | User |
| **Stakeholders and interests:** | User wants its voice through the speakers on the PC is played in real time. |
| **Preconditions:** | UC 1. Should be progressed. |
| **Success Guarantee :** | If User says, voice through the speakers on the PC is played in real time. |
| **Main Success Scenario :** | 1. Device makes user to choose transfer voice to PC speaker.  2. User chooses to transfer voice to PC speaker.  3. User should enter the voice on the Device.  4. PC output the voice that entered by the user.  5. User terminates voice transfer. |
| **Extensions :** | \*a. If system failed.  1. If between PC and Device connection is lost.  1. Device found out that disconnected from PC.  2. Device attempts to reconnect based on information used in UC.1  2.a If external problem occurs that prevent connection.  1. System notifies the user for problem by Device.  2. If system terminated.  1. User restarts system for resolving problem.  5.a If user wants to continue transfer voice.  1. User should enter the voice on the Device.(Main Scenario #3.)  5.b If user wants not to continue transfer voice.  1. User deselects voice transfer. |
| **Special Requirements:** | User's real-time voice transmission should be executed immediately.  User's voice input of the time and output to the PC's speakers of the time difference should be enough not to inconvenience.  Device and the PC support that user should be able to adjust the volume. |
| **Frequency of Occurrence:** | Frequently occurs. |
| **Open Issues:** | How could reduce voice transmission time? |
|  | C:\Users\ajou\Desktop\Dropbox\Capston Project\종설 문서\Requirements Doc\Requirements Doc\AD_ExtentionSound.png |

Figure 13: Use Case 8. Transfer Voice

|  |  |
| --- | --- |
| **Use Case 9 :** | Transfer document |
| **Scope :** | Dynamic Presentation |
| **Level :** | user-goal |
| **Primary Actor:** | User |
| **Stakeholders and interests:** | User wants to use files in the PC from the Device. |
| **Preconditions:** |  |
| **Success Guarantee :** | UC 1. Should be progressed. |
| **Main Success Scenario :** | 1. Device makes User to choose file transmission.  2. User select file transmission.  3. Device shows the complete list of available files to send.  4. User selects the file.  5. Device shows the file selected by the user.  6. Users should check the correct of the selected file.  7. Device shows the file transfer process.  8. System will execute the file selected by the user. |
| **Extensions :** | \*a. If system failed.  1. If between PC and Device connection is lost.  1. Device found out that disconnected from PC.  2. Device attempts to reconnect based on information used in UC.1  2.a If external problem occurs that prevent connection.  1. System notifies the user for problem by Device.  2. If system terminated.  1. User restarts system for resolving problem.  3a. If file list is empty.  1. System notifies to the user that file list is empty.  6a. If user wants to transfer other files.  1. User cancels the file transfer.  2. Goto Main Scenario #1. |
| **Special Requirements:** | Transfer files with other UC should be operated. |
| **Frequency of Occurrence:** | Sometimes. |
| **Open Issues:** | None. |
|  | C:\Users\ajou\Desktop\Dropbox\Capston Project\종설 문서\Requirements Doc\Requirements Doc\AD_FileHandling.png |

Figure 14: Use Case 8. Transfer Document

## 3.4. Analysis Packages

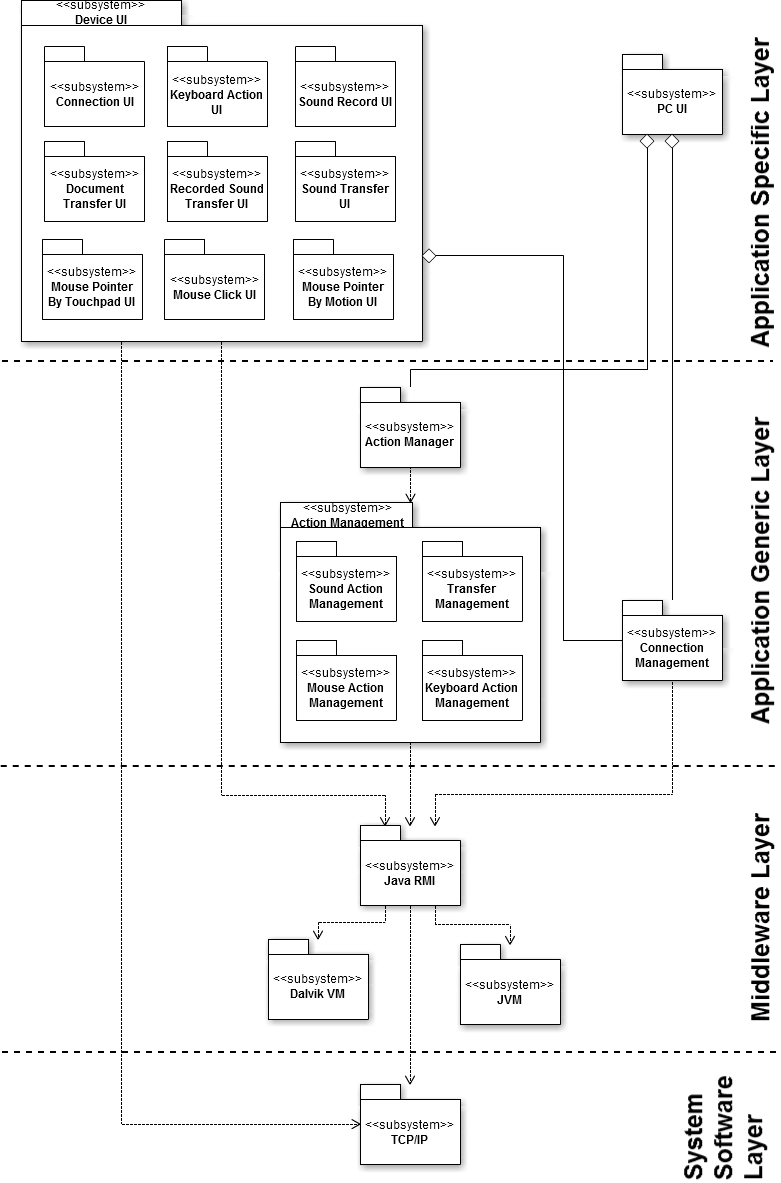


Figure 15: System Architecture

### 3.4.1. Application Specific Layer

This layer consists of the Device UI subsystem and PC UI subsystem. The Device UI further consists of the ConnectionUI, KeyboardActionUI, SoundRecordUI, DocumnetTransferUI, RecordedSoundTransferUI, SoundTransferUI, MousePointerByTouchpadUI, MouseClickUI and MousePointerByMotionUI. The role of this subsystem is to create and manage the user interfaces and screens.

### 3.4.2. Application Generic Layer

This layer consists of the various sub-systems which are independent of the system and it consists of subsystem like Action Manager Facade, Sound Action Management, Transfer Management, Mouse Action Management, Keyboard Action Management and Connection Management. The Action Manager Facade handles request sent from Device UI and invokes the appropriate subsystem on the PC UI. The Connection Management interacts with Device UI and PC UI to initiate system.

### 3.4.3. Middleware Layer

This layer consists of the system specific subsystems like Java RMI, Dalvik VM and JVM. This layer provides the medium by which system can be developed in the manner desired. The upper layers realize this subsystem to communicate with other subsystems to perform appropriate functions. Since the middleware layer supports Java platform or Android platform, all this subsystem directly depends upon the JVM subsystem and Dalvik subsystem.

### 3.4.4. Application System Software Layer

This layer consists of TCP/IP subsystem. This layer provides the functionality to other subsystem to communicate over the network.

# 4. Validation Criteria (metrics & measures)

The following items are validation method for the use cases.  
In addition, when a use case validation, strategy should be used is described.  
  
- Connecting use case (UC 1: Process Connecting)  
This use case will be tested by direct experiment.  
This use case is a prerequisite for the other use cases. Therefore, the accuracy and reliability is a key measurement target.  
  
- Control related use case (UC 2: Move mouse using touchpad, UC 3: Click mouse using touchpad, UC 4: Move mouse pointer using motion, UC 5 : Control keyboard)  
These use cases for each feature will be tested by direct experiment. The most important measure by the user whether or not the desired behavior is exactly reproducible.  
User does not feel at the time the works will be measured.  
In addition, each use case that progressed normally, user able to is make sure that.  
  
- Voice related use case (UC 6: Record voice, UC 7: Play recorded voice, UC 8: Transfer voice to PC speaker)  
These use cases for each feature will be tested by direct experiment.  
In addition, the user's voice and the voice is processed by the system's time difference is being measured is important.  
It will be measured that the difference between user’s original voice and the voice that processed by the system.

