

COMPRESSED DATA STORAGE ON CENTRALIZED HARD DISK OVER WI-FI



Faiqa Shakeel

Hamza Mehmood

Osama Jawad

Faizan Ahmed

Supervised By

Mr. Imran Asif

*Submitted for the partial fulfillment of BS Software Engineering degree to
the Faculty of Engineering & CS*

**NATIONAL UNIVERSITY OF MODERN LANGUAGES
ISLAMABAD**

JANUARY, 2018

ABSTRACT

Compressed Data Storage on Centralized Hard-Disk over Wi-Fi provides a way of storing data on storage devices using WLAN. The system provides platform for multiple users to store and access data from a single storage device. This system offers few utility programs such as share data, screen sharing, remote desktop, and cloud backup.

This system provides storage device and application for desktop and mobile, to store, access, update, delete, compress-decompress, stream video files, cloud access, share files and has extendable storage feature. Existing devices are Seagate and SanDisk Connect they have a web interface to store, access and delete files. They does not have compress-decompress, update, stream videos, cloud access, share files and extendable storage features. Another existing system is Team Viewer that allows remote access to other computer, but works over internet only. This system provides remote access to other computer over WLAN.

For the development of desktop application C# is used, for android application JAVA is used. Server side development is done by using python, HTML bootstrap, PHP and JavaScript, for designing the web interface. Compilers that are used for implementation purpose are Visual Studio and Android Studio.

System has been tested precisely to ensure the proper working of all the functionalities. Total twelve tests are conducted based on seven use cases, out of which 80% tests were successful. Client side testing is performed in visual studio while the server side testing is done in chromium developer tools.

CERTIFICATE

Dated: _____

Final Approval

It is certified that project report titled ‘Compressed Data Storage on Centralized Hard Disk over Wi-Fi’ submitted by Hamza Mehmood, Faiqa Shakeel, Faizan Ahmed and Osama Jawad for the partial fulfillment of the requirement of “Bachelor’s Degree in Software Engineering” is approved.

COMMITTEE

A/Dean Engineering & CS:	Signature:	_____
--------------------------	------------	-------

Col (R) Engr. Jamil Ahmad Zia		
Head Engineering:	Signature:	_____

Waris Ali		
Head Project Committee:	Signature:	_____

Supervisor Name		
Imran Asif	Signature:	_____

CHAPTER 1

INTRODUCTION

1.0 Introduction

“Compressed Data Storage on Centralized Hard-Disk over Wi-Fi” is a Desktop and android based application that will allow the users to exchange data, store data on storage device, remotely control desktop, play multiplayer games & share Screens through WLAN connection using Raspberry pi 3. In this project our idea is to store the data in compressed form which will save the storage for the user. The desktop and android application interface will be used to store data on storage device or to share data with the other device. Desktop interface also allows us to remotely control the desktop and share the computer’s screen. It will also provide us cloud access of important data, so it can be accessed from anywhere.

1.1 Problem Domain

There are some desktop applications which provide facility of screen sharing PC to PC only. These products only store but not share data among users. They are not providing the features like remote desktop, compression, decompression, screen sharing and gaming panel. Existing devices cannot share data between PC to PC/Mobile. Storage capacity is not extendable, and they also don’t have any cloud access or back-up.

1.2 Proposed System

Centralized Hard-Disk Storage over Wi-Fi will form a Wi-Fi connection between the devices and allow the users to store & share data with the other device (i.e. mobile, computer). The data stored in the storage device can be compressed so it will save a lot of space. Anyone who is connected to the device can access to the stored data and users can play multiplayer games on single server and watch videos from the storage device directly without copying it to personal storage. The features of application are listed below:

1.2.1 File Transfer:

Between PC to PC/Android. Between Data Storage Device to PC/Android.

1.2.2 Centralized Data Storage:

Allow the user to store the data on the storage device, as it is centralized so it will be accessible for all the users who are allowed.

1.2.3 Gaming Point:

Provide a panel for multiplayer games which will allow the user to join the online multiplayer game at the present server.

1.2.4 Screen Sharing:

User can share his screen with the other users.

1.2.5 Video Streaming:

Users can Stream videos from the storage device without copying it.

1.2.6 Compression & De-compression:

The data stored on the storage device can be in compressed and decompressed on the server side.

1.2.7 Android/Desktop Application:

Desktop application provides interface for storing data, sharing data, remote control, sharing screen and for cloud access. Android application provides interface to store data in the storage device, share data, access screen and cloud access of the storage device.

1.2.8 Remote Desktop:

User can remotely control the other device and use mouse, keyboard to perform different operations.

1.2.9 Cloud Back-up:

User can also have cloud access of the storage device.

1.2.1 Aims and Objectives

We will deliver “Complete Android/Desktop Application source code”. Currently we are working on our Desktop Application in which we have done with ‘File Transfer’ (PC-PC & PC-Raspberry pi), Screen Sharing and Video Streaming. Along with this, the contents created during development process, complete working executable file as an application and project documentation in the form of project report will also be delivered.

1.2.1.1 Accuracy

Accuracy is the degree of closeness. If we talk about the project accuracy, it involves the wireless data storage. The system should be accurate enough that allow the user to store the data on the storage device, as it is centralized so it will be accessible for all the users who are allowed.

1.2.1.2 Reliability

Reliability tell us about that software can perform its specifications affording to continuous performance. It will not be crash while using it. This reliability of a system involving both desktop and android applications. They are reliable as they work without crashing system.

1.2.1.3 Understandability

Understanding feature is how the user can be interpreted as a degree of degree. If the system contains a language or one way on which people are very difficult to know, the system is considered very complicated. People are not able to understand the system. The project involves the procedure, process or procedure of understanding of a system in which a person needs to complete the system to meet the system whose system determines how the person performs the entire process can be able to share data through sharing, and they understand and able to work the system.

1.2.2 Proposed System Features

The proposed system is a software based and hardware based Compressed Centralized Storage over Wi-Fi.

1.2.2.1 Raspberry Pi 3

There is a board credit card sized computer manufactured in the UK Raspberry Pi Foundation. Raspberry pi was released in February 2012 for the first time, and the latest model was released in February 2017. The Ram of Raspberry Pi 3 is 1 GB and Linux based operating System Raspbian. The CPU which it uses is 1.2GHz, there is also an Ethernet port, 4 USB Ports, HDMI port and audio input port.

The component of raspberry pi which involves File Transfer, Screen Sharing, Video Streaming, Remote Desktop, centralized storage and Cloud Backup is done on the Raspberry Pi and cloud access respectively.

1.3 Development Methodology

Software development in software executive is a platform which can be used to plan, control the development procedure for an information system. This includes software development work in a separate phase, which includes activities with active planning and management commitment. Common methodologies include, waterfall, spiral development, rapid development, highly development and various types of mobile methods. The technique used for the development of the project can be an incremental model. In additional models, the complete requirement is split into different constructions. Multiple development pattern is here now in a cycle. The cycles are split into small, easier structured planned modules. Each component passes through certain requirements, design, execution and testing stages. An operating version of the program is developed through the first module. Each complete release of the component works in the last release adds. This technique continues before complete system is received.

1.4 Resource Requirement

Following resources are required for development:

1.4.1 Hardware Requirements

For testing and deployment of the application, we will need devices (mobile, tablet or laptop) with built in Wi-Fi device and for the host side we need Raspberry pi 3, which includes a router that creates connection between the smart devices and centralized storage.

1.4.2 Software Requirements

The programming language used for developing the application will be Java for Android in android studio and C# for Desktop Application Development. For the graphical content development, we will use Adobe Photoshop and Adobe After Effects. Following tools are used for development:

1.4.2.1 Visual Studio

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. Visual Studio uses Microsoft PC code development systems like Windows API, home windows paperwork, windows Presentation basis, home windows save and Microsoft Silverlight. It'll turn out each native code and code managed. Visual Studio includes a code editor helping IntelliSense (the code of completion aspect) further as code refactoring. The incorporated computer software works each as a source-stage laptop software and a machine-level computer program. It accepts plug-ins that decorate the practicality at nearly each level together with adding guide for deliver control systems like Subversion and including new tool units like editors and visible designers for domain-particular languages or device sets for opportunity elements of the PC code development lifecycle.

1.4.2.2 Android Studio

Android Studio is the official IDE for Android. Android Studio offers the speediest gadgets to constructing packages on every type of Android machine. Global-class code altering, troubleshooting, execution tooling, an adaptable shape framework, and a moment deliver framework all enable you to concentrate on constructing interesting and wonderful applications. We used android studio for development of android application for mobile phones.

1.4.2.3 Adobe Photoshop

The product Adobe Photoshop gives many pictures altering features to pixel-based pictures and in addition vector illustrations. It utilizes a layer-based altering framework that empowers picture creation and modifying with different overlays. We used Adobe Photoshop for creating the main logo and interface design of the android application and desktop application.

1.4.2.4 Adobe After Effects

The Adobe After consequences is an automated visible effect, movement illustrations, and compositing software created by way of Adobe systems and utilized as part of the after-era process of film making and animations. It additionally works as an exceptionally fundamental editorial manager, sound supervisor and media transcoder. We use Adobe After effects for creating the logo animation and intro video.

1.4.2.4 Ms Visio

Ms Visio is in package of Ms-Office and used for Vector Graphics Diagramming Application. Its first Application was first introduced in 1992. We used Ms-Visio for drawing Use Cases diagram.

1.5 Basics of the Report

The opening chapter of the report highlighted the purpose of project and the system requirements. This section includes the features of the system and methods used. The second chapter provides the background of the system, limits in the current system are mentioned and the whole passive needs are discussed. The third section describes the information of the software as defined in detail.

CHAPTER 2

EXISTING SYSTEMS

2.0 Introduction

The “Compressed Data Storage Centralized on Hard-Disk over Wi-Fi” will have many functionalities like file transfer between PC to PC/android and between data storage device to PC/Android. Allow the user to store the data on the storage device, as it is centralized so it will be accessible for all the users who are allowed. Users can Stream videos from the storage device without copying it through video streaming. The data stored on the storage device will be in compressed form and will be decompressed when it is extracted from the storage device. Screen sharing will allow user to share his screen with the other users. It also provides a panel for multiplayer games which will allow the user to join the online game at the present server. It will have Interfaces for both desktop and android will be provided to manipulate with the data. User can remotely control the other device and use mouse, keyboard and access data of the other computer. User can also create back-up of his/her important data with the help of cloud access.

2.1 Existing Work

There are several current systems presented in the market with restricted functionalities. The overview of such systems is stated as follows.

2.1.1 Seagate

Seagate wireless storage device allows you to store data on a storage device. It comes with the capacity of 500 gigabytes and 1TB. You can store and extract data on the storage device over the Wi-Fi. Provide the facility of backup but only for media applications [1]. Linux operating system is not supported by the Seagate. You need appropriate browser for compatibility with Linux version.

2.1.2 Team viewer

TeamViewer is a common piece of software used for Internet-based remote access and support. TeamViewer can connect to any PC or server, so you can remote control your partner's PC as if you were sitting right in front of it by using internet connection [2]. TeamViewer is installed with a proper installation procedure. To connect to another's device, TeamViewer must be running on both machines at same time. To install TeamViewer, you required administrator access. It does not work through proxy servers, and its business packages are very expensive to buy. You need to buy the team viewer for long sessions.

2.1.3 SanDisk Connect

The Connect Wireless Stick is a USB flash drive, which enables you to stream, access and transfer your files over 802.11n Wi-Fi [3]. You can also connect wirelessly to a computer but plugging it into a computer's USB port. Browser is used to explore the data present in the USB Stick. To access files stored on the stick, you need SanDisk Connect Drive app, available for Amazon devices.

2.2 Limitations of Existing Systems

There are some desktop applications which provide facility of screen sharing mobile to mobile & PC to PC only. These products only store and share data. They are not providing the features like Gaming panel, Compression, Screen sharing and gaming panel. Existing devices cannot share data between pc to pc/mobile and vice versa. It also doesn't have any cloud access or back-up.

2.2.1 Comparison of Existing System vs proposed System

Existing System	Proposed System
Seagate	WIDA
It provides fixed storage.	WI-DA Provides extendable storage.
There is no application.	It provides application for both desktop and android.
It is not providing the facility of cloud backup.	It provides the facility of cloud backup.
It cannot share data between devices.	It allows the user to share data between devices.
Team Viewer	WIDA
It is over the internet.	It is over the WLAN, no need of internet.
Paid for the long sessions.	Free for the long sessions.
SanDisk Connect	WIDA
It provides fixed storage.	WI-DA Provide extendable storage.
There is no application it is browser based.	It provides application for both desktop and android.
It is not providing the facility of cloud Access.	It provides the facility of cloud Access.
It offers no compression.	It provides both compression and De-compression.

2.3 Overview of Proposed System

It is a Desktop and android based application which allow the users to share and store data on storage available on Raspberry pi. User can also play games & share Screens through WLAN connection using access point hosted using raspberry pi Wi-Fi. Data is stored in compressed form to save the storage for the user. The desktop and android interface will be used to store data on storage device or to share with the other device. Cloud access option is also provided to user to store important data, so it can be accessed via internet.

2.4 Related Work

The project is combination of many functionalities like remote desktop, screen sharing, cloud access and compression decompression. Team viewer is providing the functionality of remote desktop and screen sharing. On the other hand, drop box is providing the functionality of cloud storage. But the drawback of drop box & team viewer is that it's on internet, it does not work on LAN.

2.5 Interface and Code Linkage

The interface contains a window in which user have different buttons which allows users to perform the desire operations like screen sharing, data sharing, etc. The interface is modest and simply clear to the user. The system interface includes the following functions:

2.5.1 Server-Side Module

Server module is made on the raspberry pi 3 which will receive the data from the devices via Wi-Fi and store it on the storage device. Web service will allow us to interact with the data stored on the storage device. The server will use certain protocols to share the data between the devices.

2.5.2 Client Module

On the client side the interface will establish the connection between the server module and client side, which will allow the user to stored data. Share screen or establish remote desktop connection with the other device without using internet. Same wise and android application will provide the facility to the user to use the services.

2.5.3 Connection Module

Connection will be on simple server IP address which will show all the device connected

to the system and allow to access the data using FTP server. The embedded Linux on raspberry pi 3 will provide the commands to create connection between the devices.

2.6 Functional Requirement

There will be some system requirements for different modules to work. These requirements are mentioned below:

2.6.1 Data storing

The interface allows the user to connect to the raspberry pi via Wi-Fi, Raspberry pi access the storage media and user accesses the storage drive to transfer the desired data to the storage device.

2.6.2 Data sharing

The interface allows the users to connect to the raspberry pi via Wi-Fi. Now user requests the desire user for permission to share the file and on the other hand the other user grants the permission and data will be shared between two users.

2.6.3 Screen sharing

The application allows the users to connect to the raspberry pi via Wi-Fi to perform screen sharing. Host user starts server and broadcast screen with user name and password. Other client users will check for server connection and connects to the available server. Only the authorized users can now access the host screen.

2.6.4 Video streaming

In video streaming user connects to the raspberry pi via Wi-Fi and Raspberry pi access the storage media. User accesses the video file and can stream the videos online without copying it into its own storage device which can save the copying time and storage space of the user.

2.6.5 Gaming panel

Provide a platform to the users for multiplayer games.

2.6.6 Remote desktop

In remote desktop user can connect the device to the raspberry pi via Wi-Fi and establish a connection between the desire devices and remotely control the other device i-e use mouse, keyboard and access the other computer to perform the desired task.

2.6.7 Cloud backup

This application allows users to create back-up of important data through cloud access of the storage.

2.7 Non-Functional Requirements

The Non-functional requirements of the system are the requirements that are not clearly explained by the user during the earlier stage. Developer considers their presence and priority within their application on the bases of various modules. Quality of the system is enhanced by focusing on these needs in development of the system.

2.7.1 Security

Security is the core feature of the proposed system. Data stored in the storage drive is not accessible by any unauthorized user because the Raspberry pi server is password protected. Only authorized users who have password can only join to screen sharing or gaming servers.

2.7.2 Reliability

Reliability is the failure-free operation of a system over a specific time within a quantified environment for a definite purpose. It suggests that failure rate is a low if the system is out there to be used for long time which suggests system is reliable. Proposed system is reliable because of the system ensures that it has cloud access of the personal data.

2.7.3 Usability

The interface of the system is very simple and designed in such a way that it is easy for the new user to interact with the interface with a simple design for storing data, sharing data, screen sharing, video streaming, gaming panel.

CHAPTER 3

SYSTEM MODELING

3.0 Introduction

In this chapter the use cases, description of use cases and sequence diagrams are explained. Use cases are the steps of interaction between the system and user in order to perform the desired task. User can be some external system or a normal person. For describing the sequence of processes in the system sequence diagrams are used for each functionality.

3.1 Use Case Diagram

Main use case diagram is described in the figure 3.1 on the next page. There are different features provide for user in system. The use case begins with menu which gives user to select operation among different operations, user can select share data option to share the data among the connected users, store data to store the data on the storage drive attached with the raspberry pi 3 server, remote control to remotely control the other devices using raspberry pi, share screen to share the screen with other users, game point operation provide the gaming access, stream video to stream videos from the storage drive attached with the raspberry pi server, manage cloud to create the back up of important data.

