ALL IN ONE (AIO) DESKTOP REMOTE APPLICATION

By

Capt Muhammad Ammar Basra

Capt Hafiz Muhammad Umair

Capt Asfandyar Ali Khan

Submitted to Faculty of Department of Computer Software Engineering National University of Sciences and Technology, Islamabad in partial fulfillment for the requirements of a B.E Degree in Computer Software Engineering, June 2019

In the name of Allah, the All-Sacred, the Ever Tranquil

ABSTRACT

ALL IN ONE (AIO) DESKTOP REMOTE APPLICATION

Devices are the essence of the digital age, with every individual in any organization, either it's a corporation or a university, continuously strive to take the best out a plethora of devices, thereby, wants to access and handle these devices and activities on their computers. For this purpose, Remote Desktop Access and its remote control plays an important role. It allows a user to constantly access, monitor and control major aspects of data and activities on a particular desktop. Through this, an administrator can access and see concerned features of target desktop on android device. With monitoring, a person can even completely control their own PC or another's to help in resolving an issue by having access to their desktop and then controlling their Keyboard and Mouse.

The purpose of this project is to control the PC by accessing the desktop remotely we can work on a normal desktop by accessing it on Android device. The end user can exercise several useful interactions with desktop. Users can share their resources without a heavy capital expenditure on hardware and software resources. All functional logic operations are executed on remote computer and the user interface functionalities belongs to the mobile device. The mobile device controlling the computer acts as a remote control (client), which capture the user's input and depicting to show resulting updates they entail on the target desktop. The normal pc desktop can be accessed over Wi Fi remotely via an android platform and gives the user access to not only data on the desktop, but also to manifold features. The operations such as accessing system controls like shutdown, log off, restart through the mobile phone and the remote computer's desktop can be accessed from the mobile platform. Control of slide shows makes forgetting to charge or bring the ticker to your presentations a minor inconvenience at best.

CERTIFICATE FOR CORRECTNESS AND APPROVAL

It is certified that work confined in this thesis – All in One (AIO) Desktop Remote Application, carried out by Captain Muhammad Ammar Basra, Captain Hafiz Muhammad Umair and Captain Asfandyar Ali Khan under supervision of Lecturer Ayesha Naseer for completion of the Degree of Software Engineering is correct and approved.

Approved By

Lecturer Ayesha Naseer

Department of CSE,

Military College of Signals.

Dated: _____

DECLARATION

We hereby declare that this application has not been copied out from any source. We further declare that we have developed this application and accompanied report entirely based on our personal efforts, under the sincere guidance of our respected supervisor and teachers. If any part of this system is proved to be copied out from any source or found to be reproduction of someone else, we shall stand by the consequences.

DEDICATION

This thesis is dedicated to our Parents, we could never have accomplished it without their faith, support and persistent encouragement. To our respected supervisor, Lecturer Ayesha Naseer who has given us great support and valuable suggestions throughout the implementation process and lastly to other faculty members who supported us all the way.

ACKNOWLEDGMENT

Primarily, our team is grateful to Allah Almighty Who have given us strength to understand, learn and develop this project. It only became possible due to His kindness and blessings. We also admire the help and guidance of Lecturer Ayesha Naseer who motivated us and guided to work on this venture. Her continuous support in collecting information about project work, her patience, motivation, enthusiasm and immense knowledge is commendable. Deprived of her we could not imagine accomplishing the project In time. Throughout the course of effort, she never declined to help us.

Lastly, we like to acknowledge with gratitude, the support, encouragement and love of our family. It is their warmth and continuous backing with prayers vested us to attain the goal.

PREFACE

This thesis will comprise of the detailed study, designing, implementation and testing of our project “All in One (AIO) Desktop Remote Application”. To easily comprehend, the thesis is into six chapters.

Chapter One: This chapter is comprised of the introduction, objectives and scope of the system.

Chapter Two: This chapter contains requirements analysis of the Desktop Remote Application to be developed.

Chapter Three: It describes the architecture and design of the proposed “AIO Desktop Remote Android Application”. It gives the overview of the system design, description and the relationship between different modules and their interactions.

Chapter Four: It contains the details regarding graphical user interface of the “AIO Desktop Remote Android Application” to be developed.

Chapter Five: This section discusses about the quality assurance and analyzing quality testing. Test cases of each module is discussed in detail, explaining which components of the software to be tested and not to be tested and the testing mechanisms used to achieve the overall testing of the system.

Chapter Six: This user manual guide will provide required description for all the platforms and tools which are used in the development of this project and the procedure to install them. This manual will summarize the way to install All in One (AIO) Desktop Remote Application on android device and required IDEs on the computer system.

TABLE OF CONTENTS

CHAPTER # 1

INTRODUCTION

1.1 Introduction

All in One (AIO) Desktop Remote Application is to provide user friendly platform to access some basic features of their personal computer and create a signal sharing medium between mobile and computer for different functionalities.

Live Screening, Touchpad Control, File transfer, file downloading, Image viewer, Media Player, Presentation and Power Off are the main parameters which are added to the scope of this project, measure across the maximum achievable characteristics using any android portable device. All-In-One (AIO) Desktop Remote enables users to not only control their power point slides and media player but the whole computer via an Android device. Unlike many other wireless solutions or remote managers for Android, Desktop remote app need you to install additional companion software on your computer so that you can control your system through android device. This also means that Desktop Remote works on all major desktop platforms.

1.2 Background

This venture is associated to develop an android desktop remote application unlike other contemporary applications such as Team viewer is a web 2.0 program that allows user to connect to each other via internet and control his/her screen. Whereas AIO Desktop Remote App differentiate itself by providing maximum control in terms of the parameters like Live Screening, Touchpad Control, File transfer, file downloading, Image viewer, Media Player, Presentation and Power Off.

1.3 Objectives of the System

The project`s main objective is to remotely access the Desktop via any portable Android Device such as Tablet or Phone. All in One (AIO) Desktop Remote Application being compatible to all desktop operating systems i.e. Windows, Mac and Linux. This remote access will allow a portable device to access a target computer. Once you are connected remotely, you get access to all the resources on the computer. That means you get control over the mouse and the keyboard and by live screening option, the screen of the computer you've connected to is displayed. Additionally, unobstructed access to all the resources on the remote computer.