- Cloud service use case (UC 9: Transfer document)  
These use cases for each feature will be tested by direct experiment.  
Because there should be no errors will measure the integrity.

# 5. Use Case Realization Design

## 5.1. Use Case 1 - Process Connection

### 5.1.1. Use Case 1 – Realization



Figure 16: Use Case 1 – Realization Diagram

Connection begins with the PC UI display the required connection information and connection credential on input on the system. And Connection Activity display input interface for the user. Then the user input connection information and connection credential on Connection Activity on Device. That will call inputInfo() method. In DeviceConnectionManager call connect() method that has consists of connection info and connection credential. PCConnectionmanager receive credential and call verifyCredential() method for that check verification. If credential is valid PCconnectionManager call displayStatus() for valid status. And return valid to DeviceConnectionManager. DeviceConnectionManager call displayStatus() method for display valid status to the user. If not PCconnectionManager call displayStatus() for invalid status. And return invalid to DeviceConnectionManager. DeviceConnectionManager call displayStatus() method for display invalid status to the user. Also invoke confirmRetry() method that is ask the user retry connect. If the user answer yes, that action call accept() method and go to sequence 2. If not that action call deny() method, consequence invoke cancelConnection() method in DeviceConnectionManager.

### 5.1.2. Use Case 1 – Relationship

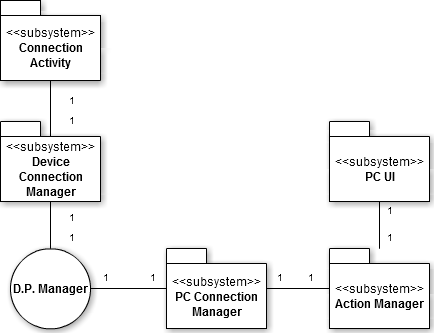


Figure 17: Use Case 1 – Relationship Diagram

The Connection Activity uses exactly one Device Connection Manager. The Device Connection Manager is used by exactly one Connection Activity. They are mutually aware of each other. The PC Connection Manager uses exactly one D.P. Manager. The D.P. Manager is used by exactly one PC Connection Manager. The Action Manager uses PC Connection Manager. The PC Connection Manager is used by exactly one Action Manager. The PCUI uses exactly one Action Manager. The Action Manager is used by exactly one PC UI. They are mutually aware of each other.

## 5.2. Use Case 2 – Move mouse using touchpad

### 5.2.1. Use Case 2 – Realization



Figure 18: Use Case 2 – Realization Diagram

This use case begins with MousePointerbyTouchpad Activity display touchpad interface for user. The user will invoke onTouch() method by touch on Device. and then MousePointerbyTouchpad Activity call sendMouse() method on DeviceConnectionManager that has consists differential x position, and differential y position. DeviceConnectionManager write on session, that received by PCConnetionManager. PCconnectionManager call moveMouse() method on MouseActionManager. MouseActionManager actually move by mouseMove() method. That's result displayed by PCUI. The user acknowledges the result.

### 5.2.2. Use Case 2 – Relationship

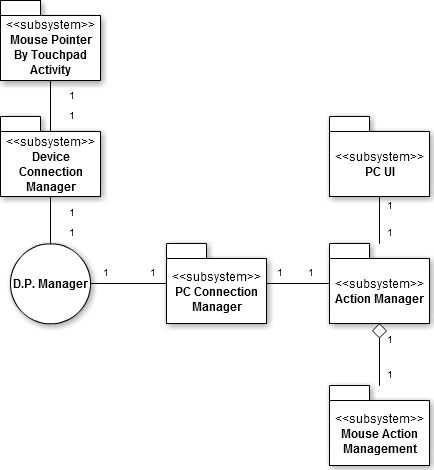


Figure 19: Use Case 2 – Relationship Diagram

The Mouse Pointer by Touchpad Activity uses exactly one Device Connection Manager. The Device Connection Manager is used by exactly one Mouse Pointer by Touchpad Activity. They are mutually aware of each other. The PC Connection Manager uses exactly one D.P. Manager. The D.P. Manager is used by exactly one PC Connection Manager. The Action Manager uses PC Connection Manager. The PC Connection Manager is used by exactly one Action Manager. The PC UI uses exactly one Action Manager. The Action Manager is used by exactly one PC UI. They are mutually aware of each other. The Action Manager is composed of Mouse Action Management. Mouse Action Management composes Action Manager. Action Manager is aware of Mouse Action Management of which it is composed, but not vice versa.

## 5.3. Use Case 3 – Click mouse using touchpad

### 5.3.1. Use Case 3 – Realization



Figure 20: Use Case 3 – Realization Diagram

This use case begins with MouseClick Activity displays touchpad interface for user. The user will invoke onTouch() method by touch on Device concurrently Mouse Click Activity call response() method for assure the user's action. And then MouseClick Activity call sendMouse() method on DeviceConnectionManager that has consists mouse factor -left or right. DeviceConnectionManager write on session, that received by PCConnectionManager. PCconnectionManager call moveMouse() method on MouseActionManager. MouseActionManager actually click by mouseClick() method. That's result displayed by PCUI. The user acknowledges the result.

### 5.3.2. Use Case 3 – Relationship

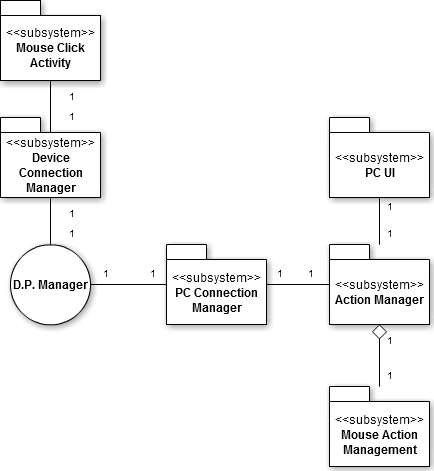


Figure 21: Use Case 3 – Relationship Diagram

The Mouse Click Activity uses exactly one Device Connection Manager. The Device Connection Manager is used by exactly one Mouse Click Activity. They are mutually aware of each other. The PC Connection Manager uses exactly one D.P. Manager. The D.P. Manager is used by exactly one PC Connection Manager. The Action Manager uses PC Connection Manager. The PC Connection Manager is used by exactly one Action Manager. The PC UI uses exactly one Action Manager. The Action Manager is used by exactly one PC UI. They are mutually aware of each other. The Action Manager is composed of Mouse Action Management. Mouse Action Management composes Action Manager. Action Manager is aware of Mouse Action Management of which it is composed, but not vice versa.

## 5.4. Use Case 4 – Move mouse using motion

### 5.4.1. Use Case 4 – Realization



Figure 22: Use Case 4 – Realization Diagram

This use case begins with MousePointer byMotion Activity display button interface for user. The user will invoke onTouch() method by touch on Device. And then the user starts move actually device. MousePointerbyMotionActviticy call sendMouse() method on DeviceConnectionManager that has consists differential x position, and differential y position by accelerometer. DeviceConnectionManager write on session, that received by PCConnectionManager. PCconnectionManager call moveMouse() method on MouseActionManager. MouseActionManager actually move by mouseMove() method. That's result displayed by PCUI. The user acknowledges the result.

### 5.4.2. Use Case 4 – Relationship

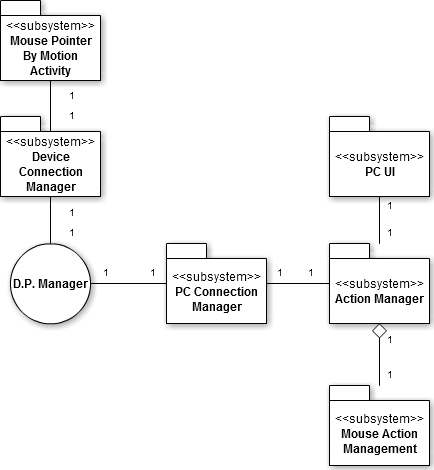


Figure 23: Use Case 4 – Relationship Diagram

The Mouse Pointer by Motion Activity uses exactly one Device Connection Manager. The Device Connection Manager is used by exactly one Mouse Pointer by Motion Activity. They are mutually aware of each other. The PC Connection Manager uses exactly one D.P. Manager. The D.P. Manager is used by exactly one PC Connection Manager. The Action Manager uses PC Connection Manager. The PC Connection Manager is used by exactly one Action Manager. The PC UI uses exactly one Action Manager. The Action Manager is used by exactly one PC UI. They are mutually aware of each other. The Action Manager is composed of Mouse Action Management. Mouse Action Management composes Action Manager. Action Manager is aware of Mouse Action Management of which it is composed, but not vice versa.