The use case contains all the tasks that the user performs on the system and all the required output that the system generates with which the user interacts. These use case helps to easily understand what are the features present in the system, that the user are interacting with. The user interacts with the interface in the device to perform operations using the server which is made on raspberry pi 3. All the users will be connected with the raspberry pi access point, which acts as a communication medium. A USB storage device is connected with the raspberry pi which stores data of the user. Raspberry pi contains Linux operating system which performs all the server-side operations.

Use case diagram in Figure 3.1, is the main use case which is showing the usage scenario of the menu which involves users as actors, raspberry pi as a server-side module. On the next page we can see the main use case diagram of a system in which all the features of a system showing the both client and server users interaction. To store data, access data, stream videos, and manage cloud user can select the corresponding option and then connect with server side module to perform the desired function. To share data among users, user will have to select share data option and connect with sever side module. For Access Desktop and Share Screen features users requires to take permission for establishing a connection between the two devices.

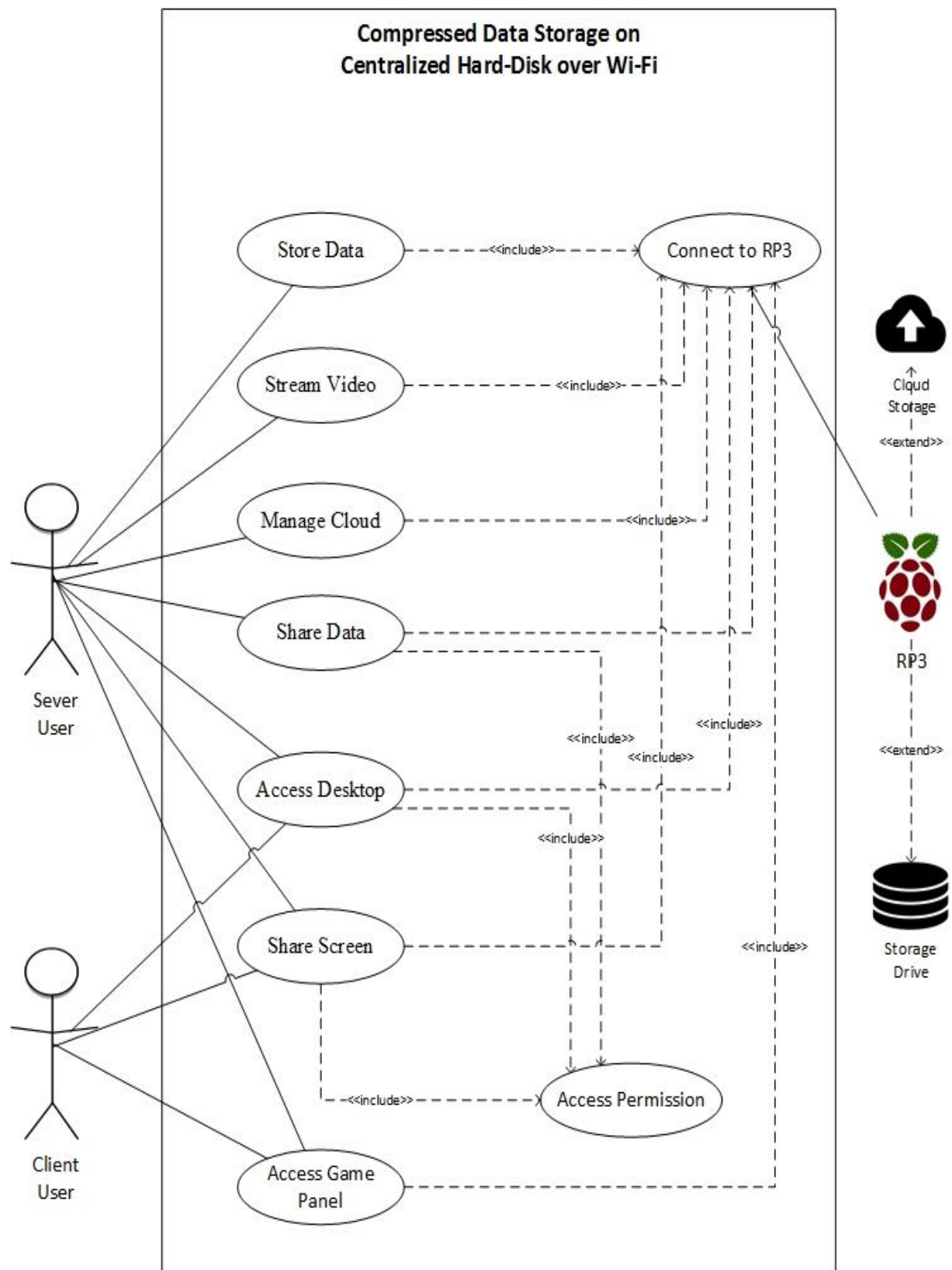


Figure 3.1 Main Use Case Diagram

3.1.1 Store Data Use Case

Figure 3.2 shows the use case scenario of the store data operation, after selecting the data store option from the main menu the system will take the user to data store menu where the system will connect with the raspberry pi server, identify the user and show user the data present on the USB drive connected with the raspberry pi. The user can store data as well retrieve data from the USB drive wirelessly via raspberry pi as a medium.

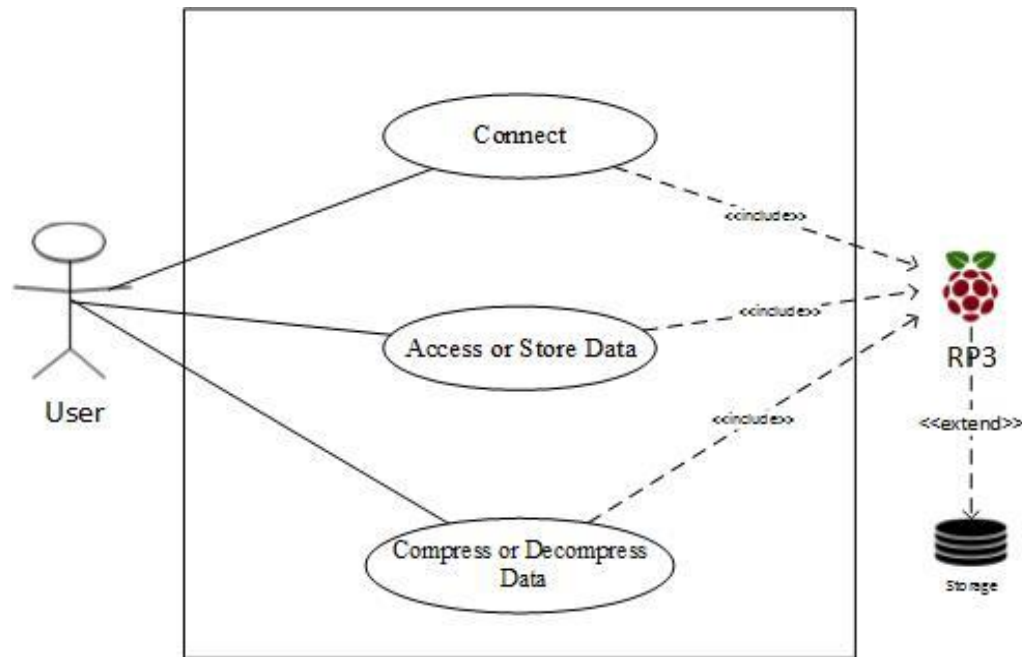


Figure 3.2 Store Data Use Case

Table 3.2 is describing the actors involved in the system, purpose of the system, overview of the system, post and pre-condition for the store data use case. In this scenario client user should be able to interact with the store data menu to transfer his data from device to the storage drive connected with the raspberry pi. Pre-condition for this use case is that the data from the drive which is connected with the raspberry pi should be visible in the interface of the user so that user can copy or paste data.

Table 3.1 Store Data Use Case

Use Case Id	UC-01
Use Case Name	Store Data
Requirement Reference	Requirement No 1
Actors	User and Interface
Purpose	To store the data
Overview	The client side user Interact with the system and select the operation to store/extract data

	from the connected storage device
Type	Primary and Essential
Pre-Condition	The data in the connected device should be accessible from the interface
Post-Condition	User should be able to transfer data to/from the connected device via the interface
Typical Course of event	
Actors Action	System Response
1. User selects the store data operation.	2. Raspberry pi 3 should establish the connection between the user interface and connected device.
	3. Display data of the connected device, on the interface.
4. User selects the data to upload/download from/to the connected device.	5. System should start responding according to the performed operation, by the user.
6. User selects data to delete the data.	7. System should delete the selected data.
8. User selects to compress/decompress the data.	9. System should compress/decompress the data.
Alternative flows	
2a. System is unable to establish connection.	
3a. System do not display the data.	
7a. System do not perform the user selected operation.	
8a. Data on the connected drive is not deleted.	
9a. System do not compress/decompress the data.	

3.1.2 Share Data Use Case

In Figure 3.3 the scenario of share data is shown. The user will connect with the raspberry pi, system will show him the available user from which the user will select to share his data with. The system will ask for the permission from other user to allow access. After accessing the permission, the system will create the connection between two devices and start transferring data.

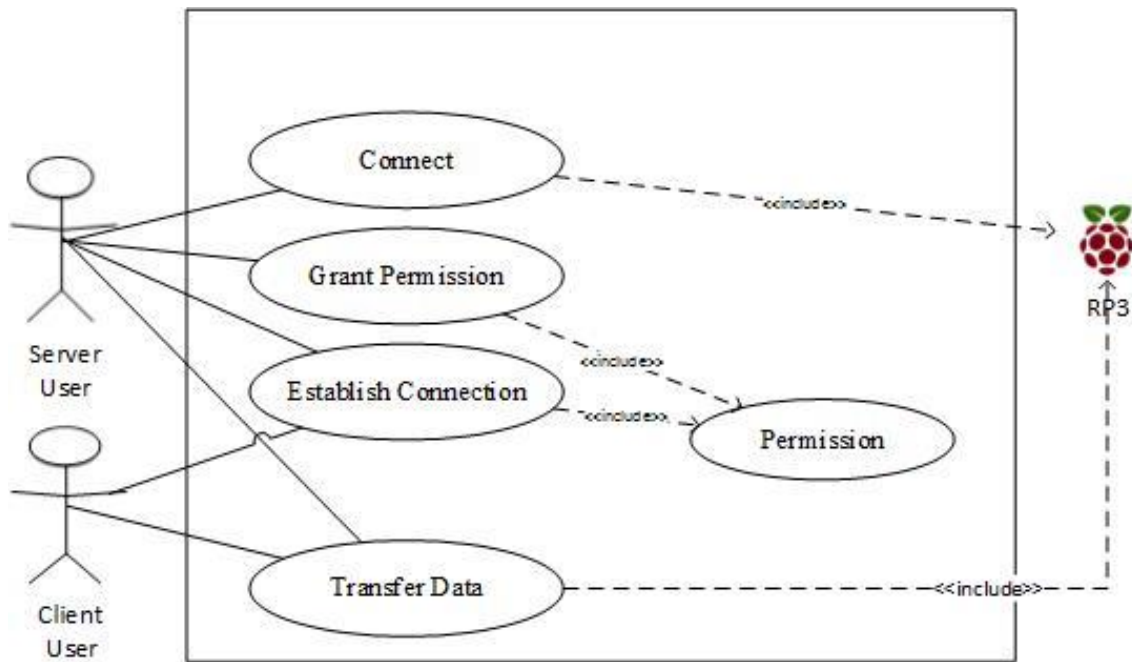


Figure 3.3 Share Data Use Case

Table 3.2 describes the actors, purpose, overview, post and pre-conditions for the share data use case. Pre-condition is that the connected devices should be visible so the user can select the device to share data with. After selecting the device, the system will ask for the permission and share data with the selected device.

Table 3.2 Share Data Use Case

Use Case Id	UC-02
Use Case Name	Share Data
Requirement Reference	Requirement No 2
Actors	Client, Server and Interface
Purpose	Share the data between users
Overview	Client user can connect with the other devices and share data with other connected devices
Type	Essential
Pre-Condition	Client user should be able to see share data interface and get a key

Post-Condition	Client user should be able to connect with other device using key and share data
Typical Course of event	
Actors Action	System Response
1. User selects the share data operation	2. System ask user to enter key.
3. User enter the key of client device.	4. System identify the user.
	5. System should establish the connection between the user's device and client's device.
6. User selects the data to send/receive from/to the client device.	7. System should start responding according to the selected operation, by the user.
Alternative flows	
4a. System is unable to identify the user.	
5a. System is unable to establish connection between user's device and client's device.	
8a. System do not perform the user selected operation.	

3.1.3 Access Desktop Use Case

In this use case the server user can control the client device remotely i.e. using the mouse and keyboard. The Figure 3.4 is showing the whole scenario for the remote desktop use case. In the start the user will connect with the raspberry pi and select the user to control desktop. After gaining the permission of the client.

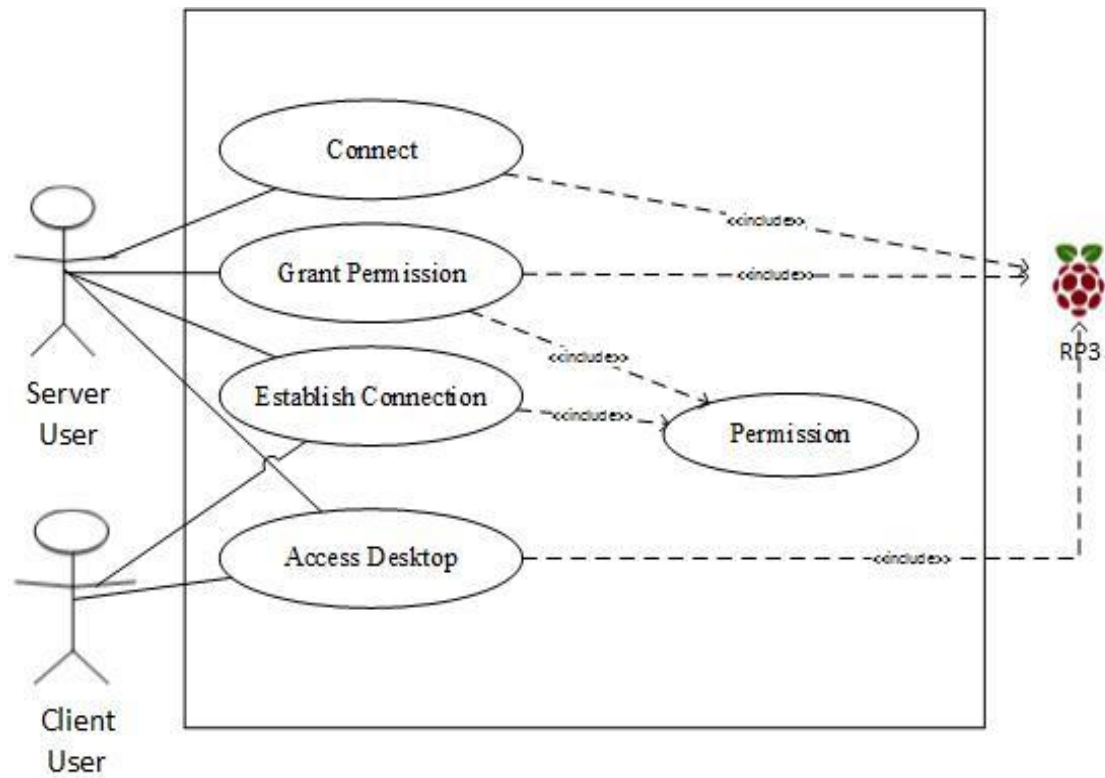


Figure 3.4 Access Desktop Use Case

Table 3.3 describes the purpose, overview, post and pre-conditions of remote desktop feature of the system, pre-condition is that user should be able to search for all the available devices on the local network, so user can connect with the device, post condition is user should get the permission and access the client's device remotely.

Table 3.3 Access Desktop Use Case

Use Case Id	UC-03
Use Case Name	Access Desktop
Requirement Reference	Requirement No 3
Actors	Client, Server and Interface
Purpose	Allow the user to access remotely with connected devices
Overview	Client user can connect with the other devices and remotely use the system
Type	Essential
Pre-Condition	The remote control interface should be visible, where user can enter an IP address.

Post-Condition	Client user should be able to remotely control after entering the IP address of server device.
Typical Course of event	
Actors Action	System Response
1. User selects the remote desktop operation.	
2. User enters the IP address of the server device.	3. System will ask for the permission to the selected device.
	4. System should establish the connection between the devices after identification.
5. User remotely uses the interface of selected device.	6. System should allow the user to use the interface.
Alternative flows	
4a. System do not allow access to the user after identification.	
4b. System allows access to without permission.	
6a. System do not allow the user to use interface remotely.	

3.1.4 Share Screen Use Case

In this use case user screen sharing scenario is shown, the server user broadcast the screen and client will access it by establishing a connection between client and server. Figure 3.5 is the case diagram of share screen feature.