1.4 Significance of the System

For understanding, assume a condition when you must meet a close-fitting time limit and you can't do all the work sitting at your office location, but you need to work on your office computer. Wouldn't it be more feasible if one could stay at your home but still access your office computer remotely? That is, accessing the computer from a remote location.

The significance of our AIO app is that we can remotely access the system anywhere and anytime. This remote access will allow a portable android device to connect to a personal computer. When connection is established, you get access to the computer.

Figure 1.1 Overview of the system

The above figure shows that how portable android device (mobile app) remotely access and communicates with the system.This process is described as under

- Mobile app establishes a connection by a local router.
- When IP address of the system is entered, the session starts.
- After establishing the session with the system, AIO Desktop Remote App provides control in terms of the parameters like Live Screening, Touchpad Control, File transfer, file downloading, Image viewer, Media Player, Presentation and Power Off.

.

CHAPTER # 2

REQUIREMENT ANALYSIS

2.1 Introduction

This chapter covers requirements analysis of AIO Desktop Remote App to be developed. It contains functional and non-functional requirements.

2.2 Scope

The core of the project is to create a signal sharing medium between mobile and computer for different functionalities. Everyone in the corporate and educational field face the problem of communicating with PCs for multimedia control and also of data access on own PC when away from it. Desktop Remote is an all in one solution that enables the user to easily connect to a remote PC using an android device on the internet. The App accomplishes the following:

- It will be able to provide access to various Input/output functions such as keyboard, mouse/touchpad volume control etc.
- It will be able to control presentations/media playback.
- It will enable user to upload / download data from a remote PC.

2.3 Definitions, Acronyms and Abbreviations

AIO	All in one
UML	Unified Modeling Language
IP	Internet Protocol
GUI	Graphical User Interface
ERD	Entity Relationship Diagram
SDK	Software Development Kit
BS	Base station

2.4 Overall Description

2.4.1. Functional Requirements. It defines what a software system is actually expected to do. The functional requirement is **describing the performance of the system** as it relates to functionality of the system. Following are the function requirements:

a. **Connect.** This function connects the android device to the PC who's IP address is entered for the connection.

b. **Touchpad.** In this function the user can get access to the touchpad of the system and can control the system through cursor and mouse keys of the system.

c. **Keyboard.** In this function the user can get access to the keyboard of the system and can control the system this.

d. **Live Screen.** In this function the user can view the live screen of the system and can control the system.

e. **File Transfer.** In this function the user can transfer or copy any file from system to android device.

f. **File Download.** In this function the user can transfer or copy any file from android device to system.

g. **Presentation.** In this function the user can control the power point slides running on the system.

h. **Image Viewer.** In this function the user can access image gallery of the device.

i. **Music Player.** In this function the user can play audio music on android device and the audio will be relayed through system's speaker.

j. **Power.** In this function the user can shutdown, restart or lock the system.

2.4.2. Non Functional Requirements. The **Non-functional Requirements** describes how the system works. These are the requirements that **elaborate a performance characteristic** of the system.

a. Mobile App should support Android As an operating System

b. RAM (1GB)

c. Android Version (2.1.0) or above

2.4.3 Operating Environment

AIO Desktop Remote app is based on client and a server. Clients are the users that use the app that is developed, and server is responsible for interpreting and analysing the results and to generate results at the end on the performance based.
The software's, OS and languages used are mentioned below

- Software: Android Studio, NetBeans
- OS: Android
- Languages: Java
- Android based cell.

2.4.4 Design and Implementation Constraints

2.4.4.1 Aim: The aim of this application is to provide following functionalities:

1. Data transfer.
2. Cursor and keyboard control.
3. Control and edit presentations.
4. Relay music.
5. Power off and restart system.

In this project there are three main phases:

Developing Mobile App:

The first and important module in this project is android based mobile app which is developed using java on Android Studio.

User Interface:

There will be an interface which may include:

1. Connect.
2. Live Screen.
3. Touchpad.
4. File transfer.
5. File Download.
6. Media Player.
7. Presentation.
8. Power Off.

Operation and Features:

1. This application is installed on the android platform of the user.
2. Can only support android cell other than that is not acceptable.

Server Side:

After developing mobile app the next step is to build a server side to control the functionality of the system through android app.

2.4.5 Assumptions and Dependencies

- Here assumption will be that we have cell phone that should have Android.
- Another Assumption is that GPRS or Wi-Fi should be active on that cell phone through which it will send data.

2.4.6 Performance Requirements

The performance requirements for the product are as follow:

1. Mobile phone/tablet running Android OS.
2. GPRS or WIFI must be on.

CHAPTER # 3

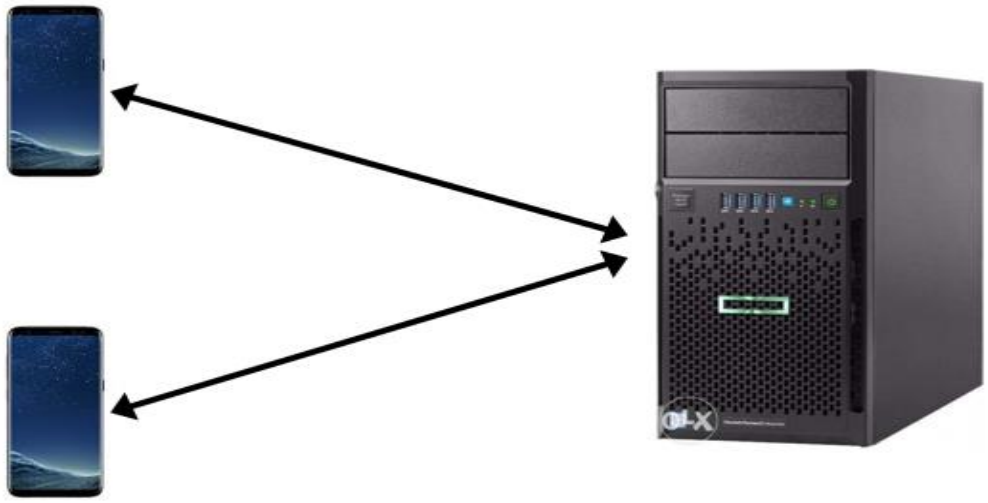
SYSTEM DESIGN

3.1 Introduction

This chapter describes the architecture and system design of the proposed "AIO Desktop Remote Application". It gives the overview of the system design, description and the relationship between different modules and their interactions. It is envisioned to guide the developers about the salients of the design of this application and its designing process. It will help the developers in the development, maintenance and usage of the Application.