## 5.5. Use Case 5 – Control keyboard

### 5.5.1. Use Case 5 – Realization



Figure 24: Use Case 5 – Realization Diagram

This use case begins with KeyboardAction Activity displays keyboard interface for user. The user will invoke onTouch() method by touch on Device concurrently KeyboardAction Activity call response() method for assure the user's action. And then KeyboardAction Activity call sendKeyboard() method on DeviceConnectionManager that has consists keyboard factor. DeviceConnectionManager write on session, that received by PCConnectionManager. PCconnectionManager call keyPress() method on KeyboardActionManager. KeyboardActionManager actually press the keyboard by keyboardPress() method. That's result displayed by PCUI. The user acknowledges the result.

### 5.5.2. Use Case 5 – Relationship

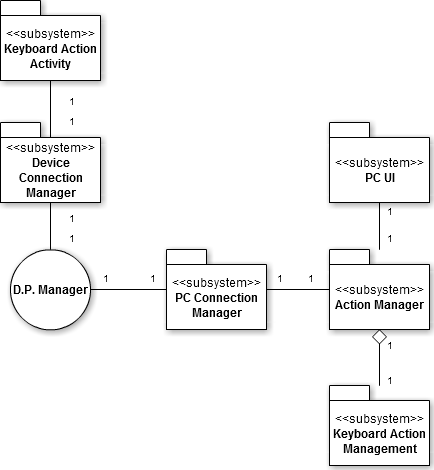


Figure 25: Use Case 5 – Relationship Diagram

The Keyboard Action Activity uses exactly one Device Connection Manager. The Device Connection Manager is used by exactly one Keyboard Action Activity. They are mutually aware of each other. The PC Connection Manager uses exactly one D.P. Manager. The D.P. Manager is used by exactly one PC Connection Manager. The Action Manager uses PC Connection Manager. The PC Connection Manager is used by exactly one Action Manager. The PC UI uses exactly one Action Manager. The Action Manager is used by exactly one PC UI. They are mutually aware of each other. The Action Manager is composed of Keyboard Action Management. Keyboard Action Management composes Action Manager. Action Manager is aware of Keyboard Action Management of which it is composed, but not vice versa.

## 5.6. Use Case 6 – Process record sound

### 5.6.1. Use Case 6 – Realization



Figure 26: Use Case 6 – Realization Diagram

Record sound process use case will starts with SoundRecord Activity display record interface to user. The user will invoke onTouch() method on Device concurrently SoundRecord Activity call response() method for assure the user's action. And SoundRecord Activity call startRecord() method on DeviceSoundActionManager. DeviceSoundActionManager will call itself initRecord() method for start recording. After the user speak and sound will be recorded. Finish speaking, the user invoke onTouch() method on Device for pause recording. SoundRecord Activity invoke pauseRecord() method on DeviceSoundActionManager consequence invoke askContinue() method for user's next selection. If the user answer yes by onTouch() method SoundRecordActivity call continueRecord() method. If the user answer no by onTouch() method SoundRecord Activity call stopRecord() method on DeviceSoundActonManager. After that DeviceSoundActionManager call itself saveRecord() method for saving recorded file.

### 5.6.2. Use Case 6 – Relationship

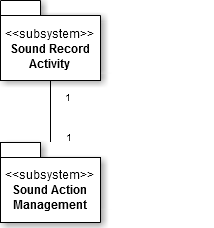


Figure 27: Use Case 6 – Relationship Diagram

The Sound Record Play Activity uses exactly one Sound Action Management. The Sound Action Management is used by exactly one Sound Record Activity. They are mutually aware of each other.

## 5.7. Use Case 7 – Process play sound

### 5.7.1. Use Case 7 – Realization



Figure 28: Use Case 7 – Realization Diagram

Paly sound process use case will starts with RecordedSoundPlay Activity display play interface to user. The user will invoke onTouch() method on Device concurrently Recorded Sound Play Activitycall response() method for assure the user's action. If device has not recorded file RecordedSoundPlay Activitycall notify() method for no file. If device has recorded file RecordedSoundPlay Activitycall displayRecordedList() method for user. And then the user select file on list that call onTouch() method.If that cannot play recorded sound, RecordedSoundPlay Activitycall notify() which consists can't play. If that can play the file RecordedSoundPlay Activity call playRecordedStart() method on DeviceSoundActionManager. DeviceSoundActionManager call itself playRecordedInit().

### 5.7.2. Use Case 7 – Relationship

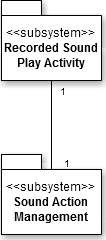


Figure 29: Use Case 7 – Relationship Diagram

The Recorded Sound Play Activity uses exactly one Sound Action Management. The Sound Action Management is used by exactly one Recorded Sound Play Activity. They are mutually aware of each other.

## 5.8. Use Case 8 – Transfer voice to PC speaker

### 5.8.1. Use Case 8 – Realization



Figure 30: Use Case 8 – Realization Diagram

Transfer voice to PC speaker use case will starts with SoundTransfer Activity display transfer voice interface to user. The user will invoke onTouch() method on Device concurrently SoundTransfer Activitycall response() method for assure the user's action. SoundTransfer Activitycall transferVoiceInit() method on DeviceSoundActionManager. Consequently DeviceSoundActionManager call transferVoiceInit() method on DeviceConnectionManager for making voice session. DeviceConnectionManager call itself sendVoice() method for PCConnectionManager preparing transfer voice. PCConnectionManager call transferVoiceInit() method on SoundActionManager. After preparing transfer voice the user speaking and SoundActionManager call palyOutput() method that actually use PC speaker. If the user wants to stop transferring voice.The user call onTouch() method on SoundTransfer Activity. SoundTransfer Activitycall transferVoiceTerm() method consequently DeviceSoundActionManager call transferVoiceTerm() for terminate session. After that DeviceConnectionManager call itself sessionClose() method.

### 5.8.2. Use Case 8 – Relationship

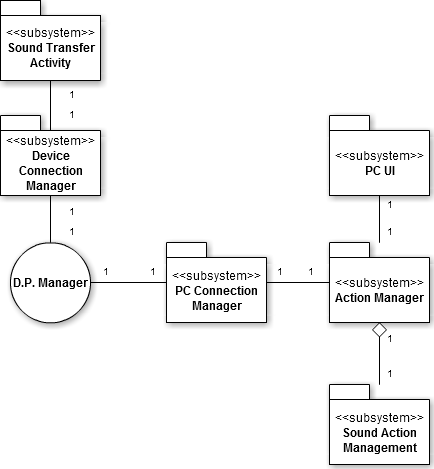


Figure 31: Use Case 8 – Relationship Diagram

The Sound Transfer Activity uses exactly one Device Connection Manager. The Device Connection Manager is used by exactly one Sound Transfer Activity. They are mutually aware of each other. The PC Connection Manager uses exactly one D.P. Manager. The D.P. Manager is used by exactly one PC Connection Manager. The Action Manager uses PC Connection Manager. The PC Connection Manager is used by exactly one Action Manager. The PC UI uses exactly one Action Manager. The Action Manager is used by exactly one PC UI. They are mutually aware of each other. The Action Manager is composed of Sound Action Management. Sound Action Management composes Action Manager. Action Manager is aware of Sound Action Management of which it is composed, but not vice versa.

## 5.9. Use Case 9 – Transfer document

### 5.9.1. Use Case 9 – Realization



Figure 32: Use Case 9 – Realization Diagram

Transfer Document use case will starts with DocumentTransfer Activity display transfer document interface to user. The user will invoke onTouch() method on Device concurrently DocumentTransfer Activity call response() method for assure the user's action. If Device has not any file to transfer DocumentTransfer Activity call notify() method to user for no file. If not DocumentTransfer Activity call displayFileList() for the user.

And then user select file by touch that call onTouch() method. DocumentTransfer Activity call confirmTransfer() method to user for that file is really wanted to transfer. If user deny by call deny() DocumentTransfer Activity call itself cancelTranfer() method. If not the user accept by call accept() method. If it ends, DocumentTransfer Activity call displayProgress() method. And call sendDocument() method on DeviceConnectionManager. DeviceConnectionManager uses sendDocument() on PCConnectionManager. PCConnectionManager call receiveDocument() method on TransferManager. In the end TransferManager call execDocumentFile() method.