Table 3.4 Share Screen Use Case

Use Case Id	UC-04
Use Case Name	Share Screen
Requirement Reference	Requirement No 4
Actors	Interface, Server and Client
Purpose	Allow the user share screen with users
Overview	Host user can connect share their screen with other users and client user can view the screen
Type	Essential
Pre-Condition	Client User should be able to see the interface to enter IP address of server device
Post-Condition	Client user should be view the screen of server device after entering the IP address
Typical Course of event	
Actors Action	System Response
1. User selects the remote desktop operation.	2. System should ask user to enter the IP address of server.
3. User enters the IP address.	4. System should ask password from the client device.
	5. System should start showing the server's screen in the client's interface, after authentication.
Alternative flows	
5a. System do not show screen to the client user after authentication.	
5a. System is not showing the server's screen.	

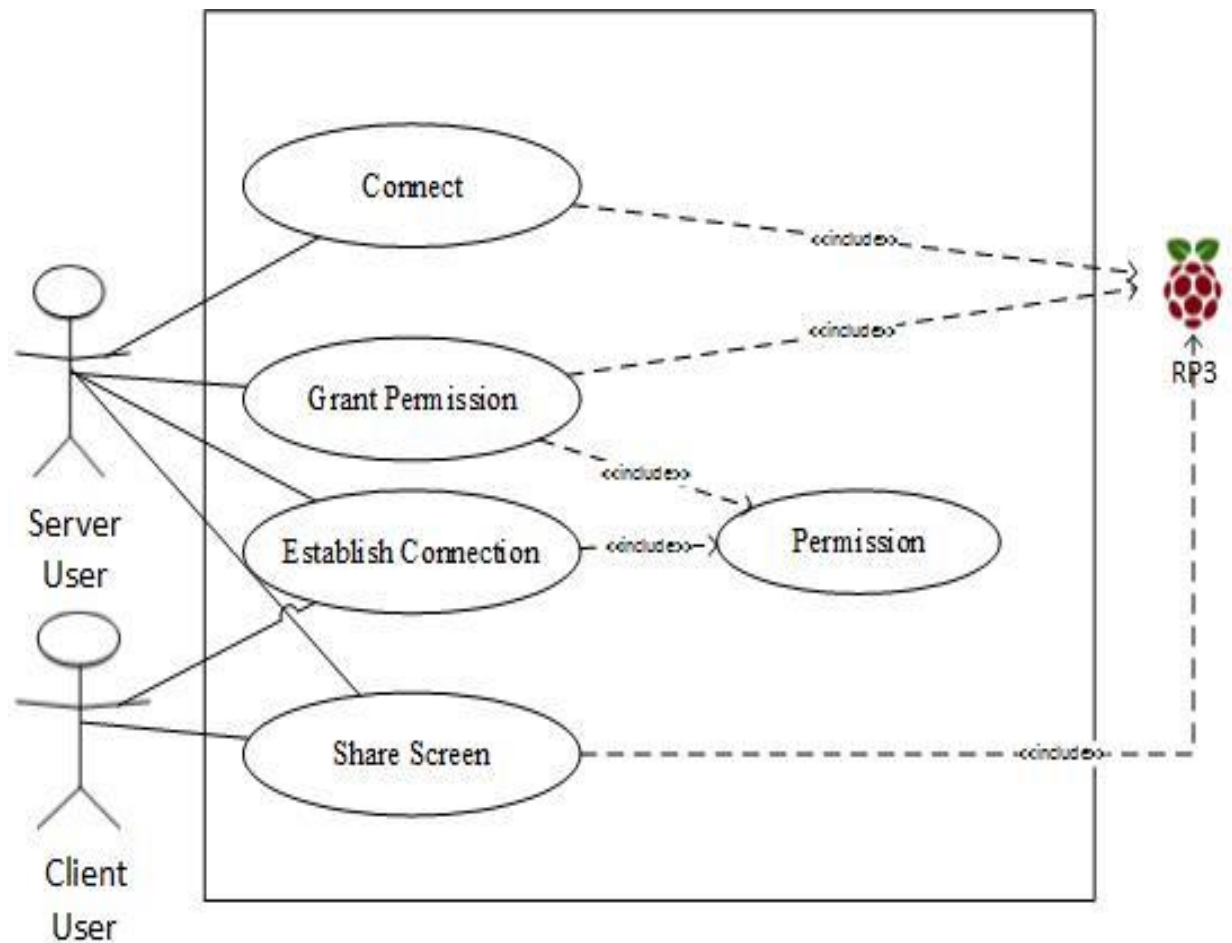


Figure 3.5 Share Screen Use Case

Table 3.4 describe the actors, overview, purpose, post- and pre-conditions of the screen sharing feature.

3.1.5 Stream Video Use Case

In this scenario the user is allowed to stream the videos from the storage drive connected with the raspberry pi directly without copying it to personal storage device. Firstly, user will connect with the raspberry pi, system will show user the list of video files available in the storage drive. User can select the video and watch it without saving it. Figure 3.6 is the use case diagram of video streaming feature.

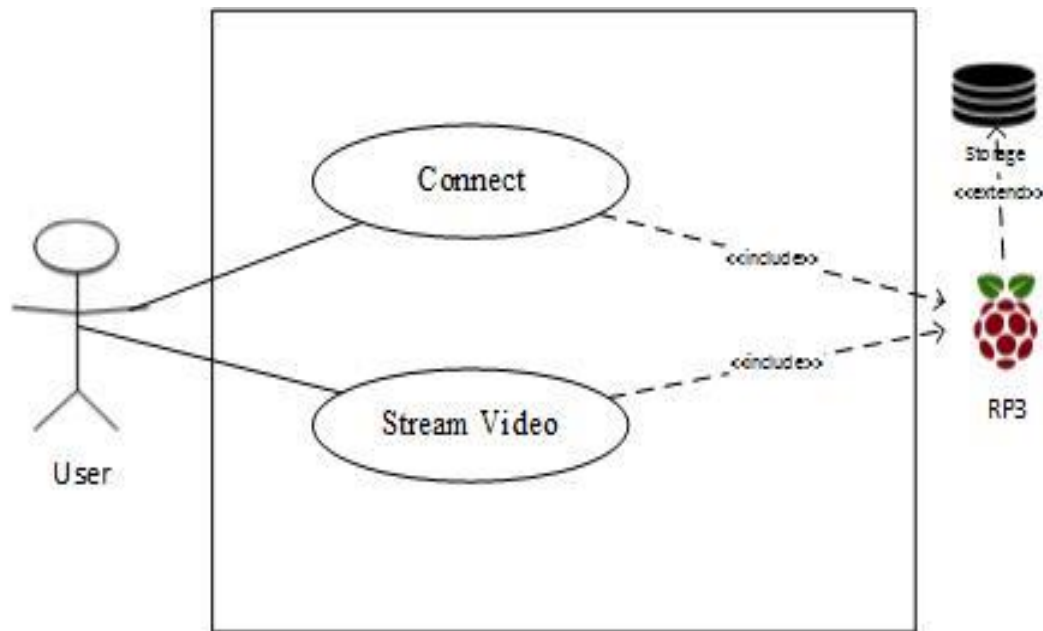


Figure 3.6 Stream Video Use Case

Table 3.5 describe the use case of stream video feature. Firstly, user will select a video to play, the system will play the video in interface without copying it into the user's system.

Table 3.5 Stream Video Use Case

Use Case Id	UC-05
Use Case Name	Stream Videos
Requirement Reference	Requirement No 5
Actors	Interface, user and server
Purpose	Allow the users to stream videos from the storage device
Overview	User can connect with RP3 and stream videos from the storage device connected to the RP3
Type	Essential
Pre-Condition	System should show all the video files present on the storage drive.
Post-Condition	Client user should be able to select and stream the desired video.
Typical Course of event	
Actors Action	System Response

1. Client user selects the stream video operation.	2. System should show the available video files on the user's interface and allow the user to select the video file.
3. Client user selects a video file	4. System should play the video on the interface.
Alternative flows	
2a. System do not show any video file.	
4a. System do not play the video.	

3.1.6 Access Game Panel Use Case:

Figure 3.7 shows the scenario of allowing the user to join the multiplayer game running on the server device. System will show the client list of running game servers on the network and allow him to join the desired server to play multiplayer game with the other player.

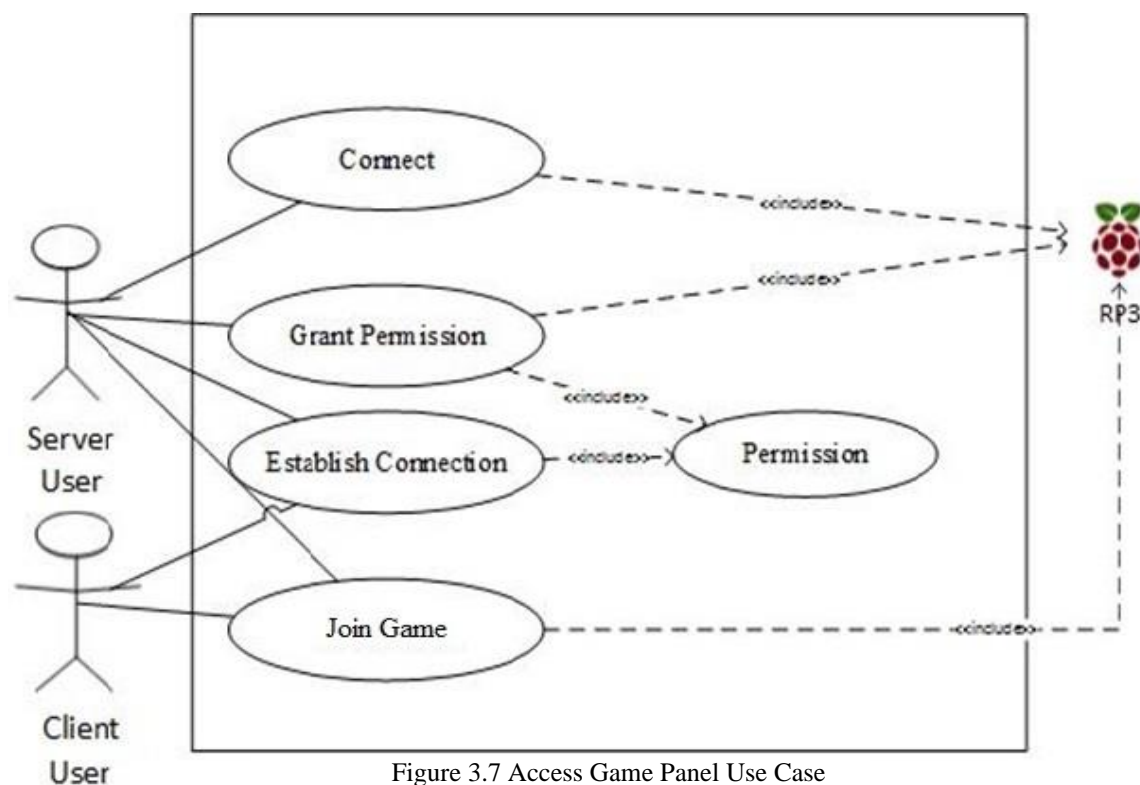


Figure 3.7 Access Game Panel Use Case

Table 3.6 shows the actors, purpose, overview, post and pre-conditions of the system. Pre-condition is that client should be able to the list of the servers. Post condition is that the user should be able to join the game.

Table 3.6 Access Game Panel Use Case

Use Case Id	UC-06
Use Case Name	Access Game Panel
Requirement Reference	Requirement No 6
Actors	Interface, server and client
Purpose	Allow the client users to join multiplayer games hosted on network
Overview	Client user can connect with the game running on the available network
Type	Essential
Pre-Condition	System should show the available game sever
Post-Condition	Client user should be able to join the game
Typical Course of event	
Actors Action	System Response
1. Client user selects the game access operation.	2. System will show hosted game servers on the network.
3. User selects sever to join the game.	4. System will establish connection and allow the user to join the game.
Alternative flows	
2a. System do not show any available server.	
4a. System don't establish the connection.	

3.1.7 Manage Cloud Use Case

Figure 3.8 shows the use case scenario of the cloud back-up feature of the system which allows the user to create a back-up of personal important data.

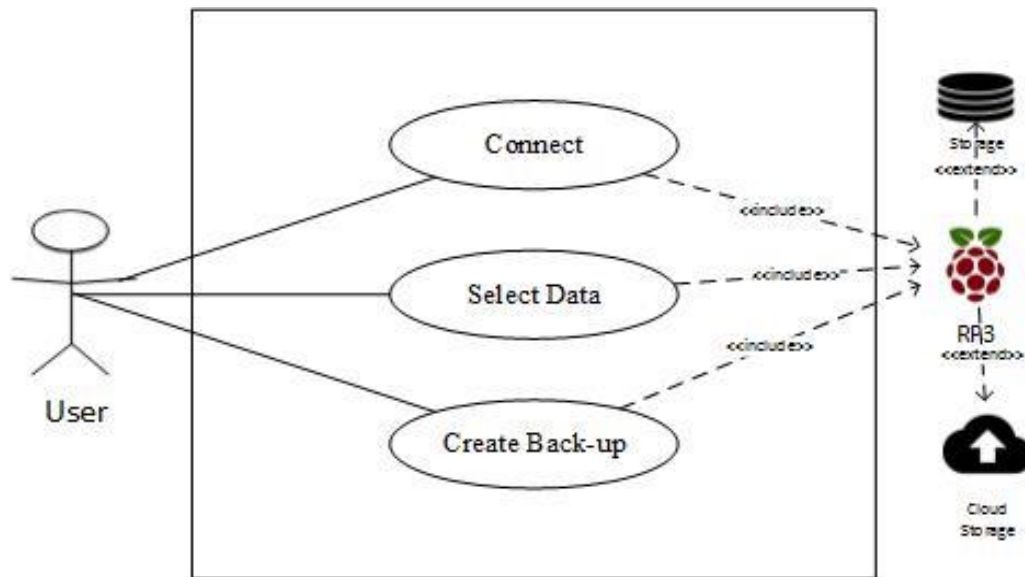


Figure 3.8 Manage Cloud Use Case

Table 3.7 describes the actor, purpose, overview, post and pre-conditions of the system. Pre- condition is that user should be able to select the data to create back up. Post condition is that system should make back-up of the data.

Table 3.7 Manage Cloud Use Case

Use Case Id	UC-WD-07
Use Case Name	Manage Cloud back-up
Requirement Reference	Requirement No 7
Actors	User and Interface
Purpose	Allow the user to back-up the data on cloud storage.
Overview	User can have the online access of the storage device for creating backup of data.
Type	Essential
Pre-Condition	User should be able to view the data over the internet.
Post-Condition	User should be able to store/extract data from the device.
Typical Course of event	

Actors Action	System Response
1. Client user selects the cloud back-up operation.	2. System should show the interface to upload data.
3. Client user selects data to create back-up,	4. System should upload the data in the storage device over the internet.
Alternative flows	
2a. System do not show the back-up options.	
4a. System do not upload files over the internet.	

3.2 Sequence Diagram

Sequence diagram is type of interaction diagram that describe interaction of processes with one and another and with user. Sequence diagram show object interaction arranged in time sequence. Sequence diagram is type of interaction diagram that describe interaction of processes with one and another and with user. Sequence diagram show object interaction arranged in time sequence.

3.2.1 Store Data Sequence Diagram

Figure 3.9 describes the flow of information between user, interface and centralized storage. In this sequence the user reaches the interface and connects to the raspberry pi. Then user access the storage media and store/extract the files.