3.2 System Architecture Description

This part describes an architectural overview of the system. This is a client/server system where mobile is a client and the end system is server. It is a relationship between two devices in which the client requests a service from the server.



USER: Android Platform **End System**

Figure 3.0 Overview of system architecture

The registered user from an android platform can connect to and control features on an end system.

3.3 Overview of Modules

3.3.1 Establish Connection:

This function connects to the end system based on IP address.

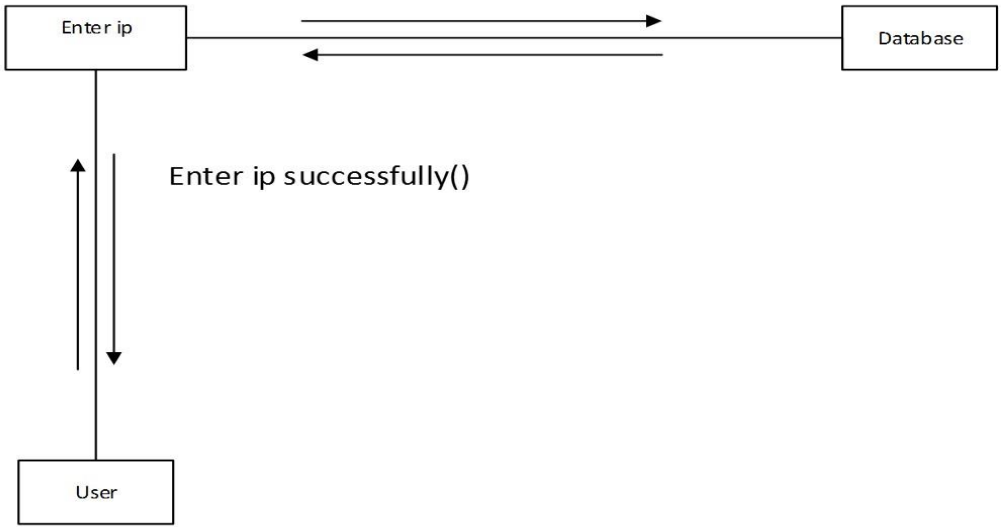


Figure 3.1 Establishing Connection

3.3.2 View Data:

After connection, user can view every kind of data from drives.

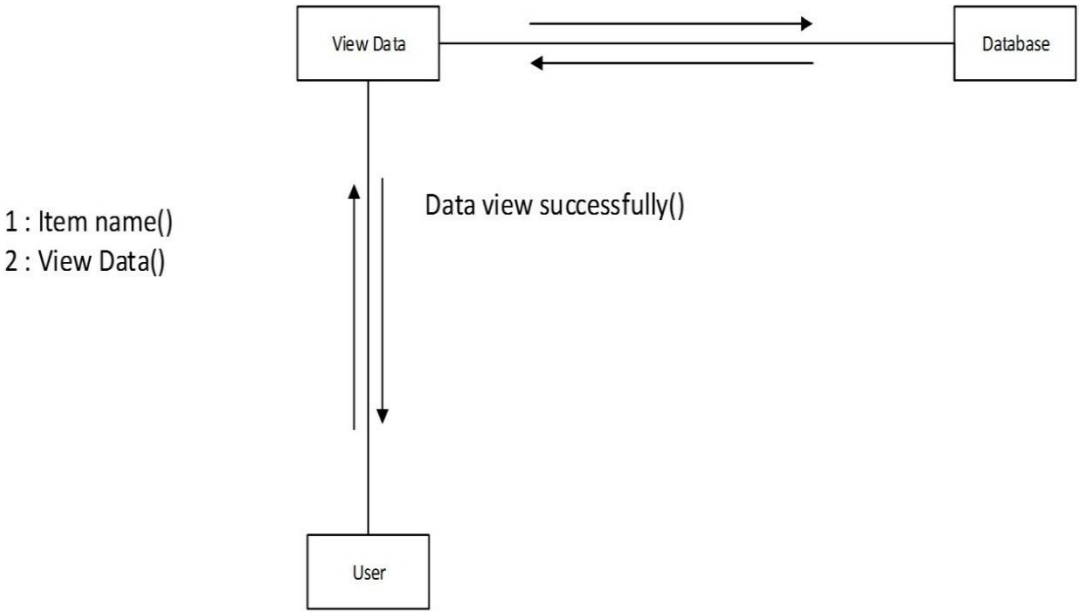


Figure 3.2 View Data

3.3.3 File Transfer:

User can transfer file

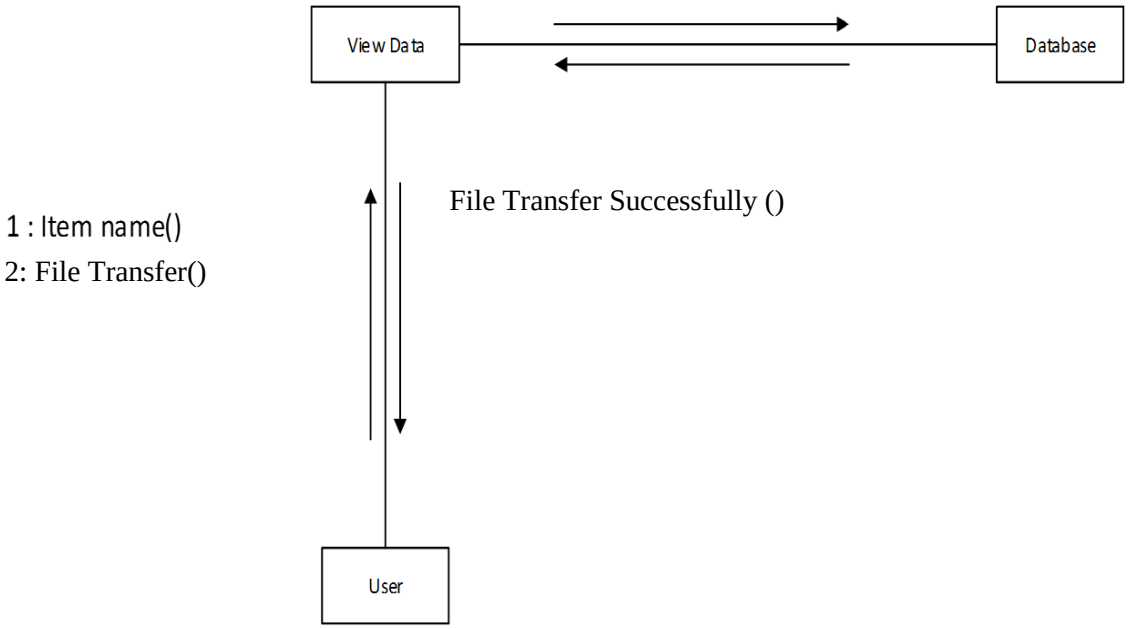


Figure 3.3 File Transfer

3.3.4 File Download:

User can download file from drives.