### 5.9.2. Use Case 9 – Relationship

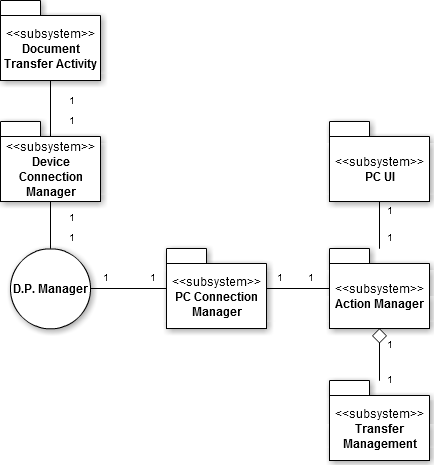


Figure 33: Use Case 9 – Relationship Diagram

The Document Transfer Activity uses exactly one Device Connection Manager. The Device Connection Manager is used by exactly one Document Transfer Activity. They are mutually aware of each other. The PC Connection Manager uses exactly one D.P. Manager. The D.P. Manager is used by exactly one PC Connection Manager. The Action Manager uses PC Connection Manager. The PC Connection Manager is used by exactly one Action Manager. The PC UI uses exactly one Action Manager. The Action Manager is used by exactly one PC UI. They are mutually aware of each other. The Action Manager is composed of Transfer Management. Transfer Management composes Action Manager. Action Manager is aware of Transfer Management of which it is composed, but not vice versa.

# 6. Subsystem Design

## 6.1. Action Manager Subsystem



Figure 34: Action Manager Subsystem Class Diagram

This subsystem is the main facade used in the system. It can invoke and call Sound Action Management, Transfer Management, Mouse Action Management, and Keyboard Action Management. It makes the decision to invoke a particular subsystem based on the request from the UI. For instance, if the user wishes to move a mouse pointer, the façade accepts this request and passes it on to a subsystem that can perform this operation. Hence, we can say that it acts as a simple interface to other subsystems.

## 6.2. Sound Action Management



Figure 35: Sound Action Management Class Diagram

Sound Action Management subsystem manages the actions related sound. According to user’s choice, this subsystem can record user voice to sound file and transfer this file from device to pc.

## 6.3. Transfer Management



Figure 36: Transfer Management Class Diagram

This subsystem is solely responsible for transfer control of the entire file. Because of this subsystem, it is possible to see the same document files –ppt, pdf, doc and so on – between Device and PC. A document file that user selected is transferred from device to PC5.4 Mouse Action Management

## 6.4. Mouse Action Manager



Figure 37: Mouse Action Manager Class Diagram

This subsystem controls user requests for moving mouse pointer on the pc-side. There are two kinds of method that user can be selected for movement of pointer. One is to use touchpad of device; the other is to use a result of motion. It receives user request, parses and insights. Then It moves mouse pointer to direction that user wants to move.

## 6.5. Keyboard Action Management



Figure 38: Keyboard Action Management Class Diagram

This subsystem management performs a requested operation that user wants. There is an action related keyboard in the requests. This subsystem analyzes these requests and invokes results of requests on the PC-side.

## 6.6. PC Connection Management



Figure 39: PC Connection Management Class Diagram

This subsystem is system that controls connection between PC and device. It uses own address and randomly generated number. Comparing these with some information provided from Device Connection Management, it can validate whether correct user attempts to connection or not.

## 6.7. Device Connection Management



Figure 40: Device Connection Management Class Diagram

This subsystem is responsible for connection management on the device. It interacts with PC Connection Management subsystem, gets initial connection information and some sequential random number from user and sends these data.