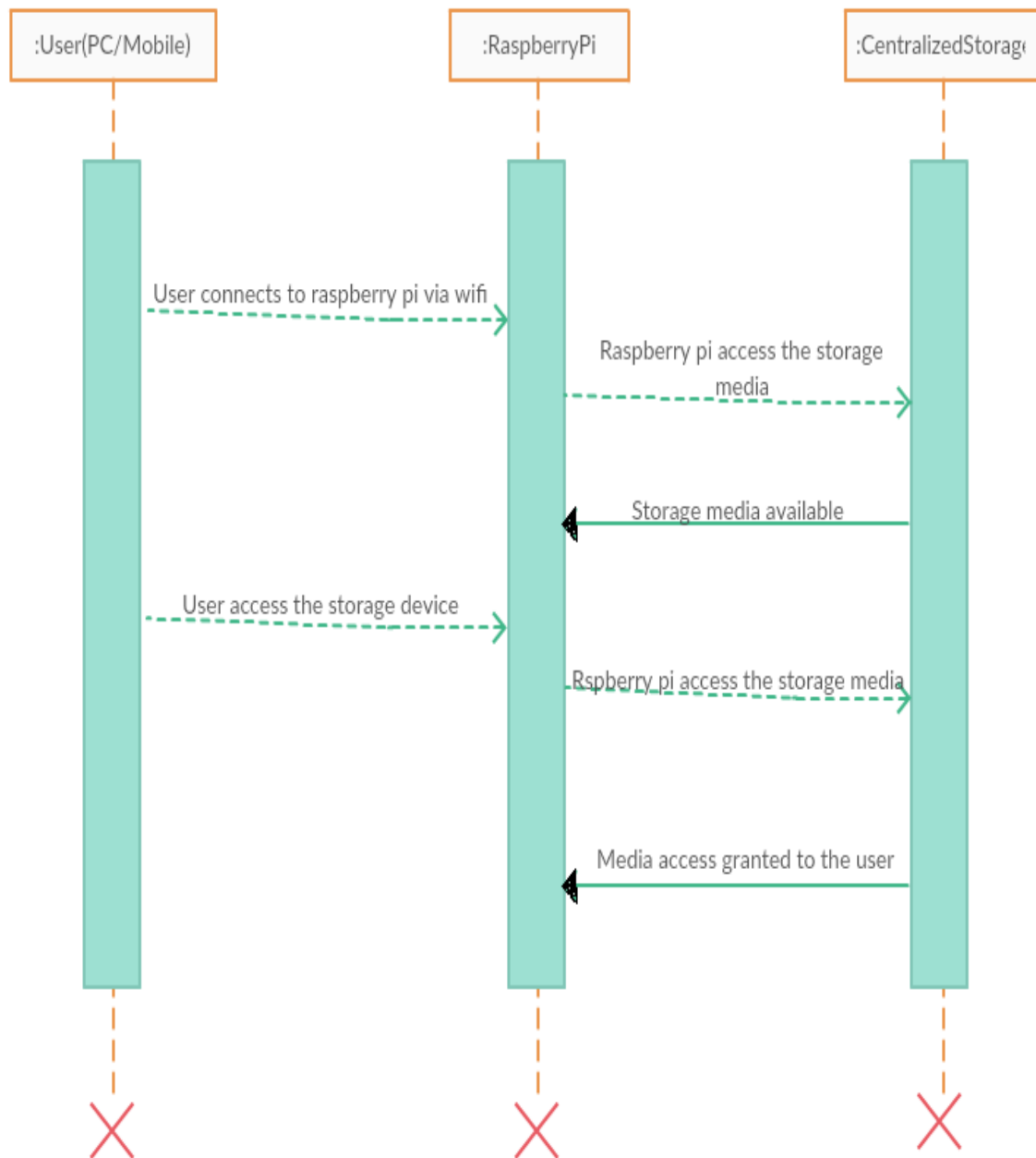


Figure 3.9 Store Data Sequence Diagram

3.2.2 Share Data Sequence Diagram

Figure 3.10 describes the flow of information from one user to other user passing through the server. In this sequence the user reaches the interface and establish the connection between the desire users. After that one user share the files with other user, then after file is received by the receiver, it notifies the senders that file is shared successfully.

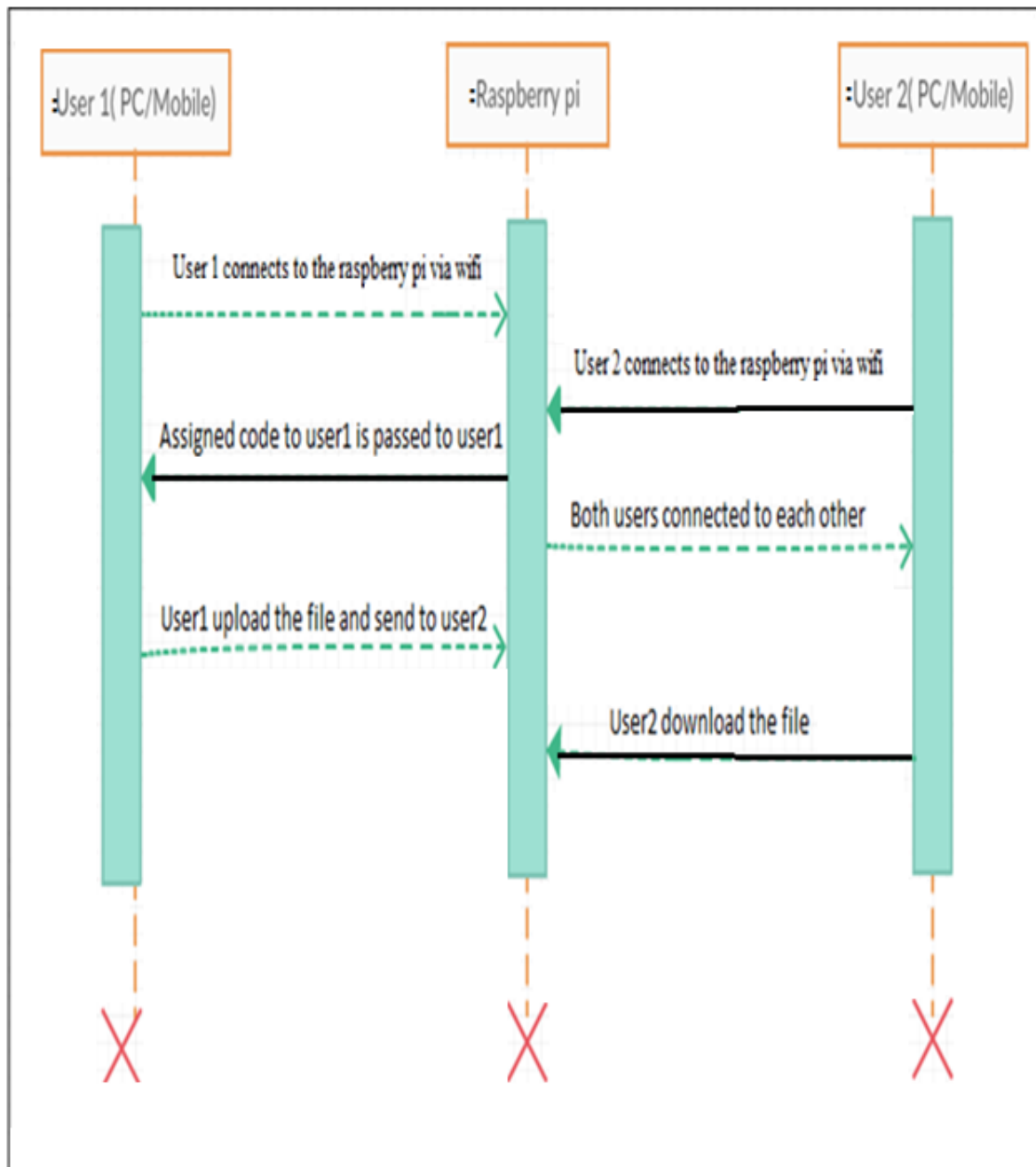


Figure 3.10 Share Data Sequence Diagram

3.2.3 Remote Desktop Sequence Diagram

Figure 3.11 describes the flow of information between two users on the WLAN using access point on raspberry pi. Server is hosted by one user and allow the other user to join using the password. Remote desktop connection established.

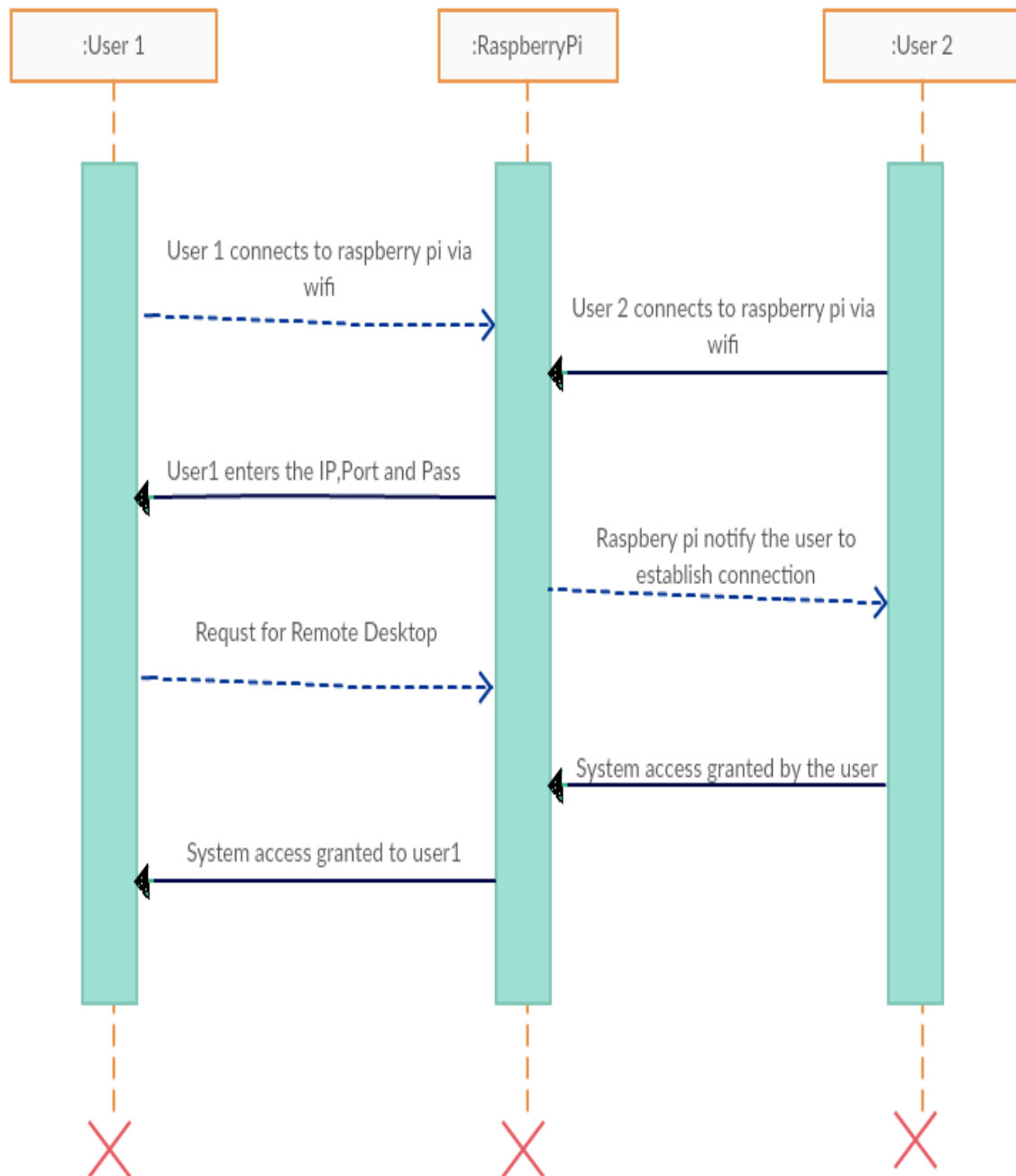


Figure 3.11 Remote desktop Sequence Diagram

3.2.4 Share Screen Sequence Diagram

Figure 3.12 describes the flow of information between Host user and the client user using the raspberry pi as an access point. In this sequence all users will connect to the raspberry pi via Wi-Fi and host user will start the screen sharing server with password protection. Other client users will access the screen using the password.

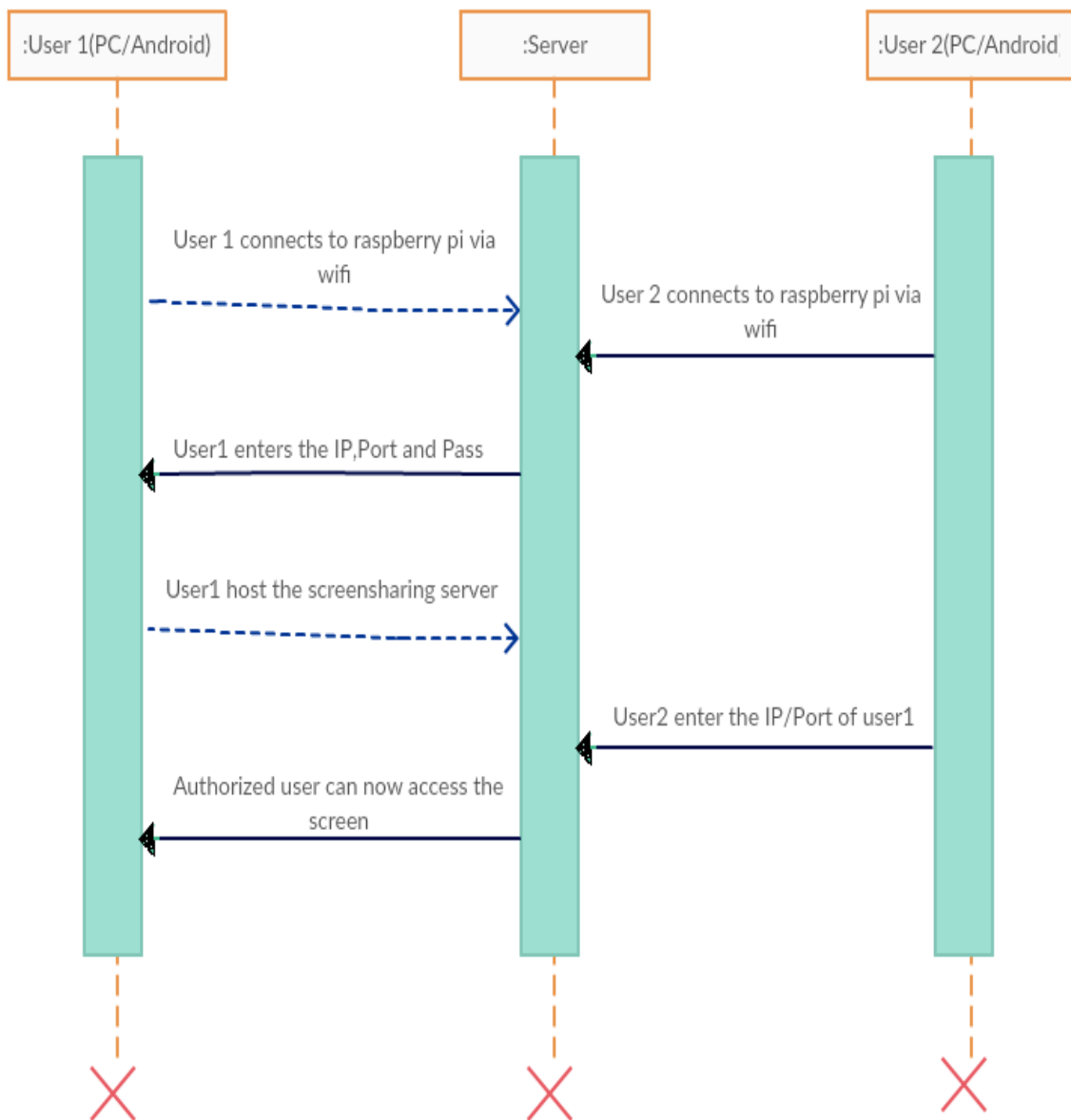


Figure 3.12 Share Screen Sequence Diagram

3.2.5 Stream Video Sequence Diagram

Figure 3.12 describes the flow of information between user and the storage available on the server. User establish the connection using the Wi-Fi and access the media storage, then user can stream the videos directly without copying it to personal storage device.

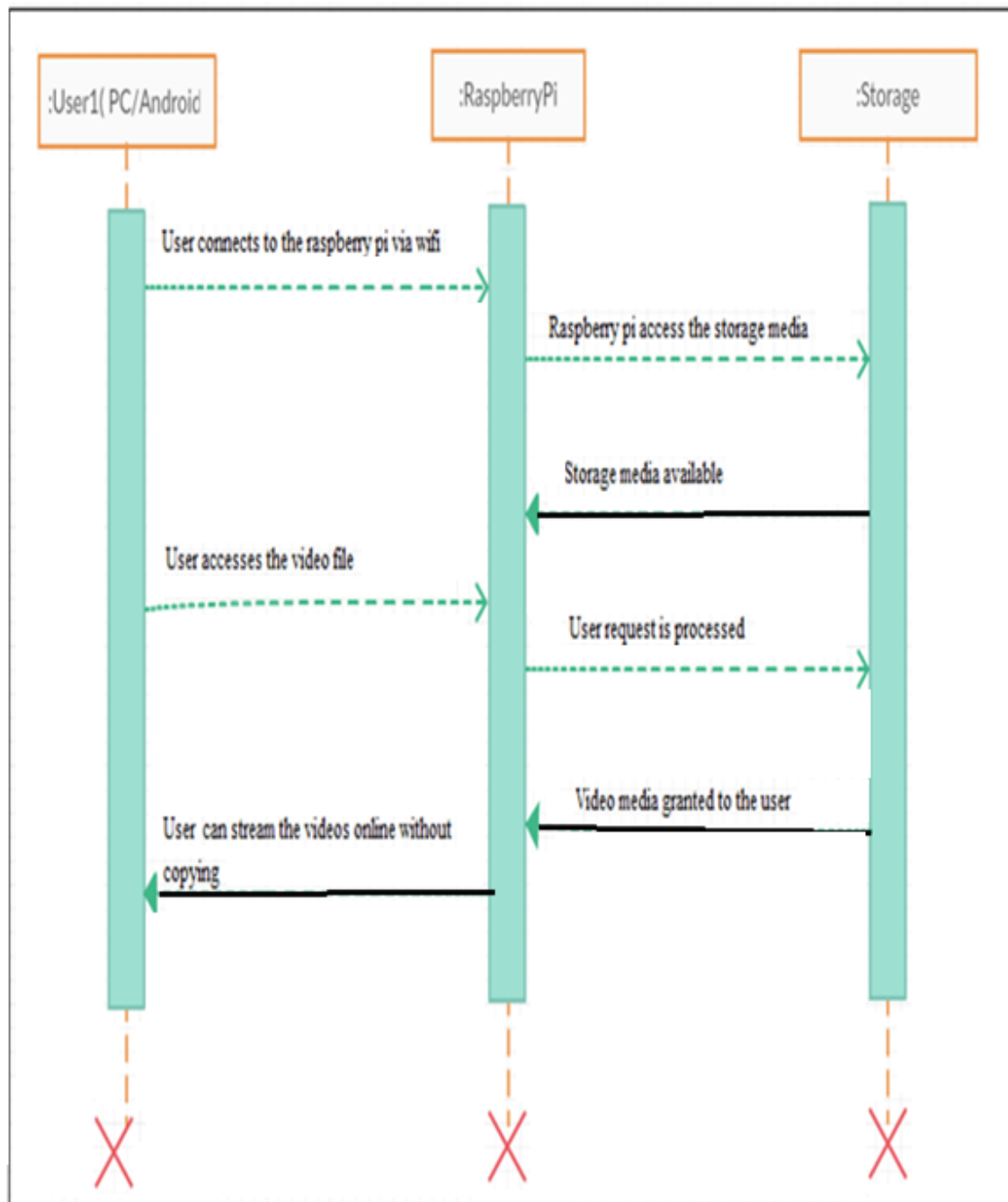


Figure 3.13 Stream Video Sequence Diagram

3.2.6 Game Point Sequence Diagram

Figure 3.14 describes the flow of information between users and the interface. In this sequence the system all the users will connect to the raspberry pi via Wi-Fi. Host user will start the server of the multiplayer game and client users will search for the hosted game server to join the game. Now player joins to the available server and can play with each other.