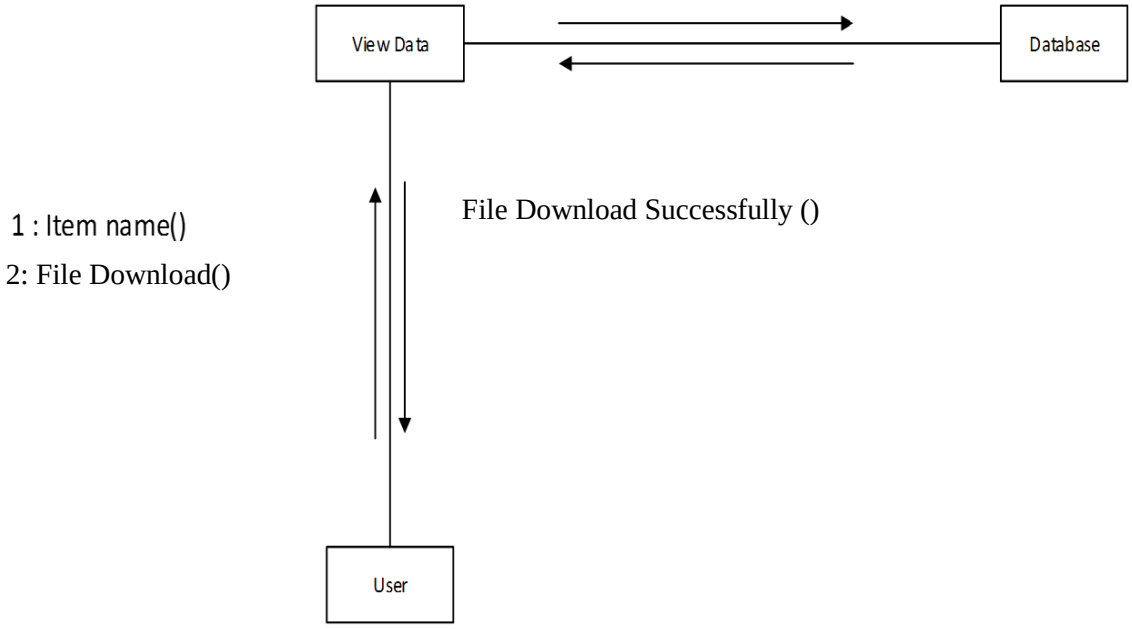


Figure 3.4 File Download

3.3.5 Music Player:

Enables the user to access and play media on client/server.

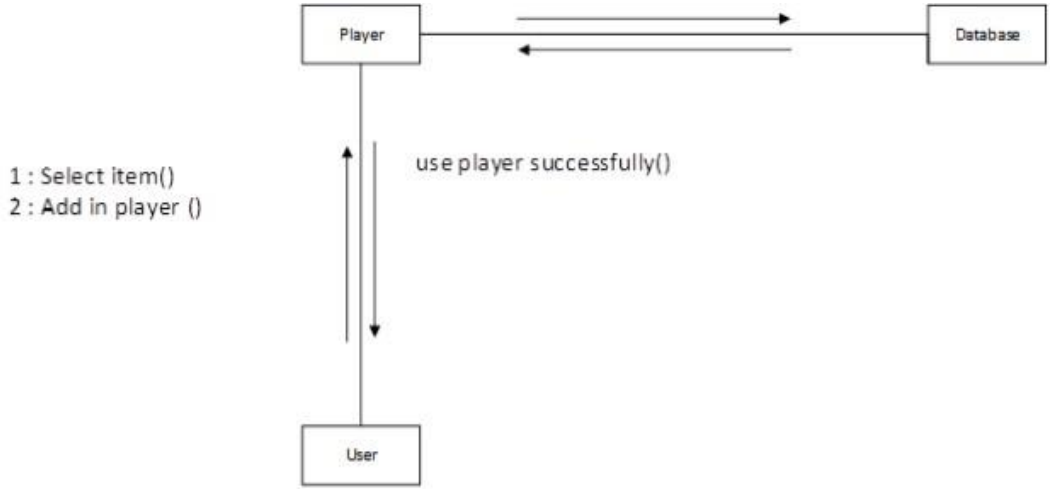


Figure 3.5 Music Player

3.3.6 Power Off:

Enables the user to power-off, restart and lock the end system.

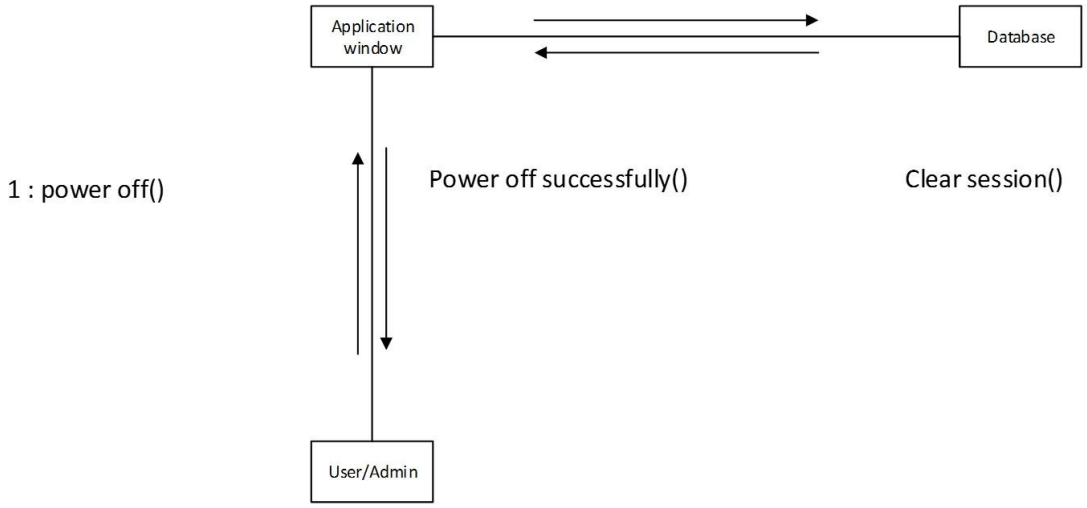


Figure 3.6 Power Off

3.3.7 Live Screen:

The user gets the live screen on the android platform.