# 7. Human Interfaces

## 7.1. Client

### 7.1.1. Main

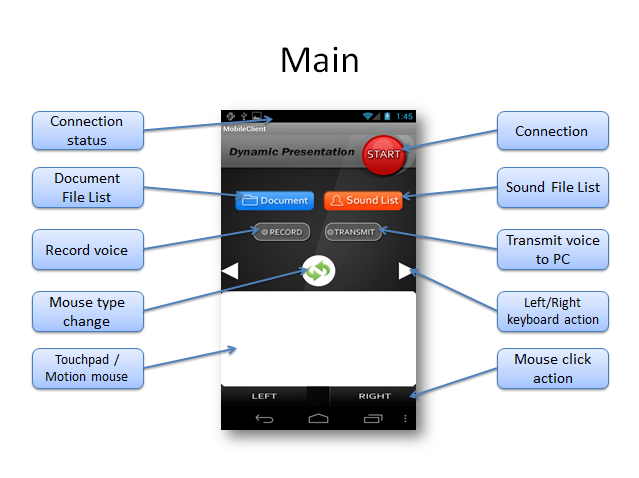


Figure 41: Main Interface

### 7.1.2. Connection and Connection Status

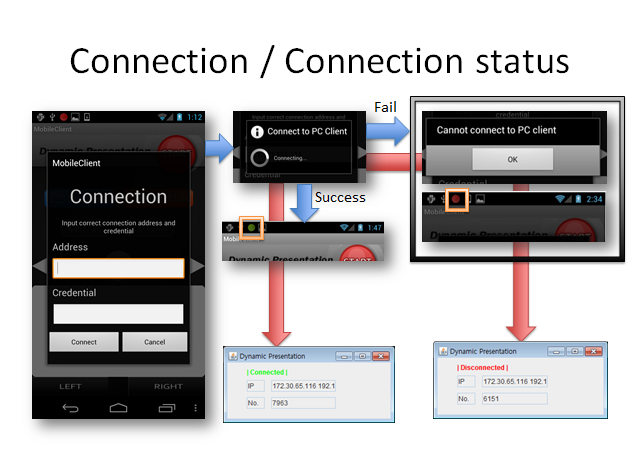


Figure 42: Connection Popup Interface

### 7.1.3. Document File List and File Transmission

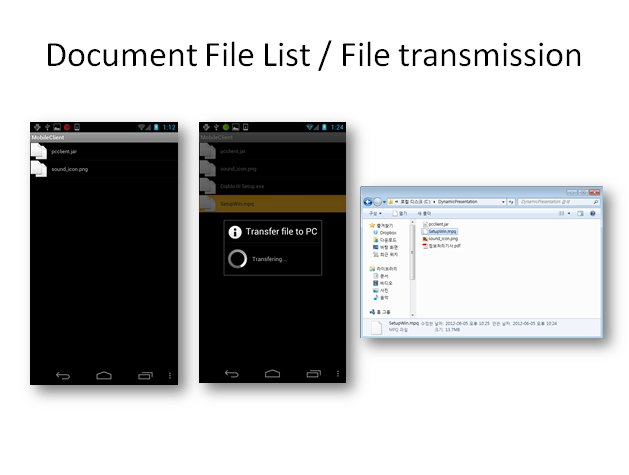


Figure 43: File List Interface

### 7.1.4. Sound File List and Play Sound

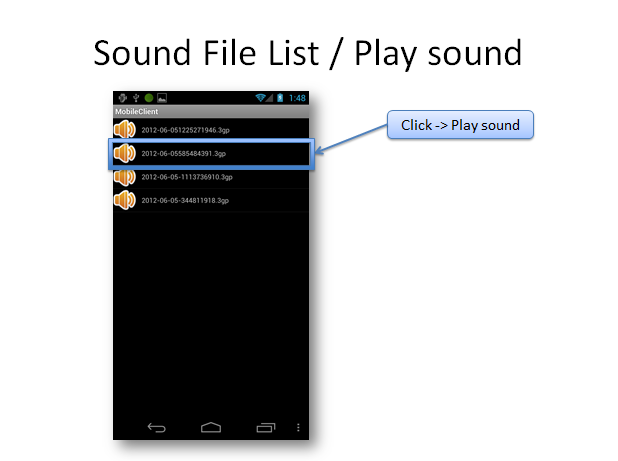


Figure 44. File Selection Popup Interface

### 7.1.5. Record Voice

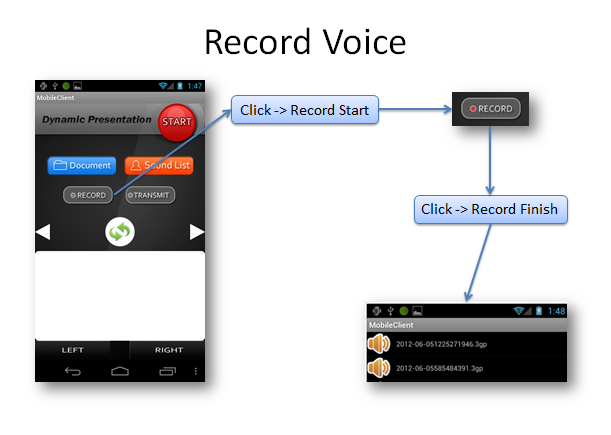


Figure 45: Record Voice Interface

### 7.1.6. Transfer voice to PC

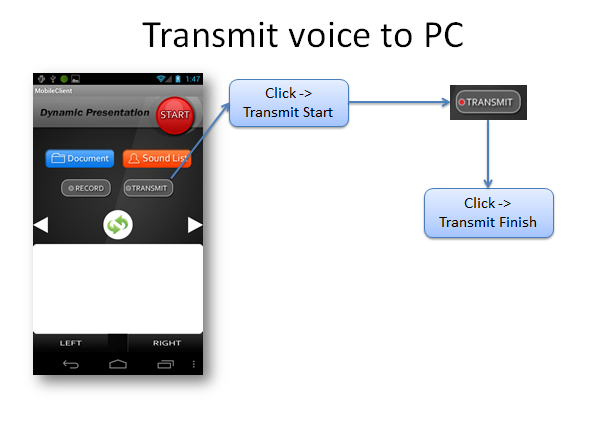


Figure 46: Transmit Voice to PC interface

### 7.1.7. Keyboard, Mouse Button Action

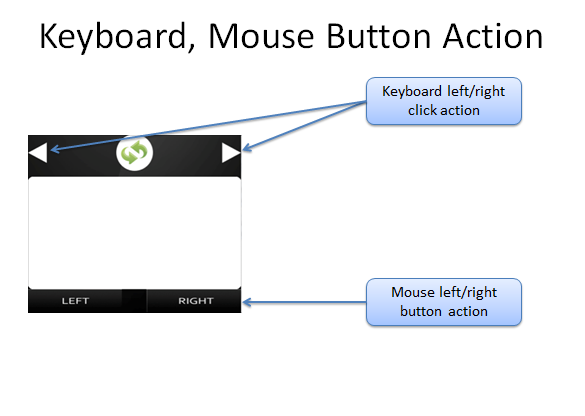


Figure 47: Keyboard, Mouse Button Action Interface

### 7.1.8. Mouse Pointing

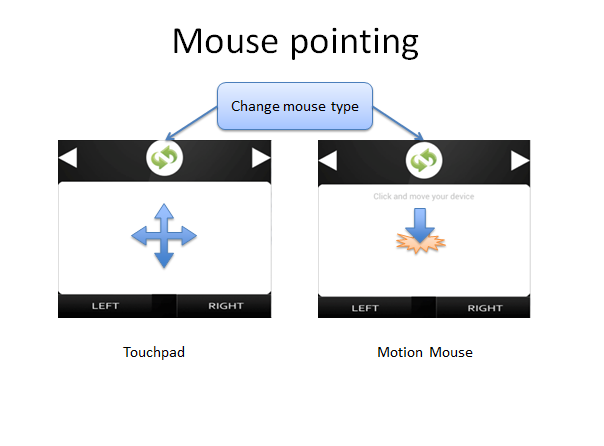


Figure 48: Mouse Pointing Interface

# 8. Use-Case Scenarios

## 8.1. Process Connecting

User want to that PC connected to Device for Presentation. A User presses a button at pc, and User confirms IP information of pc. User input PC information confirmed to DEVICE, and User press "connecting "button. If connection failed, show message called "Fail" to users at device screens "retry?" User show message called. If a user wants a repeat performance, User input again information of pc, and User press "cancellation" button, and Device cancel a connection action if User do not want a repeat performance. The Pc on the information received from the user's information if the connection is then compared with the "success" shows the message tells User.

## 8.2. Move mouse using touchpad

User wishes that a mouse pointer of pc moves. You show a touch pad screen if user press a mouse button of Device. User want moving to courses a user draws a finger to a screen of device.

If connection failed, show message called "Fail" to users at device screens "retry?" User show message called. If a user wants a repeat performance, User input again information of pc, and User press "cancellation" button, and Device cancel a connection action if User do not want a repeat performance. Pc shows a movement of a mouse pointer to the courses that a user wants if software normally operate. Device work to a former step if user presses an end button.

## 8.3. Click mouse using touchpad

User wishes that a mouse click of pc occurs. Device shows a mouse click button screen to users. A user selects a mouse click to want at device. If a user presses a button on the left of mouse, pc generates a button click event on the left of mouse and pc generates a button click event right a mouse if a user press the right button of mouse. If connection failed, show message called "Fail" to users at device screens "retry?" User show message called. If a user wants a repeat performance, User input again information of pc, and User press "cancellation" button, and Device cancel a connection action if User do not want a repeat performance. Device uses sound and vibration for notify that user’s action is normally processed. Device work to a former step if user presses an end button.

## 8.4. Move mouse pointer using motion

User wants to operate by motion. User looks if you press a mouse motion button of Device so as to be able to move in a pointer by motion. User are moved in the directions that a situation of a mouse pointer of pc wants if User move in the directions where a user wants device. If connection failed, show message called "Fail" to users at device screens "retry?" User show message called. If a user wants a repeat performance, User input again information of pc, and User press "cancellation" button, and Device cancel a connection action if User do not want a repeat performance. Device work to a former step if user presses an end button.

## 8.5. Control keyboard

User wishes that an event of a keyboard of PC occurs Device shows buttons of a keyboard shape to user. A user presses a keyboard button to want at Device. . Device uses sound and vibration for notify that user’s action is normally processed. If connection failed, show message called "Fail" to users at device screens "retry?" User show message called. If a user wants a repeat performance, User input again information of pc, and User press "cancellation" button, and Device cancel a connection action if User do not want a repeat performance. Device work to a former step if user presses an end button.

## 8.6. Process Record Sound

A user wants to record the contents that one spoke. Recording begins since a user pressed a Record button in a Device screen if you press a Start button. A Start button is changed by pause button and stop buttons. User inputs a voice to Device by a voice a user. Recording becomes a stop, and you finish recording if you press a Pause button if you press a Stop button. A recording file finished is stored to Device.

## 8.7. Process Play Sound

Voice that is recorded in device is outputted through the speakers on the Device. If a user presses a voice playing button to a Device screen, User can read file lists. A user selects a voice file wanting a voice file list after confirmation. Device displays message called "empty" if there is not a voice file recorded. And you show message called "Problem" and Device returns to a relevant issue to telephone planes if you cannot display a voice file recorded. Device work to a former step if user presses an end button.

## 8.8. Transfer voice to PC speaker

A user wants to display own voice at the PC speakers which are not Device. If a user presses a voice transmission button to a Device screen, output works at pc speakers.

## 8.9. Transfer document

A user wants to use a file of Device at PC. If User presses a file transmission button of Device, User can see the file list that transmission is possible. A system shows message called "empty" to User if there is not a file list. If select the file which User wants "is OK?" User show message called. If the file which User selected meets, you press a button. If User wants other file, you press a cancellation button, and you cancel file transmission. Device looks if you pressed an OK button. If connection failed, show message called "Fail" to users at device screens "retry?" User show message called. If a user wants a repeat performance, User input again information of pc, and User press "cancellation" button, and Device cancel a connection action if User do not want a repeat performance. Device work to a former step if user presses an end button.

# 9. System/Data Dependencies & Requirements

## 9.1. Software System

Java Runtime

Java SDK 1.6

Android SDK 4.0

Libraries: JMF

## 9.2. Hardware System

Android Platform Smartphone(Android 2.3)

Java Runtime Environment

Network Card

## 9.3. Data Dependencies & requirements

In this system, data is not occurred (just values occurred). Occurred values in device client are transferred directly. And then these values are realized directly in PC client. So that data has not exists. Also data dependencies & requirements not exist.

# 10. Testing Plan & Results

## 10.1 Introduction

<Dynamic Presentation> aimed to provide easily an All-in-one technology to user for presentation when environment for presentation is not prepared in advance or a lack of equipment. A number of microphone and remote-control application, based on Android platform, already developed. Because of some problems, for example delay, noise or UI without user experience, inconveniences often arise. Dynamic presentation can improves problems that existing system have and provides various and useful functions for user.