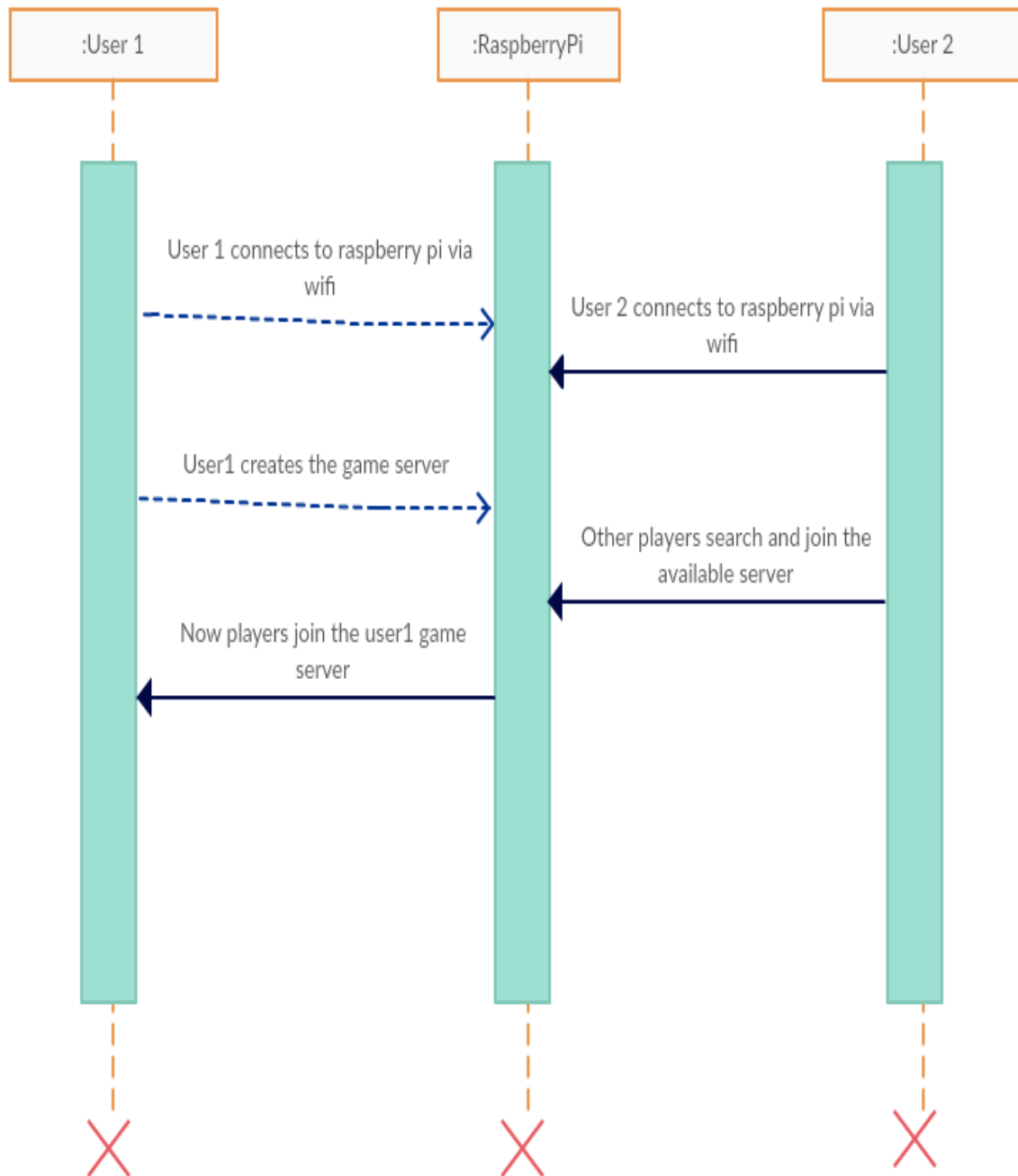


Figure 3.14 Game Point Sequence Diagram

3.2.7 Manage Cloud Sequence Diagram

Figure 3.15 describes the flow of information between the user, local storage and Cloud storage. User accesses the local storage and connects to the cloud backup to upload the desired important files.

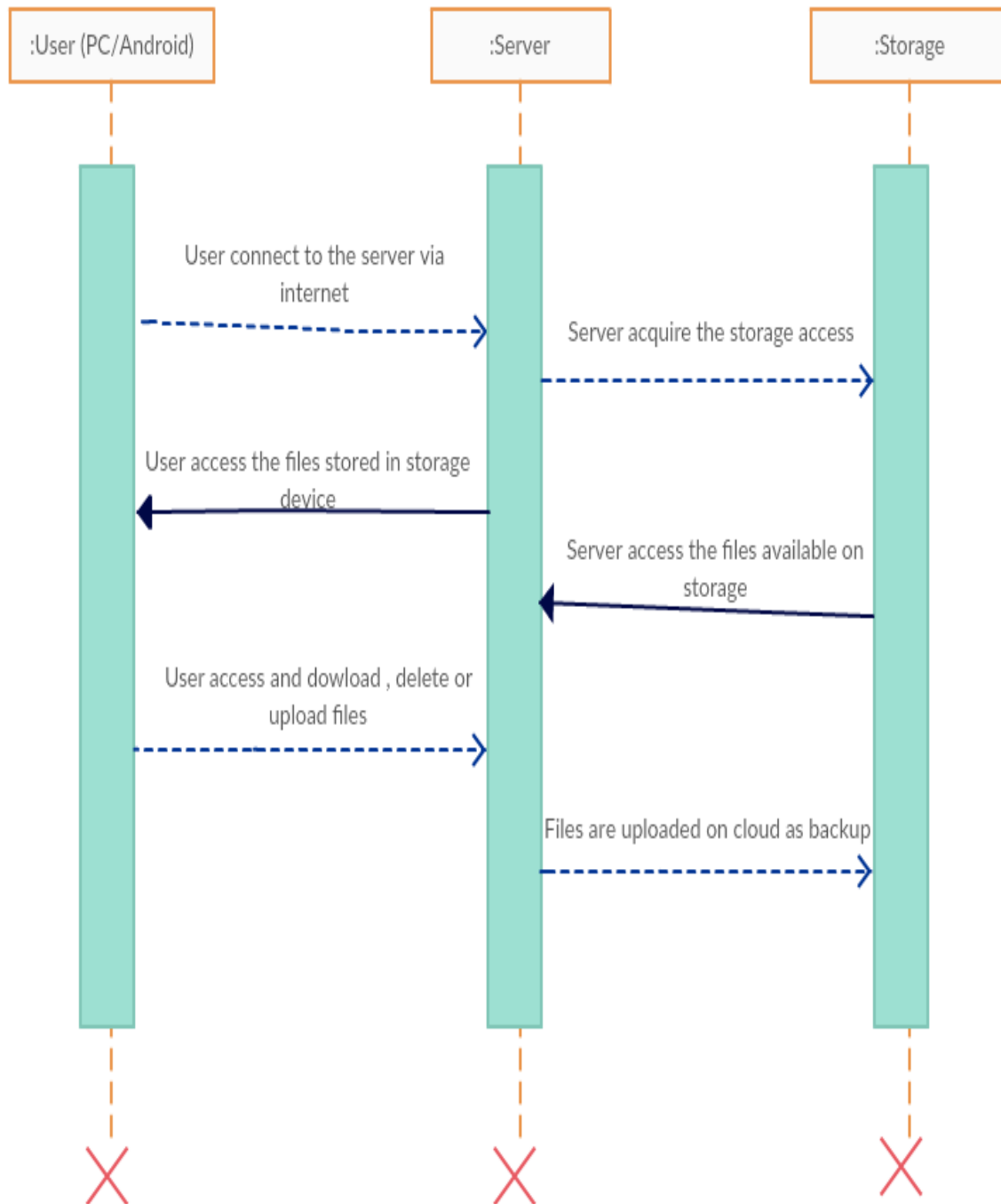


Figure 3.15 Manage Cloud Sequence Diagram

3.3 Compressed storage on centralized storage device over Wi-Fi Flow Diagram

In the beginning when application is started. It creates a network using and start searching for available devices. If any device is available system communicates with the device. In case if there is no device available, then the connection is closed. If the device is available for the connection the system will connect with it and then allow the user to select the operation.

System will perform the operation on the base of user's input. In this flow diagram user will have to select among four different options. If user wants to end the connection the system will terminate the connection.

If the selected option is data storing, then the device will create connection between the device and storage device which will allow the user to store or extract data. After performing the desired task the system will close the connection between the user and storage device.

If user selects to share data between two devices, then the system will establish a peer to peer connection between the selected deices and allow the user to transfer data between two devices, if system is unable to create connection with the other device it will automatically retry. On termination the system will close the sockets and end the connection between devices.

In case of remote desktop, the system will open a port in both devices through which it will establish a socket connection. If the connection is established the system will give the user remote access of the other device else it will reconnect. If the connection is closed the system will close the ports and end the connection.

If gaming access is selected the system will search for the online gaming device at the present network and allow the user to create connection between the host device and client device, which will enable them to join the game. If no gaming device is available the system will stop searching and end the connection. At the end after performing the tasks all the connections will be closed.

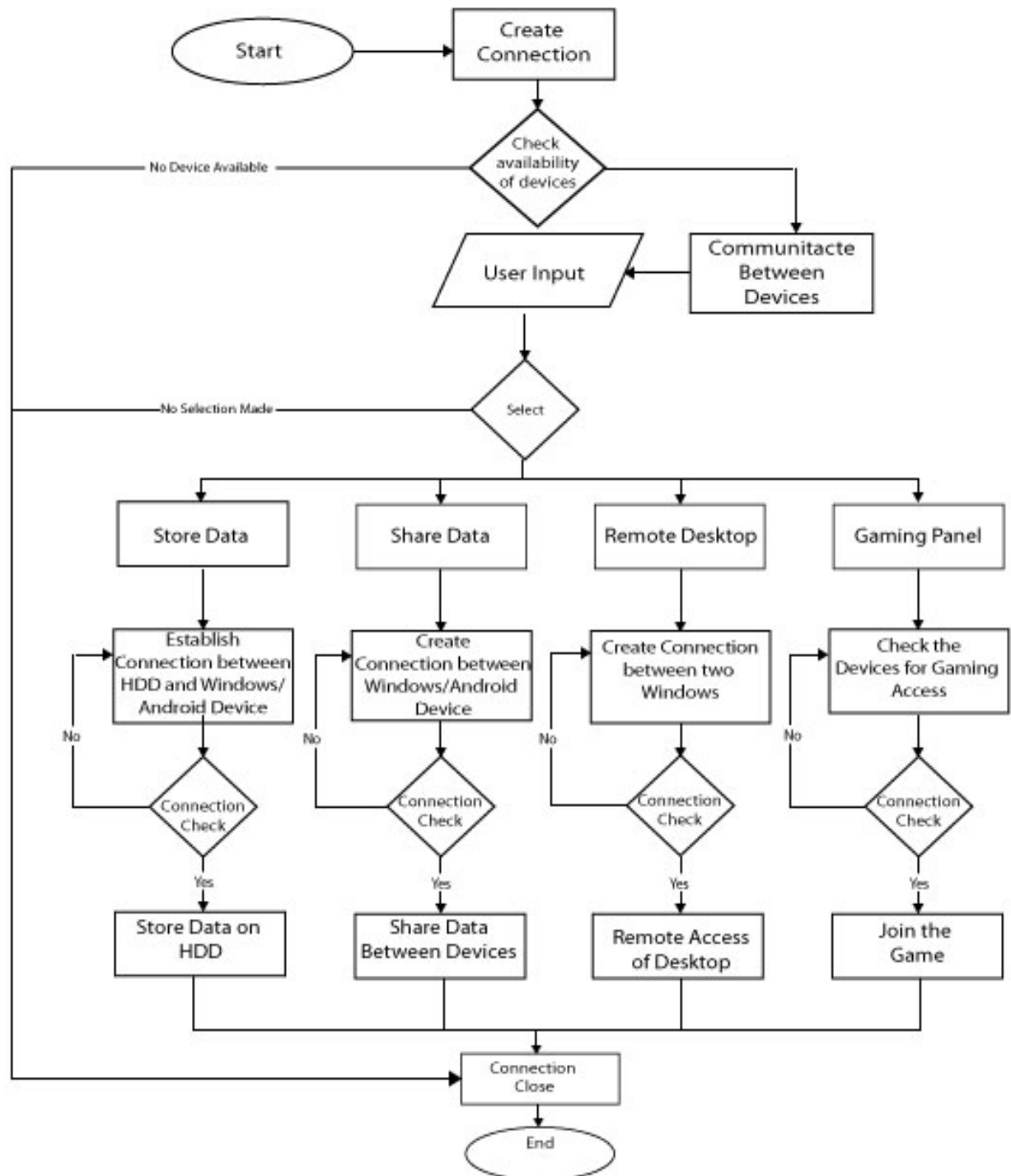


FIGURE 3.16 FLOW DIAGRAM FOR COMPRESSED DATA STORAGE ON CENTRALIZED STORAGE DEVICE OVER WI-FI

CHAPTER 4

DEVELOPMENT METHODOLOGY

4.0 Introduction

The development methodology and programming languages used for the development of the application are discussed in this chapter. It is all about development of Android & Desktop application. It's been defined thoroughly that how the applications interact with the person. Working of all functionality is also described in this chapter. This chapter includes all the information about the development and functionalities of the “Compressed Data Storage Centralized on Hard-Disk over Wi-Fi”. The entire working of the project is defined in this chapter. It also explains how the system achieves all the boundaries that have been in the previous device.

4.1 Application Development

In the first stage all requirements and features of the project were analyzed briefly. The according to these requirements modules are designed so that user can interact easily with the system. For construction of these modules and for development of the application, Visual Studio, Linux, Android Studio and Java Development Kit (JDK), Notepad++, Xamp server, Adobe after effects is used.

All the modules of application including home screen & all the functionalities for both android and desktop application. All interfaces of these modules are designed first and then these interfaces are selected which are more beneficial to interact for user. After selecting interfaces then we started implementation in the modules.

The application is developed in different languages. Java language is used for Android and backend code and instruction for the application. XML (Extensible Markup Language) is used for designing user interface which includes text input fields, buttons, keyboards, navigation tab which is used to access other features of the application and material design. HTML is used for creating the index page, we also use bootstrap for better options like dialogue boxes, Modals etc.

JavaScript & jQuery is use for functions like insert, update, delete, modify and animations. PHP is also used which is useful for server-side scripting and Apache Server is used. In Linux, Python is used to perform server-side operations. Development of android application is done in android studio IDE and for Desktop application we use Visual Studio and Notepad++. For server side, Python 3.6 is also used.

All the modules of the applications are constructed separately in Visual Studio. These modules are also tested separately. During testing of function Screen Share there was issue with port, it was not allowing to share screen due to firewall restrictions. When all modules are working properly then modules were integrated together at the end whole project is debugged and tested for final phase.

4.2 Application Design

The design of application is user-friendly so that user can interact easily with the application. Designing of the data storing and data sharing form is html bootstrap and CSS based. Client-side application designing is done by using Visual studio, Android studio, Adobe Photoshop, Adobe After effect. Use cases and sequence diagrams are designed in MS Visio diagrams are kept in mind while development of the application design.

4.3 Application Implementation

Implementation of the application is started just after the designing phase. Use cases helped to understand the application and saves a lot of time during the implementation of the application.