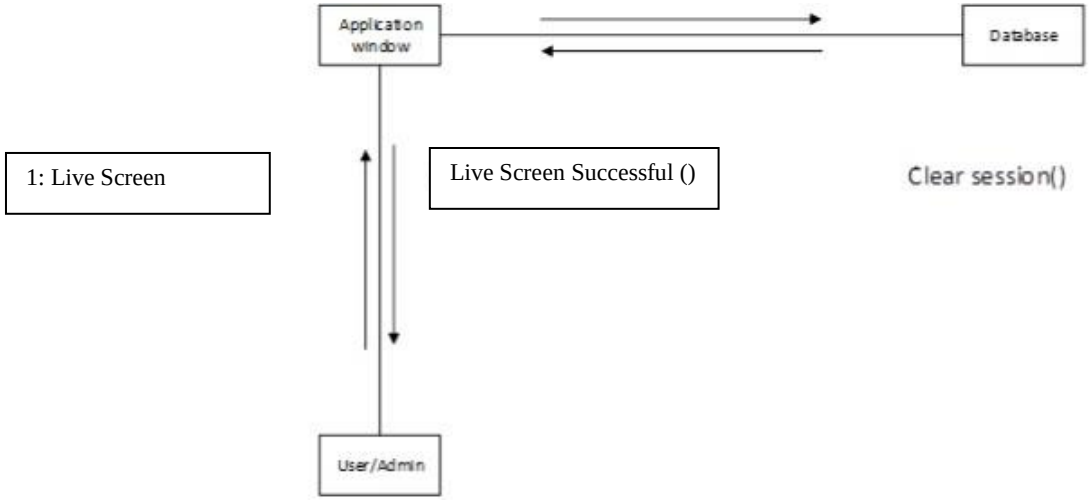


Figure 3.7 Live Screen

3.3.8 Touchpad:

The user accesses the touchpad of system on the android platform.

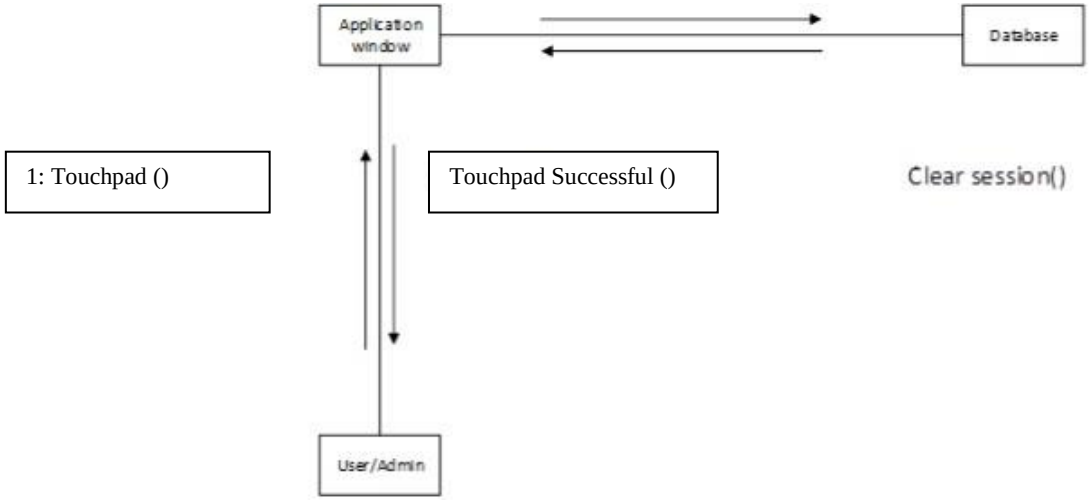


Figure 3.8 Touchpad

3.3.9 Keyboard:

The user accesses the keyboard of system on the android platform.

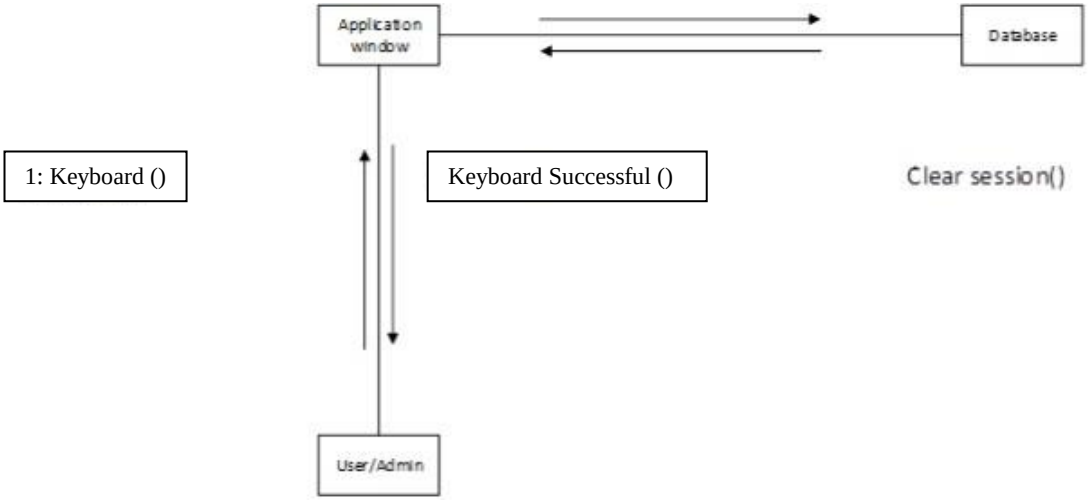


Figure 3.9 Keyboard

3.4 Structure and Relationships

This section covers the overall technical description of AIO Desktop Remote App. It shows the working of application in perspective of different viewpoints and shows relationships between different components.