## 10.2 Goals and Objectives

### 10.2.1 Goals

The goals and objectives of testing the Dynamic presentation application are as follows:

* To ensure that our system requirements are verifiable and modifiable
* To ensure that Functionality, Usability, Human Factors, Reliability, Security, Efficiency, Accuracy
* To ensure that the data collected from the system user is real data
* To ensure that the persistence layer is well-designed
* To ensure systems users and friend added by the system user are authenticated using proper encryption techniques
* To ensure the security of the entire end-to-end application through the successful integration of the basic objectives of security which are confidentiality, availability, integrity, and usage

### 10.2.2. Statement of scope

There are several important aspects of the Dynamic presentation we seek to test. The functionality that will be tested includes the following components:

#### 10.2.2.1. Mobile Client

- Mobile Graphic User Interface

- Communication for connecting PC

- Mike, Recording

- Cloud service

- Keyboard action control

- Mouse action control

#### 10.2.2.2. PC Client

- Communication for connecting mobile device

- Output voice streaming

- Cloud service

- Keyboard action control

- Mouse action control

* User interfaces to ensure usability, Accuracy, Reliability
* Authentication for security purpose

## 10.3. Major constraints

Several constraints that impact our testing plan include:

* Lack of familiarity with testing and verification techniques by group members
* Lack of adequate time for testing and implementation
* Lack of ability to test all critical pieces of the Dynamic presentation before the final deadline of project

## 10.4 Test Plan

### 10.4.1. Software to be tested

The Dynamic presentation Java-based device-driven application

### 10.4.2. Testing Strategy

#### 10.4.2.1. Unit testing

Our unit testing strategy for the Dynamic presentation will be testing the critical components required for successful execution of the application and will include the following components to be tested:

* The implementation of the main Dynamic presentation class

#### 10.4.2.2. Integration testing

Our integration testing strategy will include taking the individual units and combining them into a whole. To accomplish this, we will look at each use case and link the individual unit tests together to test for interactions between units. For example, socket, voice recored, pc to device connection test.

### 10.4.3. Testing resources and staffing

Our testing resources include as follows:

* Hee jun is responsible for the overall testing plan and writing of all unit tests
* Chulryong and Dae wook and tae sun will then correct any issues that unit test brings up (Unit test failure)
* The testing plan itself is a resource that help keep us within guidelines
* The Aspect Java Eclipse plug-in
* Eclipse IDE
* Libraries: JMF, JRTP
* Java Runtime
* Android SDK 2.3.3.

## 10.5. Test metrics

Our testing metrics will include data from regression testing of components to determine relationships between any errors implemented in code and the frequency of those errors. These metrics will be analyzed using common testing. Common statistical packages will be employed to perform these metrics. The metrics will then be analyzed and documented in a log that is kept along with our testing log.

## 10.6. Testing tools and environment

The testing tools and environment used for testing the Dynamic presentation will be listed as follows:

* Our main integrated development environment will be Eclipse 3.6
* Java SDK version 1.6
* Android SDK version 2.3.3.
* Notebooks for logging test results (A template for this aspect of the testing plan will be used)
* Laptop computer or any PC workstation

## 10.7. Test Schedule

Because of time constraints in finishing our project by the end of term, unit tests will be run continuously on a daily basis within a testing suite. As new functionality is added, new tests will be added to our testing suite as needed. The test of the function which added an individual does. The entire test of a project Hee jun does.

It is the ultimate goal to finish a test for implementing Final.

## 10.8. Unit Test

Our general procedure for testing the Dynamic Presentation is listed in the next set of bullet points. This procedure will give us a systematic way to test and validate the Dynamic Presentation application:

### 10.8.1. Testing procedure:

We will try to separate the developers from the testers so that the developers who developed the code do not perform the tests.

The tests are written before integration in Eclipse 3.7. As follow any of the unit tests is tested by each feature. And then we will test each usecase by feature.

### 10.8.2. DeviceConnectionManager Test

* **connect with valid address**
* **connect with invalid address**
* **connect with valid credential**
* **connect with invalid credential**
* **connect when networks available**
* **connect when networks unavailable**
* **sendMouse when networks available**
* **sendMouse when networks unavailable**
* **sendKeyboard when networks available**
* **sendKeyboard when networks unavailable**
* **transferVoiceInit when networks available**
* **transferVoiceInit when networks unavailable**
* **sendVoice when when networks available**
* **sendVoice when networks unavailable**
* **transferVoiceTerm when networks available**
* **transferVoiceTerm when networks unavailable**
* **transferVoiceTerm when session is keeped**
* **transferVoiceTerm when session is not keeped**
* **sendDocument when networks available**
* **sendDocument when networks unavailable**
* **keepConnection when disconnected**
* **restoreConnection when timeout event occured**

### 10.8.3. ConnectionActivity Test

* **inputInfo with valid credential type**
* **inputInfo with invalid credential type**
* **inputInfo with valid address form**
* **inputInfo with invalid address form**
* **displayStatus with valid status**
* **displayStatus with invalid status**
* **accept(retry) when networks available**
* **accept(retry) when networks unavailable**

### 10.8.4. MousePointerbyTouchpadActivity Test

* **touchpadMoveActiion when device moved**

### 10.8.5. MouseClickActivity Test

* **response immediatly**
* **leftClickAction when called**
* **rightClickAction when called**

### 10.8.6. MousePointerbyMotionActivity Test

* **motionMoveAction when touch event occured**

### 10.8.7. KeyboardActionActivity Test

* **response immediatly**
* **leftAction when called**
* **rightAction when called**

### 10.8.8. SoundRecordActivity Test

* **askContinue when called in pause status**

### 10.8.9. RecordedSoundPlayActivity Test

* **response immediatly**
* **notify with no file status**
* **notify with cannot play status**

### 10.8.10. SoundTransferActivity Test

* **response immediatly**

### 10.8.11. DeviceSoundActionManager Test

* **startRecord when called**
* **initRecord with available device status**
* **initRecord with unavailable device status**
* **pauseRecord when called**
* **stopRecord when called**
* **saveRecord when called with available store capacity**
* **saveRecord when called with unavailable store capacity**
* **playRecordedStart with available device status**
* **playRecordedStart with unavilable device status**
* **playRecordedInit with valid sound data**
* **playRecordedInit with invalid sound data**
* **transferVoiceInit when called**
* **transferVoiceTerm when called with session keeped**
* **transferVoiceTerm when called with session disconnected**

### 10.8.12. DocumentTransferActivity Test

* **response immediatly**
* **notify with no file status**
* **confirmTransfer with valid file**
* **accept when called**
* **deny when called**
* **cancelTransfer when called by deny**
* **displayProgress with normal transfer**

### 10.8.13. PCConnectionManager Test

* **run(create socket) with available PC resource**
* **run(create socket) with unavailable PC resource**
* **verifyCredential with valid credential**
* **verifyCredential with invalid credential**
* **keepConnection when network is disconnected**
* **restoreConnection when timeout event occured**

### 10.8.14. MouseActionManager Test

* **clickMouse with valid data**
* **moveMouse with valid data**

### 10.8.15. KeyboardActionManager Test

* **keyPress with valid data**

### 10.8.16. SoundActionManager Test

* **transferVoiceInit when session created**
* **playOutput with available PC resource**

### 10.8.17. TransferManager Test

* **receiveDocument when normally stored**
* **execDocument when normally execute file**