4.3.1 File Transfer

File transferring uses webRTC (web Real-Time Communications) peer to peer connection to transfer the data between the connected users. Basically, webRTC provides browsers and mobile applications with Real-Time Communications (RTC) capabilities using simple JavaScript. First step is to get the file from the user using JavaScript after getting file from the user, the file is converted into chunks to send it over the data channel. Maximum chunk size is 16KB. After making chunks or slices the sender is ready to connect with the other peer. The peer connection between the systems is made by using RTC Peer Connection. For this purpose, signaling process is done using simple node.js server and web socket. For the signaling process system is using a peer id which will allow the systems to connect with peer. After signaling, ICE (Interactive Connectivity Establishment) find ways for two systems to communicate as directly as possible, enables the system to communicate in various conditions like firewall. ICE is using STUN server which will allow the systems to communicate over the firewall. The data channel is created between two systems using RTC Data Channel. Using this data channel, the files are transferred from one device to the other. On the receiver's side the receiver receives

the chunks and reassemble them into files again. All these functionalities are implemented using notepad++, npm, chromium developer tools, visual studio windows command prompt and Linux bash shell. Notepad++ is used for the development of the html bootstrap form, Java scripting. Visual studio is used for making a client-side application which will show the web form created on notepad++.

4.3.2 Centralized Data Storage

Centralized data storage module allows the user to connect storage device with raspberry pi and access its data on mobile or desktop using Wi-Fi. Its implementation is done for multiple platforms i.e. windows, mobile. For this purpose, it is implemented using html bootstrap, JavaScript, jQuery, PHP which gives us a web form. This web form is hosted on the raspberry pi 3 using apache server and accessible for the users which are connected with the raspberry pi 3 through Wi-Fi. The form will display the data which is present in the storage device. The implementation on the raspberry pi 3 is done by using Python, which will detect the storage device connected to the raspberry pi 3 and mount the storage device in the /var/www/html/storage directory, so it can be accessible in apache server hosted web form. For the implementation purpose Python 3.6 compiler, visual studio, android studio, bash shell, notepad ++, and apache server is used. Visual studio is used for making the client-side desktop application which will show the web form created on notepad++. For the android client-side android studio is used for making android application which will show the web form. Python 3.6 compiler, bash shell and apache server are used on the server side which stores the web form.

4.3.3 Remote Desktop

The Remote Desktop application is commonly used feature. With Remote Desktop application, you can connect to a computer by running application from another computer running application without using internet connection. This functionality is done in Visual Studio and the C# language is used for implementation of the remote desktop.

4.3.3.1 Host Side Module

Remote desktop is a program that is used to control another window based computer that is running the remote desktop server using TCP (Default port 4000) and encrypted packets and this program allows users to control "Home Editions" of Windows desktop in just the same way as Microsoft's remote desktop without having to upgrade the license.

Simply enter the IP-Address and password of the computer to control and then press the "Connect" button or change the port number first if the default port of 4000 has been changed on the server. If Windows firewall is being used, then press the "Firewall" button to add a new firewall rule to allow this program on the client machine but please note that this option will only work if this program is running with administrator's privileges.

Pressing the little down button on the left of the blue title bar at the top of the screen once connected will display the settings menu where the size of the desktop can be tuned using the scroll-bars along with several performance options that trade screen quality and security for network bandwidth and include Compression/ Color (16/32bit)/Refresh Speed.

Windows Metro (Phone type interface) can be used by moving the mouse pointer just in from the top right corner which may also bring up the local window metro window or by pressing the "Metro" button on the dropdown remote desktop menu. Remote desktop works with the windows clip-board for cut and paste of text between computers but does not allow files to be transferred using cut and paste however this nice to have might be added at a later stage.

The remote server side of the program can service up to three connections at the same time with various configurations for each connection and can even be shut down by client connections, but you will need to tick the "Debug" box to see the service on the remote machine. For security reason the setting on the remote machine cannot be changed by a remote client running this program.

4.3.3.2 Client-Side Module

Remote desktop server is a service program much like windows remote desktop that allows client connections from other computers to control another computer and desktop as if the user was sitting in front of the computer but without having to upgrade to a more expensive version of Microsoft Windows.

Please change the password from "LetMeIn" to your own password or change the "Listen Port" number which you will need to share with anyone using remote desktop Remote desktop to connect to your computer and then press the "Save settings" button to save changes and to start the service. By default this program uses TCP port 4000 which will need to be allowed in the windows firewall if it is running on the local machine and this can be achieved by pressing the "Add Firewall Rule" button but please note that the

program will need to be ran with administrator for this to work and if the client computer connecting to the desktop server is using the internet then you may need to also add a NAT rule to your routers firewall to allow incoming TCP packets on port 4000 to be forwarded to the machine that is to be controlled.

The program can also be set to auto run as a user logs in to the computer by pressing the "Auto Start at logon" button which will create a scheduled task to start the service.

Each desktop server may host up to three remote desktop clients at any one time and can be hidden from view by pressing the "Hide" button, but the program will pop back up again if the client connection is set to "Debug" mode. Many windows system settings can only be controlled or accept mouse clicks from programs that have administrator privileges, so this program needs to be ran with administrator rights if the remote user is to be allowed to make changes to windows settings.

The program is a stand-alone ".EXE" program and can be uninstalled by simply deleting the file and removing the desktop shortcut icon.

4.3.4 Screen Sharing

Screen sharing involves sharing access to a given computer screen. Multiple clients can access host user screen. Its code implementation is done in visual studio and the language that we use is C#. Host side is implemented in C# while the client side is implemented in html and java script. Host user starts the server on desktop system and on the other hand client user writes the IP address/port in the application (Android and Desktop application). We have provided the host side application only for desktop devices while the client-side feature for both android and desktop application is provided. Clients are connected to the host side Server through same IP and host can share its screen to the client's PC or mobile.

4.3.5 Compression & De-Compression

Data compression and de-compression is done on the server side which allows the user to save space and computing power. Its implementation is done using JavaScript and PHP. JavaScript allow the user to select folder and PHP method converts the selected folder into .zip file. For de-compression the JavaScript will allow the user to select the file and PHP method decompresses the folder. For the implementation purpose of Compression & De-Compression notepad++ is used.

4.3.6 Video Streaming

The Video Streaming is used in application to stream video on different devices. Code implementation is done in visual studio, notepad++. For the implementation purpose html bootstrap, JavaScript, jQuery and PHP is used. Which allows the user to stream video from the applications on both desktop and android without copying it.

4.3.7 Gaming Point

Gaming point provide Access point for playing multi-player games. In this all the users connected to the Raspberry-pi for same Wi-Fi connection. The server connection is established from host side and all the users connected to the host Side Server and then they can play multi-player game.

4.3.8 Cloud-Backup

Cloud backup is a feature which allows the user to create a back-up of data over the internet. For the implementation purpose Visual studio and notepad++ is used. Notepad++ us used to create the web form which gives the user option to create backup and visual studio is used to create application which shows the web form on the client side.

CHAPTER 5

TESTING, ANALYSIS AND VALIDATION

5.0 Introduction

In this chapter functionalities of the framework are tested by using test cases which are according to the requisites of the user. These test cases are tested by giving inputs to each component of the system. These tests have been performed on Android emulator, smart phone with Android operating system and Visual Studio with windows operating system. Testing is important part to make sure functionalities of the system are working properly. Bugged and failure of the system are recognized and resolved.

5.1 Test Bed

Test Bed is a platform, or we can say it is method which is used to test a particular module. In test bed hardware and software involved for testing of the application. The test bed used for Android Application is smart phone running android operating system 3.0 and for Desktop Application we used Visual Studio. All functionalities of the application are tested by giving different inputs.

5.2 System Test Case

System tests are the third phase in the testing life cycle. The purpose of the system test is to ensure that the system is working fine and met the customer requirements. System tests are mostly "black box" tests since we are testing the application without seeing the source code. Testing is important because it ensure each functionality of the system is working properly. During testing many errors and bugs are arisen these errors and bugs are removed by making correction in the code.

5.3 Test Cases

The test case is a procedure through which a framework is checked that it is working properly or not. The method of developing test cases can also assist find problems inside the requirements or design of an application. After receiving results tester decided that a test case is passed or failed. Components of test cases are test case ID, Reference requirement, Version, Pre-condition, Test input, and Steps.

5.3.1 File Transfer

This test case is tested for uploading, downloading, deleting, compression-decompression and creating folder in the storage device. Tester tested the system by giving valid and invalid inputs to the system. Tester observed the system according to the input provided and how systems processes the input and gives output. Three tests have been performed

in table 5.1 Upload Test (uploading the data, checking the maximum upload limit, uploading speed). In table 5.2 File Update Test (deleting and renaming the files). And in table 5.3 Compression and Decompression is tested.

Table 5.1 Upload Test Case-I

Module Name: Upload Test		
ID	TC-1	
Version	1.0	
Pre-Condition	User must see the panel where user can upload files	
Item to test	Uploading file	
Test input	File to upload	
Steps	Click on the upload button Choose file to upload	
Expected Result	The Progress Bar will appear which shows the progress and after completion the message will appear “File Uploaded Successfully”.	
Actual Result	The Progress Bar appear which shows the progress and on completion it displayed the message “File Uploaded Successfully”.	
Conclusion	The test Case-I is successful. The exception handling code was handling the exception successfully.	
Status	Pass ✓	Fail Not Executed

Table 5.2 Panel Update Test Case-II

Module Name: Panel Update Test	
ID	TC-2
Version	1.0
Pre-Condition	User must see the panel where user can update files i.e. Delete, Rename
Item to test	File to Rename or Delete
Test input	File
Steps	<p>Select file to delete option to delete the file</p> <p>Select the rename button to rename the file</p>
Expected Result	System should ask for confirmation of deletion of the selected file, and then delete the file. Rename modal should be displayed, where user can enter the new name.
Actual Result	<p>System didn't prompt the user any confirmation dialogue.</p> <p>Rename modal was visible to the user.</p>
Conclusion	The test case-II is not successful. The system didn't show any confirmation dialog, but it deleted the file.
Status	<div> Pass Fail Not Executed </div> <div>✓</div>

Table 5.3 Compression & De-compression Test Case-III

Module Name: Compression & De-Compression	
ID	TC-3
Version	1.0
Pre-Condition	User must see the panel where user can Compress or De-compress the files.
Item to test	Folder to compress or De-compress
Test input	Folder
Steps	Select folder to compress Select folder to De-compress
Expected Result	The selected folder should show a dialog of compressing and compressed file should be visible in the list of files. The system should de-compress the selected file and show in the list of files.
Actual Result	The system displayed compressing dialog and added zip file in the list of files. System decompressed the zip file and added the decompressed file in list of files.
Conclusion	The test case-III was successful. Compression and decompression was done successfully with the notification dialog.
Status	<div> Pass Fail Not Executed </div> <div>✓</div>

Table 5.4 Data Transfer Test Case-IV

Module Name: Data Transfer	
ID	TC-4
Version	1.0
Pre-Condition	User must be connected with another device.
Item to test	File sending
Test input	File
Steps	Select any file from computer and send it.
Expected Result	When user selects the file and press send button the system should send start sending file with a progress bar, as indicator.
Actual Result	The system sent the file to the other connected device, with a progress bar.
Conclusion	The test Case-V is successful. The user was able to connect and transfer file from one device to the other
Status	<div> <div>Fail</div> <div>Pass</div> <div>Not Executed</div> </div> <div>✓</div>

Table 5.5 Data Transfer Test Case-IV Cont.

Module Name: Data Transfer	
ID	TC-5
Version	1.0
Pre-Condition	User must see the data transfer form and receive a unique key.
Item to test	Connection
Test input	Key
Steps	Enter the other device's key in the other device's code field.
Expected Result	When user enters the key of another device it should connect, and a file uploading box should be visible.
Actual Result	The connection established successfully and file choosing dialog was visible.
Conclusion	The test case-IV is successful. The exception handling code was

	handling the exception successfully.
Status	Fail Pass Not Executed ✓

Table 5.6 Remote Desktop Test Case-V

Module Name: Remote Desktop	
ID	TC-6
Version	1.0
Pre-Condition	User must be able to see the text field to enter the IP address of the other computer device.
Item to test	Connection
Test input	IP address
Steps	<ol style="list-style-type: none"> 1. Enter the IP address and port to connect 2. Press the connect button
Expected Result	When user enters the IP key of another device and press the connect button it should connect with the other computer device.
Actual Result	The connection wasn't established with error log of "System was unable to open the port".
Conclusion	The test case-VI was unsuccessful. The exception handling code was handling the exception successfully.
Status	Fail Pass Not Executed ✓

Table 5.7 Remote Desktop Test Case

Module Name: Remote Desktop	
ID	TC-7
Version	1.0
Pre-Condition	User's computer device should be connected with the other device and should be able to see the desktop screen of other system.
Item to test	Remote operations
Test input	Mouse events, Key strokes
Steps	<ol style="list-style-type: none"> 1. Perform mouse events (left click, right click) 2. Press the keyboard buttons

Expected Result	User should be able to perform mouse and keyboard event on the other computer remotely.
Actual Result	The user was able to perform functions on the other computer device, remotely.
Conclusion	The test case-VII was successful. The tester was able to perform operations remotely.
Status	<div> <div>Fail</div> <div>Pass</div> <div>Not Executed</div> </div> <div>✓</div>

Table 5.8 Screen Sharing Test Case-VI

Module Name: Screen Sharing	
ID	TC-8
Version	1.0
Pre-Condition	Host Side: User should be able to see the interface which allow to broadcast the computer's screen
Item to test	Broadcasting screen
Test input	Start button
Steps	Press the start server button.
Expected Result	After pressing the button user should be able to broadcast screen.
Actual Result	The user was able to broadcast screen without any error.
Conclusion	The test case-VIII was successful. User was able to broadcast screen over the WLAN.
Status	<div> <div>Fail</div> <div>Pass</div> <div>Not Executed</div> </div> <div>✓</div>

Table 5.9 Screen Sharing Test Case-VII

Module Name: Screen Sharing	
ID	TC-9

Version	1.0
Pre-Condition	Client Side: User must see the text field to enter the IP address of the broadcasted screen.
Item to test	Broadcasted screen
Test input	IP address
Steps	Enter the IP address and press go button.
Expected Result	After pressing the go button user should be able to see the broadcasted screen in applications' panel.
Actual Result	The user was able to see the broadcasted screen in the application.
Conclusion	The test case-IX was successful. User was able to see broadcasted screen over the WLAN.
Status	<div> Fail Pass Not Executed </div> <div> ✓ </div>

Table 5.10 Stream Video Test Case-IX

Module Name: Stream Video	
ID	TC-10
Version	1.0
Pre-Condition	User should be able to browse videos in the application interface.
Item to test	Video Streaming
Test input	Video
Steps	Press on the video
Expected Result	After pressing the video, it should start streaming.
Actual Result	The user was able stream the video without downloading it.
Conclusion	The test Case-X was successful. User was able to stream the video without downloading it.
Status	<div> Fail Pass Not Executed </div> <div> ✓ </div>

Table 5.11 Cloud backup Test Case-X

Module Name: Cloud backup	
ID	TC-11
Version	1.0
Pre-Condition	User should be able to see the interface where list of files is displayed.
Item to test	Accessing data
Test input	Data
Steps	Press on the backup
Expected Result	After pressing the backup button on the file, the system should show the file stored in the storage device.
Actual Result	The system was not unable to show the data present on storage device and gave the error “unable to access”.
Conclusion	The test case-XI was successful. System didn’t provide the cloud access due to VPN error.
Status	<div> <div>Fail</div> <div>Pass</div> <div>Not Executed</div> </div> <div>✓</div>

Table 5.12 Storage device Mount Test Case-XI

Module Name: Storage device Mount	
ID	TC-12
Version	1.0
Pre-Condition	The Linux terminal system should be up and running
Item to test	Storage Device mounting
Test input	Storage Device
Steps	Insert the storage device in the Raspberry Pi
Expected Result	After inserting the storage device, the data in the storage device should appear in the /var/www/html/storage directory.

Actual Result	The data from the storage device didn't appear in the /var/www/html/storage directory, because of the different file system of the storage device.
Conclusion	The test case-XII was unsuccessful due to the difference of file system of linux and storage device.
Status	<div> <div>Fail</div> <div>Pass</div> <div>Not Executed</div> </div> <div>✓</div>

5.4 Testing Results

The testing results of Desktop and Android application “WIDA” are given in table 5.13. Total twelve test cases were executed out of which nine were giving the positive result. Three test cases did not provide the desired output. Overall 80% test cases were successfully executed and 20% were not.

Table 5.5 Testing Results

Serial	Test Case ID	Test Case Description	Actual Test Result	Status
1	TC-01	Upload Test case	Displayed the message “File Uploaded	Pass
2	TC-02	Update Test case	Didn't prompt the user any confirmation dialogue	Fail
3	TC-03	Compression & De-Compression test case	Compressing and decompressing dialog appeared	Pass
4	TC-04	Data Transfer Test Case-I	File choosing dialog was visible	Pass
5	TC-05	Data Transfer Test Case-II	The system sent the file to the other connected device, with a progress bar.	Pass

6	TC-06	Remote Desktop Test Case-I	The connection wasn't established with error log of "System was unable to open the port".	Fail
7	TC-07	Remote Desktop Test Case-II	The user was able to perform functions	Pass
8	TC-08	Screen Sharing Test Case-I	User was able to broadcast screen without any error.	Pass
9	TC-09	Screen Sharing Test Case-II	The user was able to see the broadcasted screen	Pass
10	TC-10	Stream Video Test Case-I	The user was able stream the video	Pass
11	TC-11	Cloud backup Test Case-I	System gave the error "unable to access".	Fail
12	TC-12	Storage device Mount Test Case-I	System did not mount the storage device	Fail

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.0 Overview

In the development of Desktop & Android application, fundamental programming, Photoshop for designing models, ideas, systems, and improvement procedures were utilized.