3.4.1 Static View

3.4.1.1 Use Case Diagram

Following diagram shows course of events that takes place during the user and system interaction.

Use Case Diagram

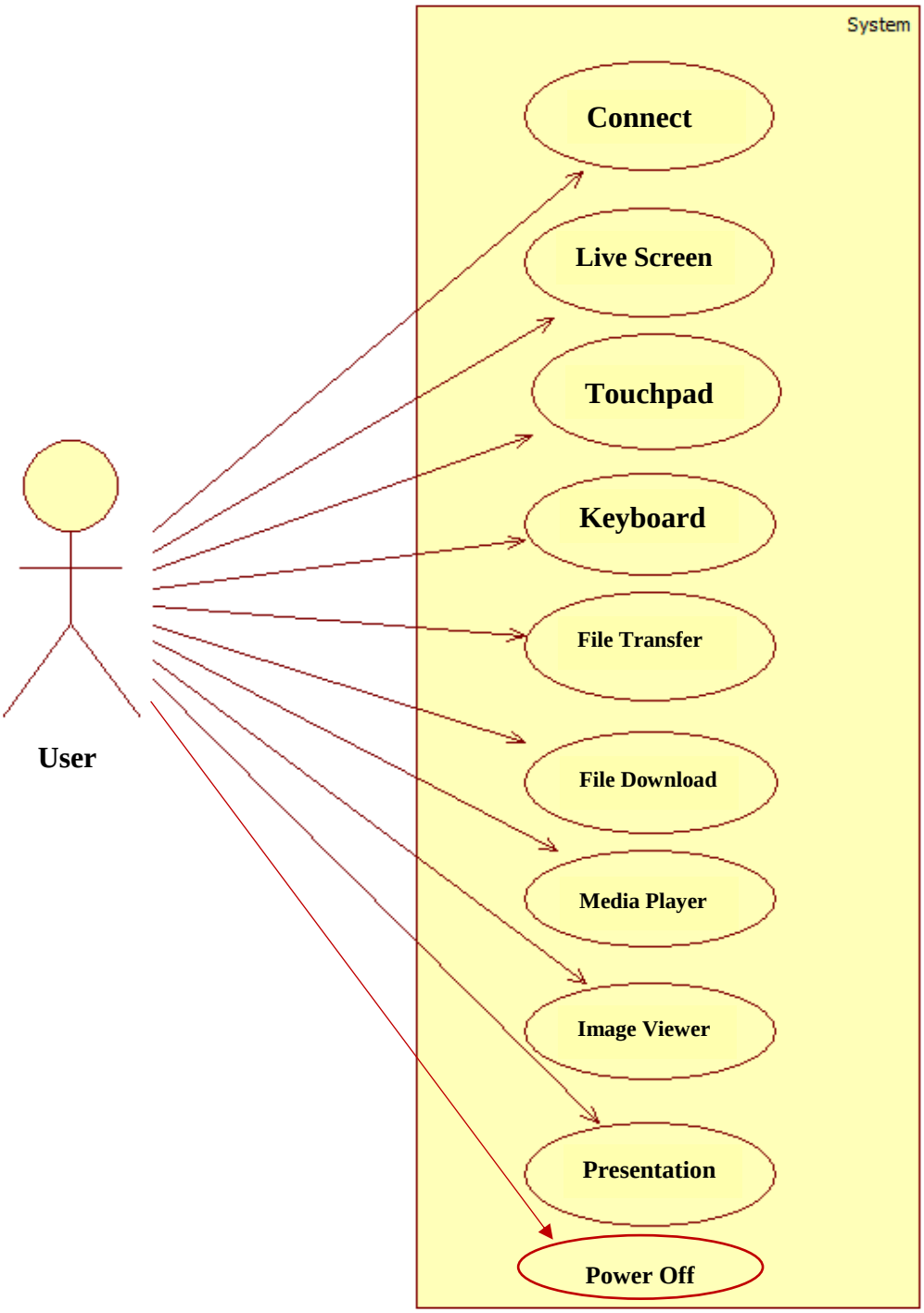


Figure 3.10 Use Case Diagram of Application

3.4.2 Use Case Descriptions

- **Connect:**

<table>
<tr><td>Use Case Title</td><td>Connect</td></tr>
<tr><td>Use Case Id</td><td>1.0</td></tr>
<tr><td colspan="2">Description:

This function connects the android device to the PC whose IP address entered for the connection.</td></tr>
<tr><td colspan="2">Pre-Conditions: Have an IP address and Port Number.</td></tr>
<tr><td>Task Sequence</td><td>Exceptions</td></tr>
<tr><td>1. User will enter the IP address and the Port Number.</td><td>• Incorrect IP address is entered.
• Incorrect Port Number is entered.</td></tr>
<tr><td>2. System will validate the IP address and the Port Number.</td><td>• User already logged in.
• IP address did not match.</td></tr>
<tr><td colspan="2">Post Conditions:

• User has been successfully connected.</td></tr>
<tr><td colspan="2">Actors:

• User</td></tr>
</table>

- **Login:**

Use Case Title	**Touchpad**
Use Case Id	1.1
Description: This function ensures that user at android end can get access to the touchpad of the system and can control the system through cursor and mouse keys.	
Pre-Conditions: Have an established connection.	
Task Sequence	**Exceptions**
1. User will open menu and select touchpad button.	• User not connected.
2. Access Touchpad on android device.	• System disconnected.
Post Conditions: • User has successfully accessed touchpad and mousekeys.	
Actors: • **User**	

- **Live Screen:**

Use Case Title	**Live Screen**
Use Case Id	1.2
Description: This function ensures that user can get access to Live Screen of the system connected.	
Pre-Conditions: Have an established connection.	
Task Sequence	**Exceptions**
1. Click Live Screen button on top right.	• User not connected.
Post Conditions: • User has successfully accessed Live Screen.	
Actors: • **User**	

- **File Transfer**:

Use Case Title	**File Transfer**
Use Case Id	1.3
Description: This function ensures that user can select and copy any file from system to android device.	
Pre-Conditions: View data	
Task Sequence	**Exceptions**
1. User can view and select data.	• Internet is not available.
2. User can copy data.	• Not enough space on android platform.
Post Conditions: • User has successfully transferred the file.	
Actors: • **User**	

- **File Download**:

Use Case Title	**File Download**:
Use Case Id	1.4
Description: This function ensures that user can select and copy any file from android device to system.	
Pre-Conditions: View data	
Task Sequence	**Exceptions**
1. User can view and select data.	• Internet is not available.
2. User can copy data.	• Not enough space on system drive.
Post Conditions: • User has been successfully copied.	
Actors: • **User**	

- **Presentation**:

Use Case Title	**Presentation**
Use Case Id	1.5
Description: This function ensures that user controls the power point slides running on the system.	