## 10.9. Integration / System Tests

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 1 | |
| **Test Case Name :** | | PC - Process Connecting login | |
| **Functionality Tested :** | | PC - User connect | |
| **Test Type :** | | Positive | |
| **Precondition :** | | Pc is operated, and it is state. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | A user clicks an icon in pc. | | A user can get from information of pc. |
| 2 | A user confirms ip information at pc. | | User confirms ip information of PC. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 2 | |
| **Test Case Name :** | | PC - Process Connecting login | |
| **Functionality Tested :** | | PC - User connect | |
| **Test Type :** | | Negative | |
| **Precondition :** | | Pc is operated, and it is state. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | A user clicks an icon in pc. | | A user can get from information of pc. |
| 2 | A user confirms ip information at pc. (Network connection is Stopped before confirming.) | | The user Network connection can read an error message that network did not. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 3 | |
| **Test Case Name :** | | PC - Process Connecting | |
| **Functionality Tested :** | | PC - User connect | |
| **Test Type :** | | Negative | |
| **Precondition :** | | Pc is operated, and it is state. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | A user clicks an icon in pc. | | A user can get from information of pc. |
| 2 | A user confirms ip information at pc. | | A user can read an error message that access became discontinuance. You can confirm the button which there is of to cancel the button which there is of what it reconnects after him and connection. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 4 | |
| **Test Case Name :** | | PC - Process Connecting logout | |
| **Functionality Tested :** | | PC - User connect logout | |
| **Test Type :** | | Positive | |
| **Precondition :** | | The network is connected with the PC is in the state. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | The user clicked the logout button and initiates a logout process. | | User is successfully logged out of the system and is directed to the Window.  User does not have access to perform any action. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 5 | |
| **Test Case Name :** | | Device - Process Connecting login | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Positive | |
| **Precondition :** | | Ip of the PC should be aware of the information. Device must be operational. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User clicks a Connect button of a Device screen. | | Device screen, showing enter the ip address of pc form. |
| 2 | User inputs PC ip information to input window, and you click "a Confirm" "button. | | PC compares information to get from users with own information. Connection normally worked at Device screens, and you inform through messages. And continuously shows state through icon colors connection. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 6 | |
| **Test Case Name :** | | Device - Process Connecting login | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Negative | |
| **Precondition :** | | Ip of the PC should be aware of the information. Device must be operational. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User clicks a Connect button of a Device screen. | | Device screen, showing enter the ip address of pc form. |
| 2 | User inputs the information that PC ip is wrong to input window, and you click "a Confirm" "button. | | "Ip address again, make sure" show an error message, represents a form that can be re-input |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 7 | |
| **Test Case Name :** | | Device - Process Connecting login | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Positive | |
| **Precondition :** | | Ip of the PC should be aware of the information. Device must be operational. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User clicks a Connect button of a Device screen. | | Device screen, showing enter the ip address of pc form. |
| 2 | User inputs PC ip information to input window, and you click "a Confirm" "button. | | Device for the network connection is lost "Network Please check your" error message shows. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 8 | |
| **Test Case Name :** | | Device - Process Connecting logout | |
| **Functionality Tested :** | | Device - User connect logout | |
| **Test Type :** | | Positive | |
| **Precondition :** | | Ip of the PC should be aware of the information. Device must be operational. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User press an end button of a device phone. | | User is successfully logged out of the system and is directed to the Device.  User does not have access to perform any action. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 9 | |
| **Test Case Name :** | | Device – Move mouse using touchpad | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Positive | |
| **Precondition :** | | Device must be connected with Pc. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | You move to courses User draws a finger to a Device screen, and to want. | | Pc shows a movement of a mouse pointer to the courses that user wants. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 10 | |
| **Test Case Name :** | | Device – Move mouse using touchpad | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Negative | |
| **Precondition :** | | Device must be connected with Pc. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | You move to courses User draws a finger to a Device screen, and to want. | | Device for the network connection is lost "Network Please check your" error message shows.  The screen below shows the reconnect button and a Cancel button. |
| 2 | User reconnects at Device screens, and you press a button. | | Device that allows you to enter information in the form of pc will appear. |
| 3 | User inputs PC ip information to input window, and you click "a Confirm" "button. | | Device is connected, and Pc. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 11 | |
| **Test Case Name :** | | Device – Move mouse using touchpad | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Negative | |
| **Precondition :** | | Device must be connected with Pc. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | You move to courses User draws a finger to a Device screen, and to want. | | Device for the network connection is lost "Network Please check your" error message shows.  The screen below shows the reconnect button and a Cancel button. |
| 2 | User presses a cancellation button at Device screens. | | A program is finished. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 12 | |
| **Test Case Name :** | | Device – Click mouse using touchpad | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Positive | |
| **Precondition :** | | Device must be connected with Pc. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User must click the left button in the Device. | | You show by the effect that clicked a button on the left of mouse at Pc. |
| 2 | User must click the right button in the Device. | | You show by the effect that clicked a button on the right of mouse at Pc. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 13 | |
| **Test Case Name :** | | Device – Click mouse using touchpad | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Negative | |
| **Precondition :** | | Device must be connected with Pc. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User must click the left button in the Device. | | Device for the network connection is lost “Network Please check your” error message shows.  You reconnect to the following screen, and you show button and cancellation button. |
| 2 | User reconnects at Device screens, and you press a button. | | Device that allows you to enter information in the form of pc will appear. |
| 3 | User inputs PC ip information to input window, and you click “a Confirm” “button. | | Device is connected, and Pc. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 14 | |
| **Test Case Name :** | | Device – Click mouse using touchpad | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Negative | |
| **Precondition :** | | Device must be connected with Pc. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User must click the right button in the Device. | | Device for the network connection is lost “Network Please check your” error message shows.  You reconnect to the following screen, and you show button and cancellation button. |
| 2 | User reconnects at Device screens, and you press a button. | | Device that allows you to enter information in the form of pc will appear. |
| 3 | User inputs PC ip information to input window, and you click “a Confirm” “button. | | Device is connected, and Pc. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 15 | |
| **Test Case Name :** | | Device – Click mouse using touchpad | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Negative | |
| **Precondition :** | | Device must be connected with Pc. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User must click the left button in the Device. | | Device for the network connection is lost “Network Please check your” error message shows.  You reconnect to the following screen, and you show button and cancellation button. |
| 2 | User reconnects at Device screens, and you press a button. | | Device that allows you to enter information in the form of pc will appear. |
| 3 | User presses a cancellation button at Device screens. | | A program is finished. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 16 | |
| **Test Case Name :** | | Device – Move mouse pointer using motion | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Positive | |
| **Precondition :** | | Device must be connected with Pc.  Device must operate in the acceleration sensor. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User clicks a button of an arrow shape at Device. | | Pc of the device is ready to represent the movement. |
| 2 | User moves in Device to courses to want. | | PC moves a mouse pointer of pc to the courses that Device moved in. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 17 | |
| **Test Case Name :** | | Device – Move mouse pointer using motion | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Negative | |
| **Precondition :** | | Device must be connected with Pc.  Device must operate in the acceleration sensor. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User clicks a button of an arrow shape at Device. | | Device for the network connection is lost “Network Please check your” error message shows.  You reconnect to the following screen, and you show button and cancellation button. |
| 2 | User reconnects at Device screens, and you press a button. | | Device that allows you to enter information in the form of pc will appear. |
| 3 | User inputs PC ip information to input window, and you click “a Confirm” “button. | | Device is connected, and Pc. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 18 | |
| **Test Case Name :** | | Device – Move mouse pointer using motion | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Negative | |
| **Precondition :** | | Device must be connected with Pc.  Device must operate in the acceleration sensor. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User clicks a button of an arrow shape at Device. | | Device for the network connection is lost “Network Please check your” error message shows.  You reconnect to the following screen, and you show button and cancellation button. |
| 2 | User presses a cancellation button at Device screens. | | A program is finished. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 19 | |
| **Test Case Name :** | | Device – Move mouse pointer using motion stop | |
| **Functionality Tested :** | | Device - User connect stop | |
| **Test Type :** | | Positive | |
| **Precondition :** | | Device must be connected with Pc.  Device must operate in the acceleration sensor. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | The user clicked the motion button and initiates a stop the motion. | | User does not have access to perform the motion action. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 20 | |
| **Test Case Name :** | | Device – Control keyboard | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Positive | |
| **Precondition :** | | Device must be connected with Pc. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User click the arrow button of the device to the left. | | Pc lets you perform an arrow event on the left of keyboard Ui. |
| 2 | User click the arrow button of the device to the right. | | Pc lets you perform an arrow event on the right of keyboard Ui. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 21 | |
| **Test Case Name :** | | Device – Control keyboard | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Negative | |
| **Precondition :** | | Device must be connected with Pc. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User click the arrow button of the device to the left. | | Device for the network connection is lost “Network Please check your” error message shows.  You reconnect to the following screen, and you show button and cancellation button. |
| 2 | User reconnects at Device screens, and you press a button. | | Device that allows you to enter information in the form of pc will appear. |
| 3 | User inputs PC ip information to input window, and you click “a Confirm” “button. | | Device is connected, and Pc. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 22 | |
| **Test Case Name :** | | Device – Control keyboard | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Negative | |
| **Precondition :** | | Device must be connected with Pc. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User click the arrow button of the device to the left. | | Device for the network connection is lost “Network Please check your” error message shows.  You reconnect to the following screen, and you show button and cancellation button. |
| 2 | User presses a cancellation button at Device screens. | | A program is finished. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 23 | |
| **Test Case Name :** | | Device – Process Record Sound | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Positive | |
| **Precondition :** | | Device must be connected with Pc.  Device must have space to store the file in.  Device has an audio input device. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User selects a Record button at Device. | | Voice recording begin, and Device informs. |
| 2 | User inputs a voice to Device. | | You display a voice of user at pc. |
| 3 | User selects a voice recording button. | | A button changes to pause, and voice recording stops. |
| 4 | User selects a stop button. | | A voice of a user is recorded. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 24 | |
| **Test Case Name :** | | Device – Process Record Sound | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Negative | |
| **Precondition :** | | Device must be connected with Pc.  Device must have space to store the file in.  Device has an audio input device. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User selects a Record button at Device. | | Device for the network connection is lost “Network Please check your” error message shows.  You reconnect to the following screen, and you show button and cancellation button. |
| 2 | User reconnects at Device screens, and you press a button. | | Device that allows you to enter information in the form of pc will appear. |
| 3 | User inputs PC ip information to input window, and you click “a Confirm” “button. | | Device is connected, and Pc. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 25 | |
| **Test Case Name :** | | Device – Process Record Sound | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Negative | |
| **Precondition :** | | Device must be connected with Pc.  Device must have space to store the file in.  Device has an audio input device. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User selects a Record button at Device. | | Voice recording bigin, and Device informs. |
| 2 | User must enter the voice on the Device.  Network connection stopped during Play. | | Device for the network connection is lost “Network Please check your” error message shows.  You reconnect to the following screen, and you show button and cancellation button. |
| 3 | User reconnects at Device screens, and you press a button. | | Device that allows you to enter information in the form of pc will appear. |
| 4 | User inputs PC ip information to input window, and you click “a Confirm” “button. | | Device is connected, and Pc. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 26 | |
| **Test Case Name :** | | Device – Process Record Sound | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Negative | |
| **Precondition :** | | Device must be connected with Pc.  Device must have space to store the file in.  Device has an audio input device. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User selects a Record button at Device. | | Device for the network connection is lost “Network Please check your” error message shows.  You reconnect to the following screen, and you show button and cancellation button. |
| 2 | User presses a cancellation button at Device screens. | | A program is finished. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 27 | |
| **Test Case Name :** | | Device – Process Play Sound | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Positive | |
| **Precondition :** | | Device must be connected with Pc. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User clicks a sound play button at Device. | | Device shows voice list files recorded. |
| 2 | User selects a voice file recorded to Device. | | Device makes the user has chosen to play the file. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 28 | |
| **Test Case Name :** | | Device – Process Play Sound | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Negative | |
| **Precondition :** | | Device must be connected with Pc. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User clicks a sound play button at Device. | | Device shows voice list files recorded. |
| 2 | User selects a voice file recorded to Device.( If you do not have a recorded voice) | | Device informs fact that there was not a voice recorded. |
| 3 | User can watch a basic screen of a program through Device. | | You can select the action that User wants at Device. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 29 | |
| **Test Case Name :** | | Device – Process Play Sound | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Negative | |
| **Precondition :** | | Device must be connected with Pc. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User clicks a sound play button at Device. | | Device for the network connection is lost “Network Please check your” error message shows.  You reconnect to the following screen, and you show button and cancellation button. |
| 3 | User inputs PC ip information to input window, and you click “a Confirm” “button. | | Device is connected, and Pc. |
| 2 | User reconnects at Device screens, and you press a button. | | Device that allows you to enter information in the form of pc will appear. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 30 | |
| **Test Case Name :** | | Device – Process Play Sound | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Negative | |
| **Precondition :** | | Device must be connected with Pc. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User clicks a sound play button at Device. | | Device for the network connection is lost “Network Please check your” error message shows.  You reconnect to the following screen, and you show button and cancellation button. |
| 2 | User presses a cancellation button at Device screens. | | A program is finished. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 31 | |
| **Test Case Name :** | | Device – Transfer document | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Positive | |
| **Precondition :** | | Device must be connected with Pc. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User clicks a Document button at Device. | | Device shows the file list that transmission is possible to User. |
| 2 | User selects a file to want. | | Device shows the file which a user selected.  And you ask whether or not the file correct. |
| 3 | User shall determine the fit of the selected file. | | Device shows the file transfer process.  System selected by the user executes the file. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 32 | |
| **Test Case Name :** | | Device – Transfer document | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Negative | |
| **Precondition :** | | Device must be connected with Pc. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User clicks a Document button at Device. | | Device shows the file list that transmission is possible to User. |
| 2 | User selects a file to want. (If a user wants other file.) | | A user cancels file transmission. |
| 3 | User can watch a basic screen of a program through Device. | | You can select the action that User wants at Device. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 33 | |
| **Test Case Name :** | | Device – Transfer document | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Negative | |
| **Precondition :** | | Device must be connected with Pc. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User clicks a Document button at Device. | | Device for the network connection is lost “Network Please check your” error message shows.  You reconnect to the following screen, and you show button and cancellation button. |
| 3 | User inputs PC ip information to input window, and you click “a Confirm” “button. | | Device is connected, and Pc. |
| 2 | User reconnects at Device screens, and you press a button. | | Device that allows you to enter information in the form of pc will appear. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 34 | |
| **Test Case Name :** | | Device – Transfer document | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Negative | |
| **Precondition :** | | Device must be connected with Pc. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User clicks a Document button at Device. | | Device shows the file list that transmission is possible to User. |
| 2 | User selects a file to want. (If the file does not match) | | A user cancels file transmission. |
| 3 | User can watch a basic screen of a program through Device. | | You can select the action that User wants at Device. |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST INFORMATION** | | | |
| **Test Case Number :** | | 36 | |
| **Test Case Name :** | | Device – Transfer document | |
| **Functionality Tested :** | | Device - User connect | |
| **Test Type :** | | Negative | |
| **Precondition :** | | Device must be connected with Pc. | |
| **TEST STEPS** | | | |
| **#** | **Step** | | **Expected Results** |
| 1 | User clicks a Document button at Device. | | Device for the network connection is lost “Network Please check your” error message shows.  You reconnect to the following screen, and you show button and cancellation button. |
| 2 | User presses a cancellation button at Device screens. | | A program is finished. |