This project has been done with most of its modules and targets. Each one of the portions and focuses of the undertaking are done and working fine. Various test cases have been performed on each of the module of the venture and are passed effectively.

This project is both a Desktop Application and android application for the users of android smartphones, tablets, PC's and laptops. With help of this application users can Access the data from centralized storage, exchange data, playing games, watching videos, remotely access partner's PC's and access the data through Cloud-Backup. User just have to select option and provide some inputs for that appropriate function to avail desired function. User can also take advantage from Remote Desktop feature of the application. User simply have to type IP to the partner's PC and then user can get access to the partner's PC. In addition, administrators can easily remotely access headless computers (without monitor, keyboard or mouse).

6.1 Achievements and Objective

The target that has been completed and project is successfully developed. Knowledge of building up the Desktop and Android based application is achieved, integration of different functionalities of the system is also performed.

Using the desktop or android open source APIs for android and desktop development the application features are enhanced, and the performance of application is increased by working on new features and integrating with legacy ones. Visual Studio is used to test the developed desktop application and for android. Final phase testing of the system was performed in smart phone with Android operating system.

The entire project was built using windows operating systems, raspberry Pi, Java, C#, Php, JavaScript & jQuery as a programming language, XML & Photoshop is used for light weight layout of smartphone applications and designing. Application is developed to install and run for Android 5.0 and upper version of Android operating systems and on all windows.

6.2 LIMITATIONS

Limitations in every project of developing systems are important because these limitations are the basis of starting new rules, improvements in features and development of functionalities. While developing the Compressed Data Storage on Centralized Hard-Disk over Wi-Fi, the system faces some limitations which are as follows:

Our system does not work over the internet. Size of Data vary from file to file, everyone is in hurry now a day but sending large files through this system is time consuming it takes time. To communicate with different users, users have to type the IP address manually.

6.3 Future Recommendations

This system has been designed in keeping mind each aspect of user friendly interface and more work can be done easily to adjust the system according to future needs of users or clients. Facilitate work may center to enhance the elements, innovation and the UI. In future developers may introduce more APIs of Desktop or Android application that enhances the technology and development. Raspberry pi and Android gives an open source improvement to design and develop application with passage of time.

6.4 Possible Improvements

As development in Desktop and Android field is increasing day by day developers keep in mind that their applications can be extended with passage of time. Following are the possible upgrades of this application which can be executed to enhance and improve the existing framework later on.

6.4.1 Compatible With Different Platforms

The application developed is for Desktop or smart phones running Android operating system and windows, other devices having operating systems such as Windows on web, IOS, Blackberry and Symbian can't use this system.

6.4.2 File Transfer for Web

We can transfer the file only in Desktop or android application. This functionality can also have performed by using internet. By using web RTC via peer-to-peer browser connection.

APPENDICES

Appendix I – Definition of Terms

API

It is stands for "Application Program Interface.

SDK

It is stands for Software Development Kit.

OS

OS stands for Windows Operating System.

XML

XML stands for Extensible Markup Language.

APK

APK stands for Android Package Kit.

Appendix II – Installation Guide

Application is installed on Laptops, smartphones, tablets and any device on Android or windows operating system in it. Through Android Studio an “APK” file is generated which is copied on Android device and through installation WIDA application is automatically installed on smartphone. Android Studio version 3.0 is used for application development.

For Desktop application it comes up with .exe file. We have to install this .exe file. When you run that .exe file the system will start installation. After installation of application you can open the application and use it.

Appendix III– User Manual

Main Interface

When the user opens the application, main interface appears which is the landing page. Which allows users to perform or select the different tasks such as Store data, Share data, Access screen, Cloud backup.

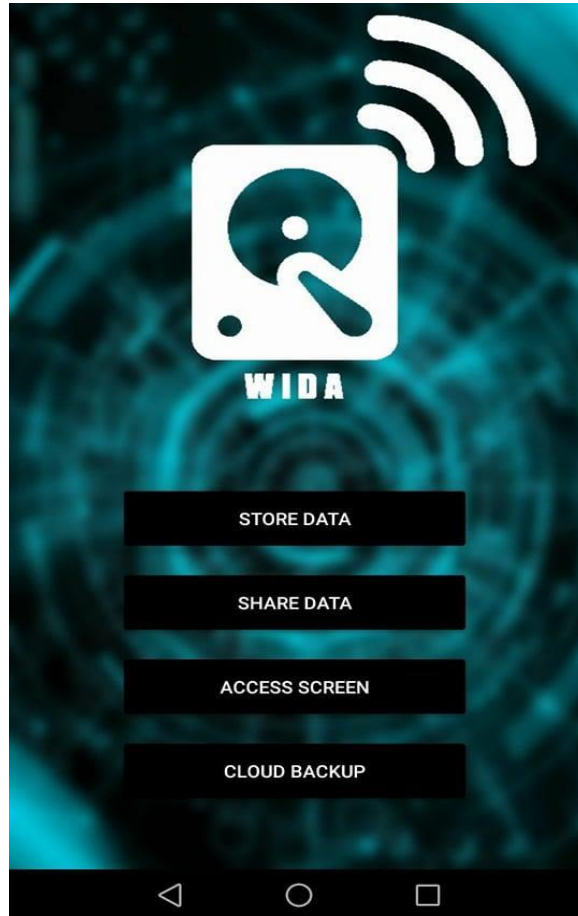


Figure AIII.1 Main Interface (Android)

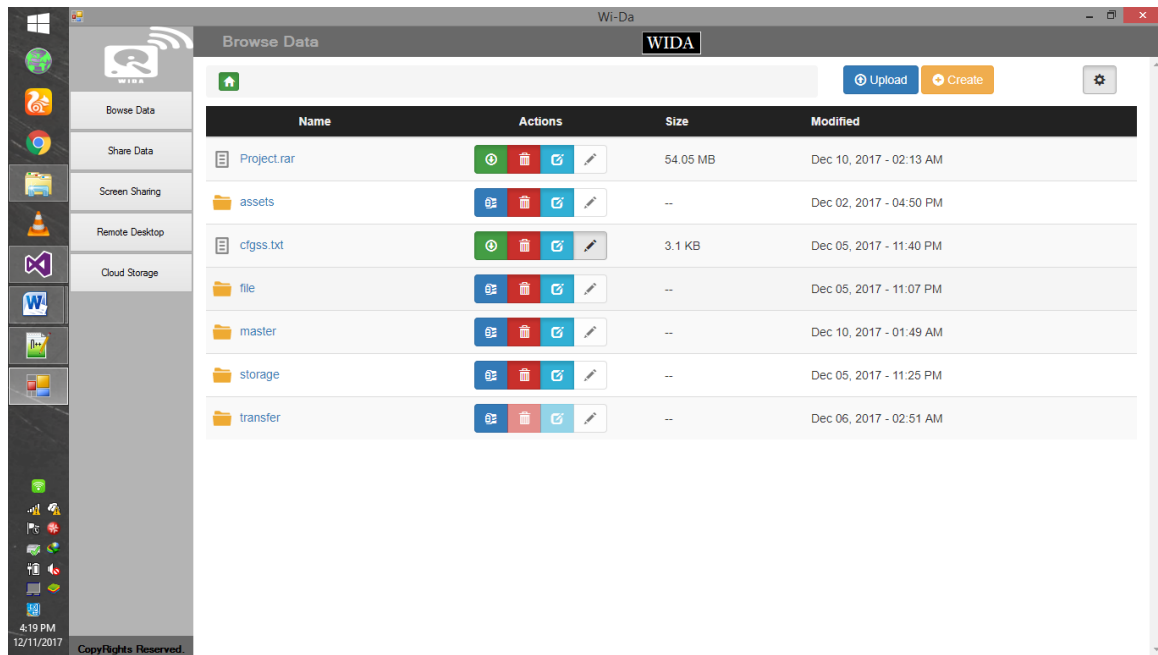


Figure AIII.1.1 Main Interface (Desktop)

Store Data Interface

When user click on button store data then application performs the function to store the data. One form will appear as you can see in figure AIII.1 or AIII.2. The form will display the data which is present in the storage device and user can store, download and delete the data.

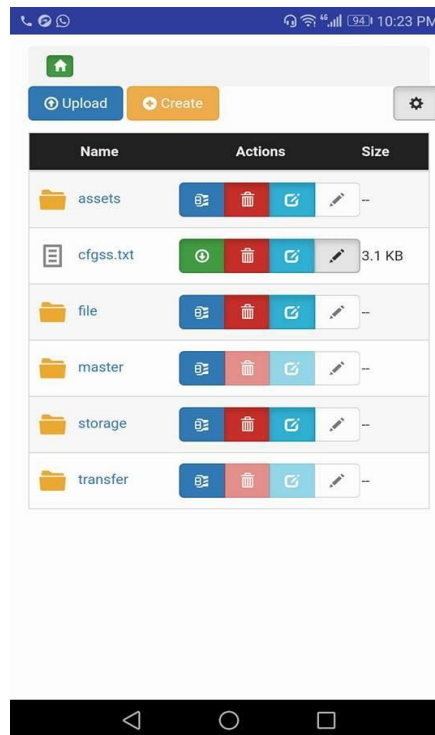


Figure AIII.2 Store Data Interface (Android)

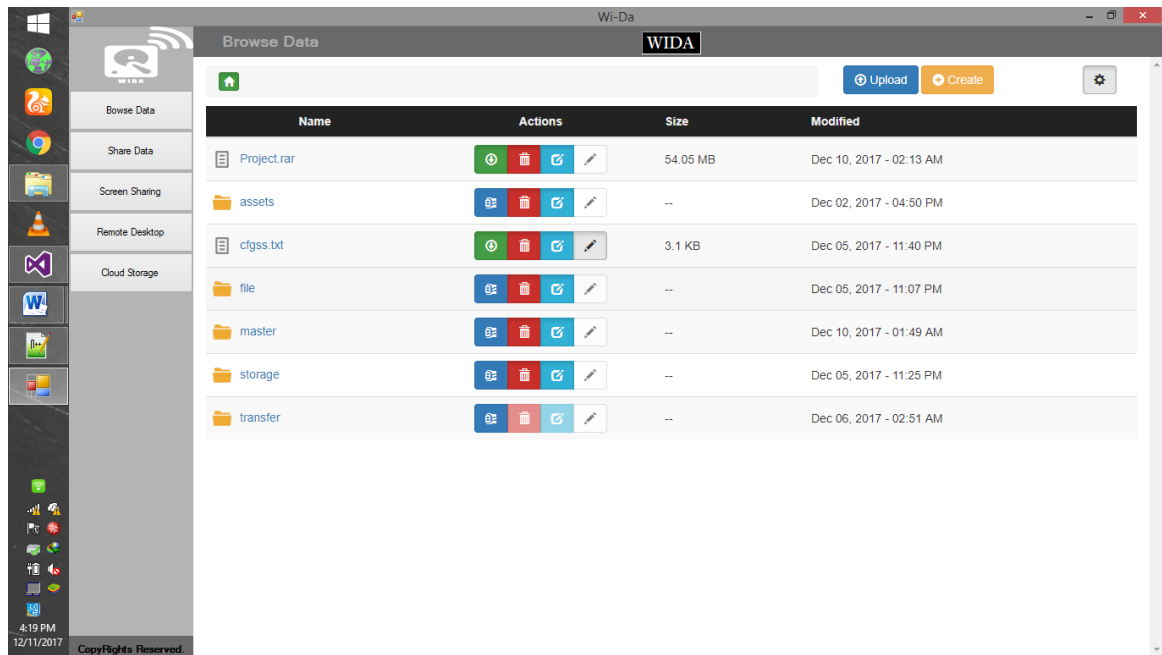


Figure AIII.2.1 Store Data Interface (Desktop)

Share Data Interface

When user selects Share Data from given menu down from store data side new interface appears where the File Share Panel is opened.

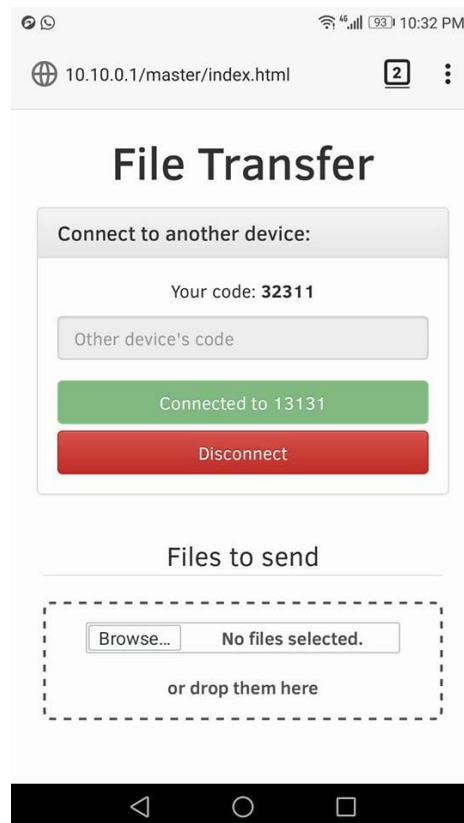


Figure AIII.3 Share Data Interface (Android)

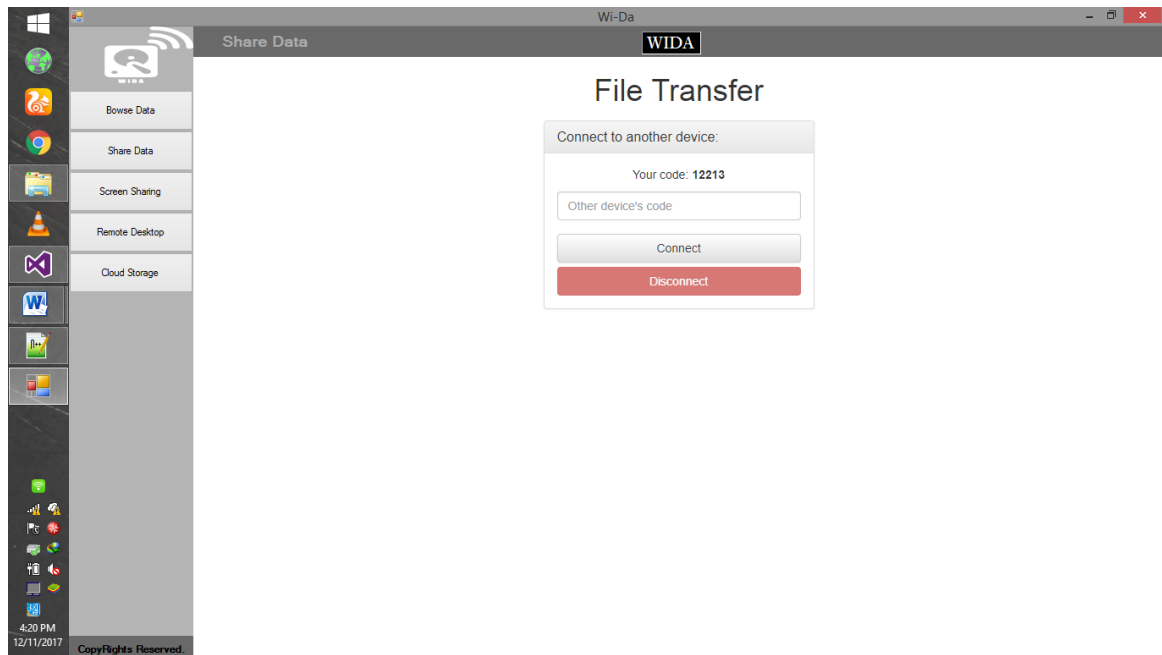


Figure AIII.3.1 Share Data Interface (Desktop)

Access Screen

When user select option Access Screen, user can watch the screen of client PC's and at the same time multi users can watch that same movie on their personal laptops or mobiles.