Pre-Conditions: User have an established connection.	
Task Sequence	**Exceptions**
1. User run the slideshow on the system.	• .Power point not open on system.
Post Conditions: • User has successfully controlled the slides.	
Actors: • **User**	

- **Media Player**:

Use Case Title	**Media Player**
Use Case Id	1.6
Description: This function ensures that users can access, run and relay audio files on client server.	
Pre-Conditions: User wants to play the music	
Task Sequence	**Exceptions**
1. User wants to listen the music	• Connecton not established.
Post Conditions: • User has successfully relayed the audio to/from system.	

Actors: • **User**

- **Power Off**:

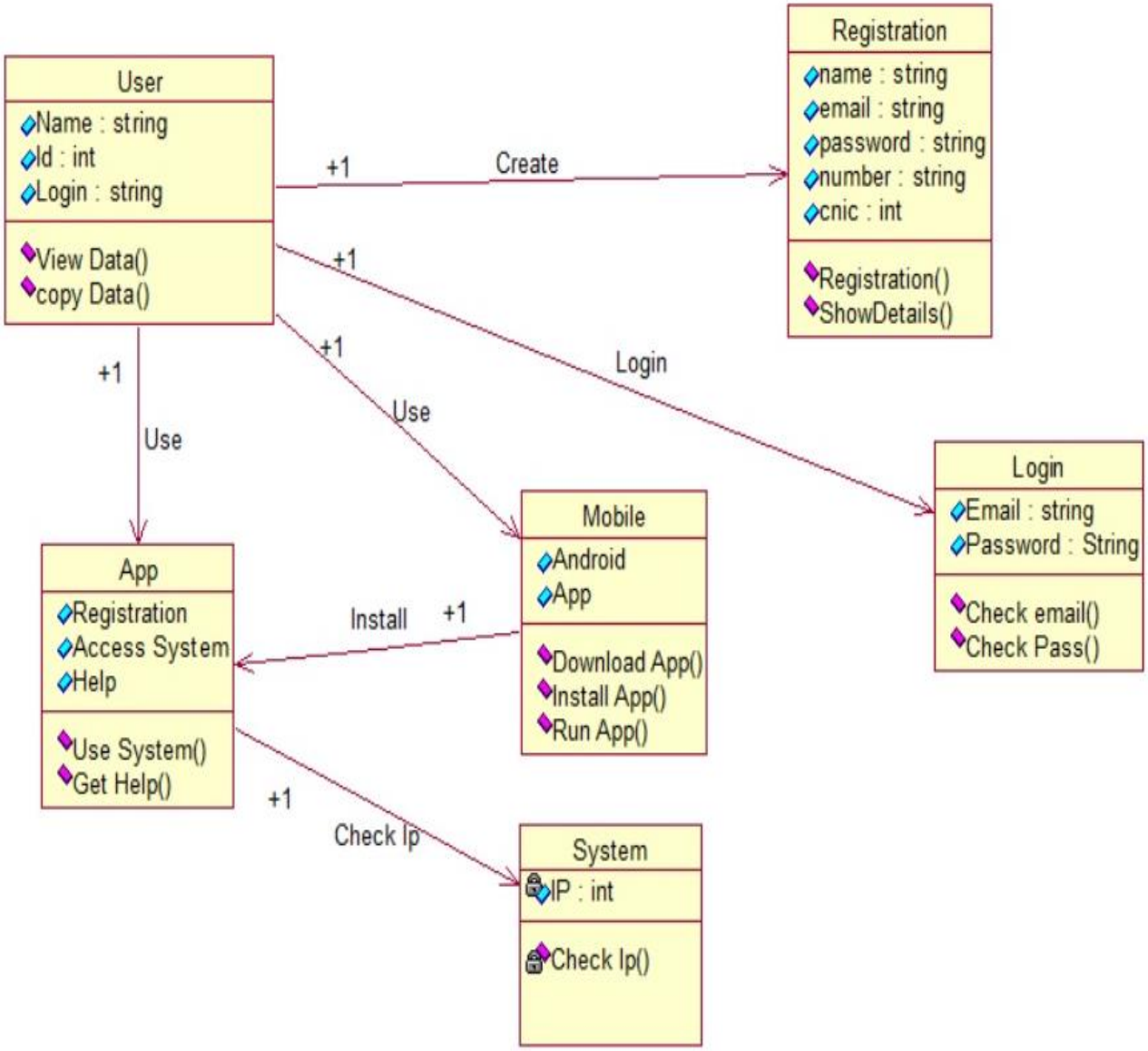
Use Case Title	**Power Off**
Use Case Id	1.7
Description: This function ensures that user can shut down, restart or lock the system.	
Pre-Conditions: User have an established connection.	

Task Sequence	**Exceptions**
1. Select power off from the slide menu.	• System already locked.