## 10.10. Test Result

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case Number | Test Case Description | Result(Pass/Fail) | Note |
| 1 | Connecting(positive) | passed |  |
| 2 | Connecting(negative-network doesn’t work) | passed |  |
| 3 | Connecting(negative-wrong code) | passed |  |
| 4 | Connecting(negative-wrong address) | passed |  |
| 5 | Move by touchpad(positive) | passed |  |
| 6 | Move by touchpad(negative-network doesn't work) | passed |  |
| 7 | Click mouse(positive) | passed |  |
| 8 | Click mouse(negative- network doesn’t work) | passed |  |
| 9 | Move by motion(positive) | passed |  |
| 10 | Move by motion(negative – network doesn’t work) | passed |  |
| 11 | Move by motion terminate(positive) | passed |  |
| 12 | Control keyboard(positive) | passed |  |
| 13 | Control keyboard(negative – network doesn’t work) | passed |  |
| 14 | Record(positive) | passed |  |
| 15 | Play recorded(positive) | passed |  |
| 16 | Transfer document(positive) | passed |  |
| 17 | Transfer document(negative – network doesn’t work) | passed |  |
| 18 | Transfer voice(positive) | passed |  |
| 19 | Transfer voice(negative - network doesn't work) | passed |  |

Figure 49: Test Result Table

# 11. Appendices

## 11.1. Task Distribution

오희준: Director / Network related main programming, PC & Mobile sub programming..

신대욱: Writer / PC main programming, network sub programming.

박태선: Coder / Mobile main programming, network sub programming.

박철룡: Designer / ui & graphic design, scheduling, testing.

## 11.2. Project Schedule

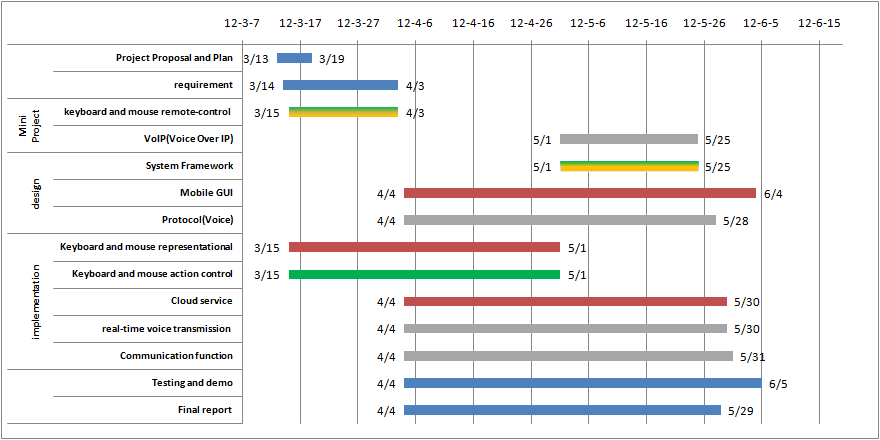


Figure 50: Project Status