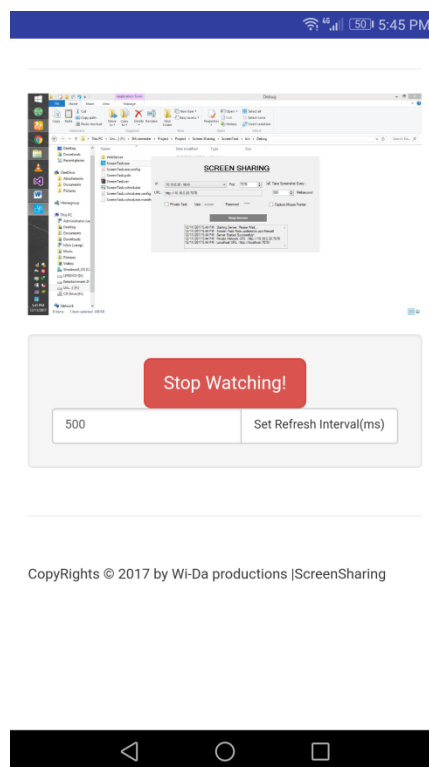


Figure AIII.4 Access Screen Interface (Android)

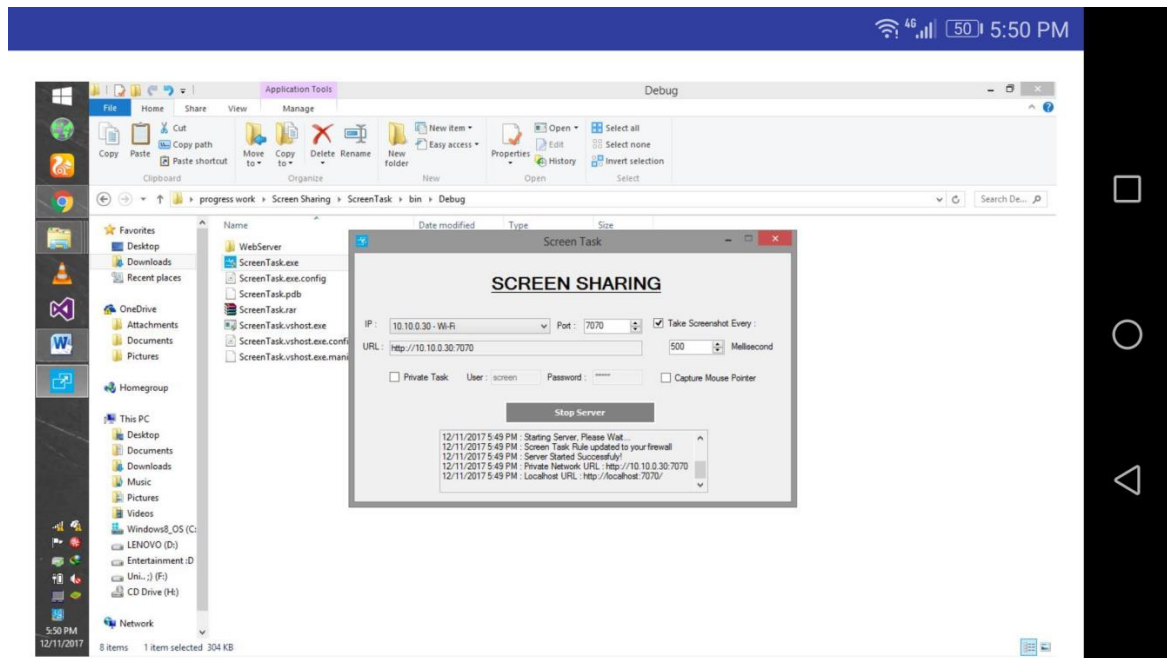


Figure AIII.4.1 Access Screen Interface (Android)

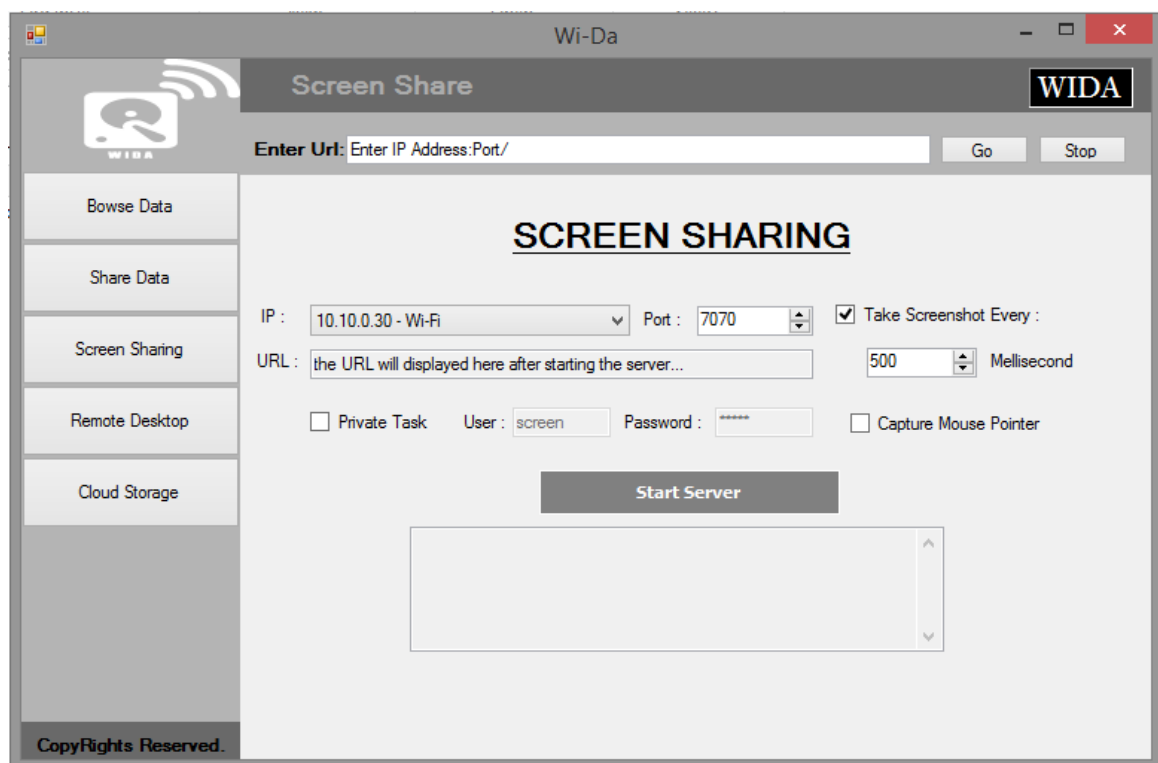


Figure AIII.4.2 Screen Sharing Interface (Desktop)

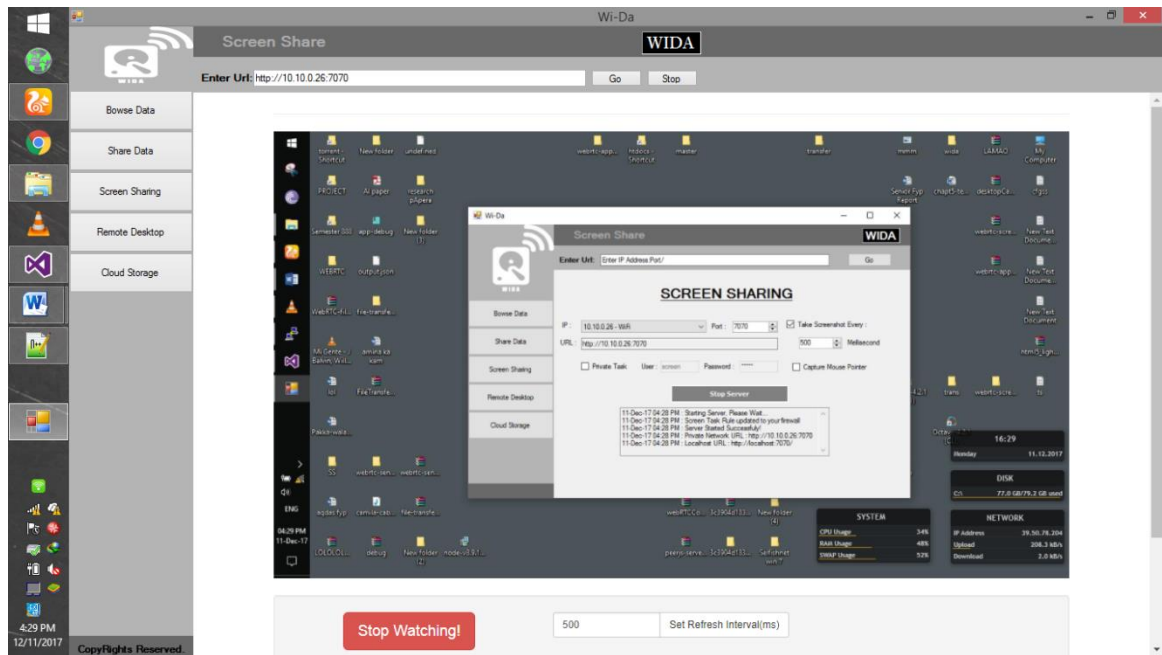


Figure AIII.4.3 Screen Sharing Interface (Desktop)

Remote Desktop

Host Side

When user click on button Remote Desktop Host another window will open where you can set the port and password though which client will connect and access your PC. The interface that appears after clicking the RD host button is:

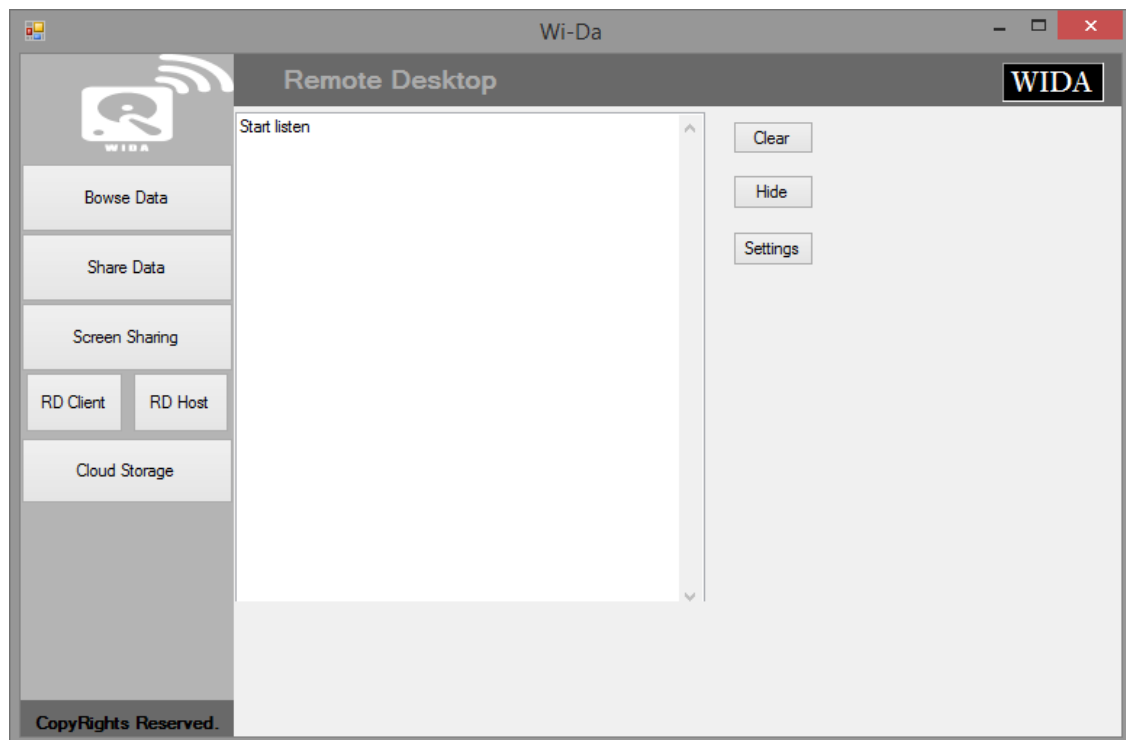


Figure AIII.5 Remote Desktop-Host Interface

Client Side

When user click on button Remote Desktop Clint another window will appear where you can set the IP, port and password to connect and access host PC. User can also add rules in the firewall. The interface that appears after clicking the RD client button is:

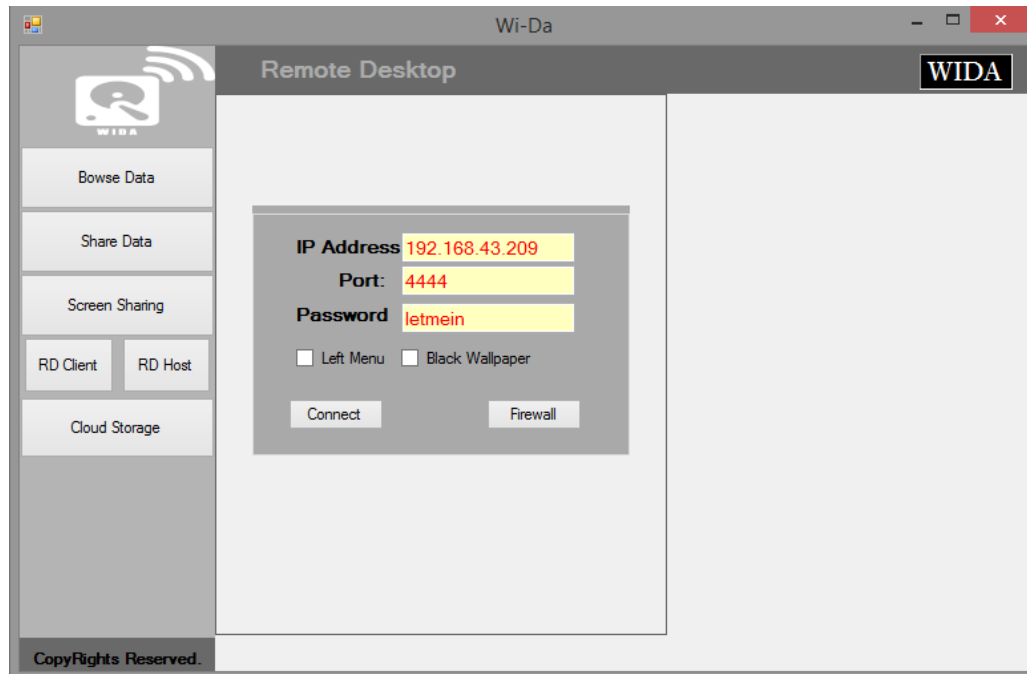


Figure AIII.5.1 Remote Desktop-Host Interface

Here is the Screenshot when the Client user is connected to host user and controlling the host pc.

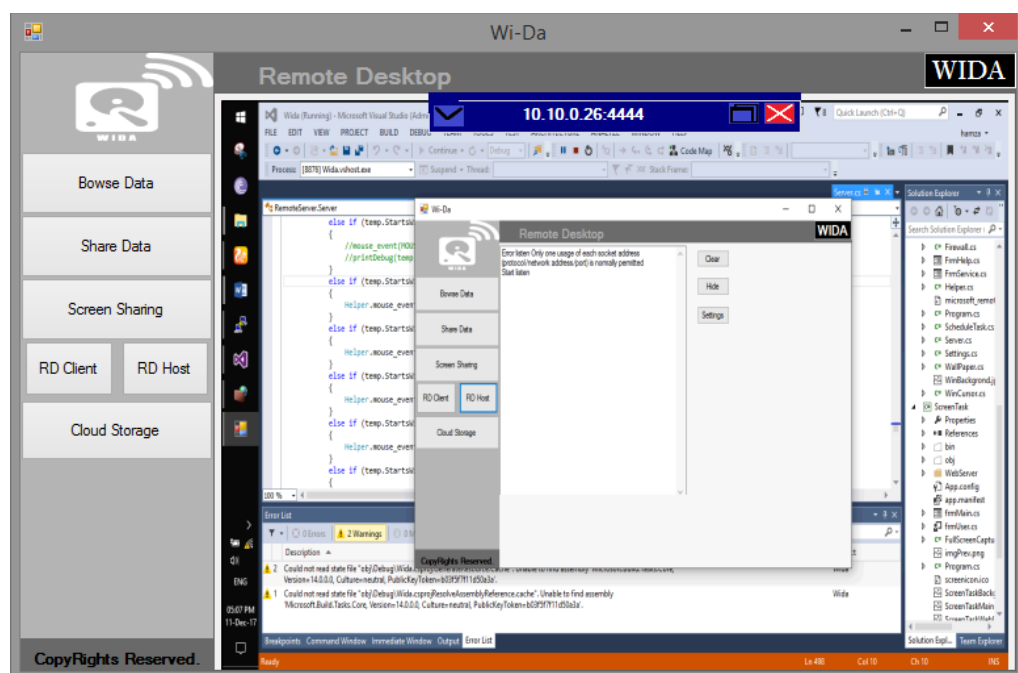


Figure AIII.5.2 Remote Desktop-Host Interface

REFERENCES

- [1] Seagate, “Seagate wireless Device,” [Online]. Available: <http://www.seagate.com/in/en/about-seagate/>.
- [2] unknow, “Teamviewer,” [Online]. Available: www.teamviewer.int.com.
- [3] Sanddisk, “Sanddisk USB stick,” [Online]. Available: www.sanddisk/usbstick.uk.
- [4] A. S. O. I. f. Android, “Android Developers,” 12 JAN 2018. [Online]. Available: <https://developer.android.com/studio/index.html>. [Accessed 15 FEB 2018].
- [5] W. i. A. S. -. D. f. Techopedia, “Techopedia.com,” [Online]. Available: <https://www.techopedia.com/definition/4220/android-sdk>. [Accessed 12 SEP 2017].
- [6] W. i. J. D. K. (. -. D. f. Techopedia, “Techopedia.com,” [Online]. Available: <https://www.techopedia.com/definition/5594/java-development-kit-jdk>. [Accessed 16 SEP 2017].