Post Conditions: • System has been shut down. • System have been restarted. • System is locked.
Actors: • **User**

3.4.2 Logical View Point of the System

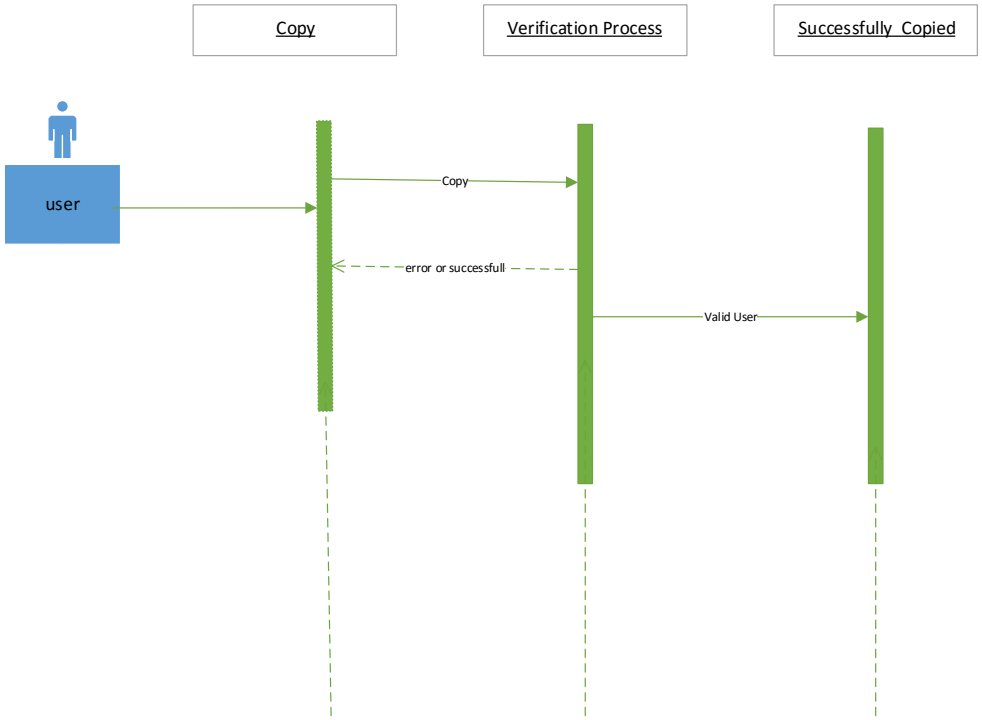
Class Diagram



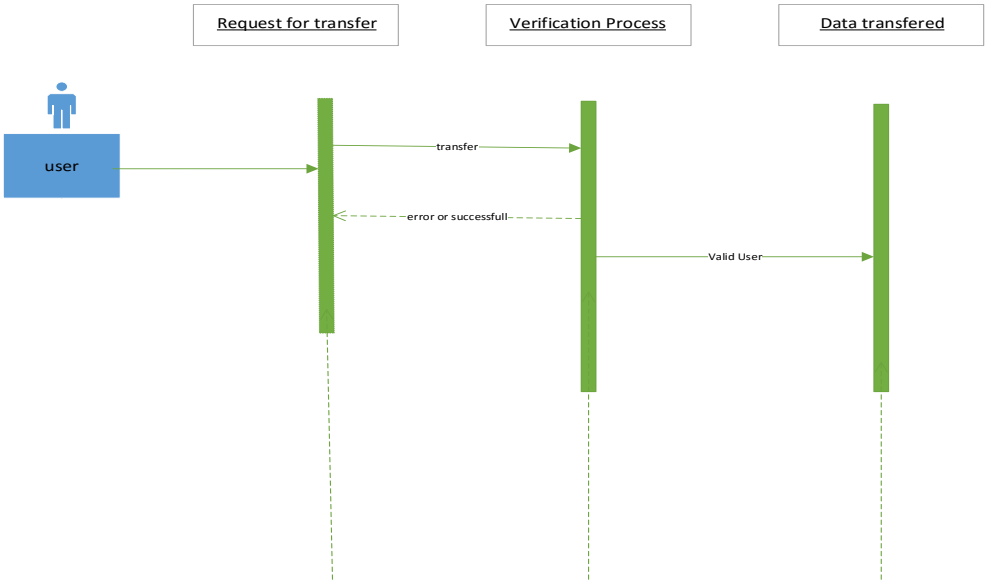
3.4.2 Dynamic View

Sequence Diagrams

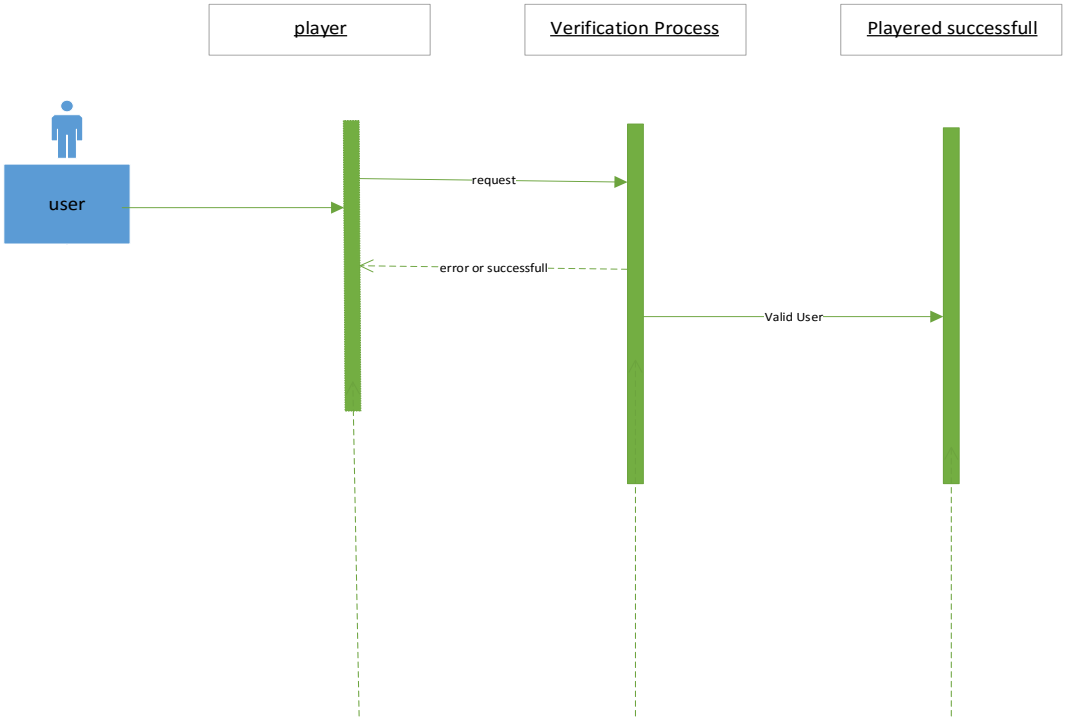
3.4.2.2 File Copy:



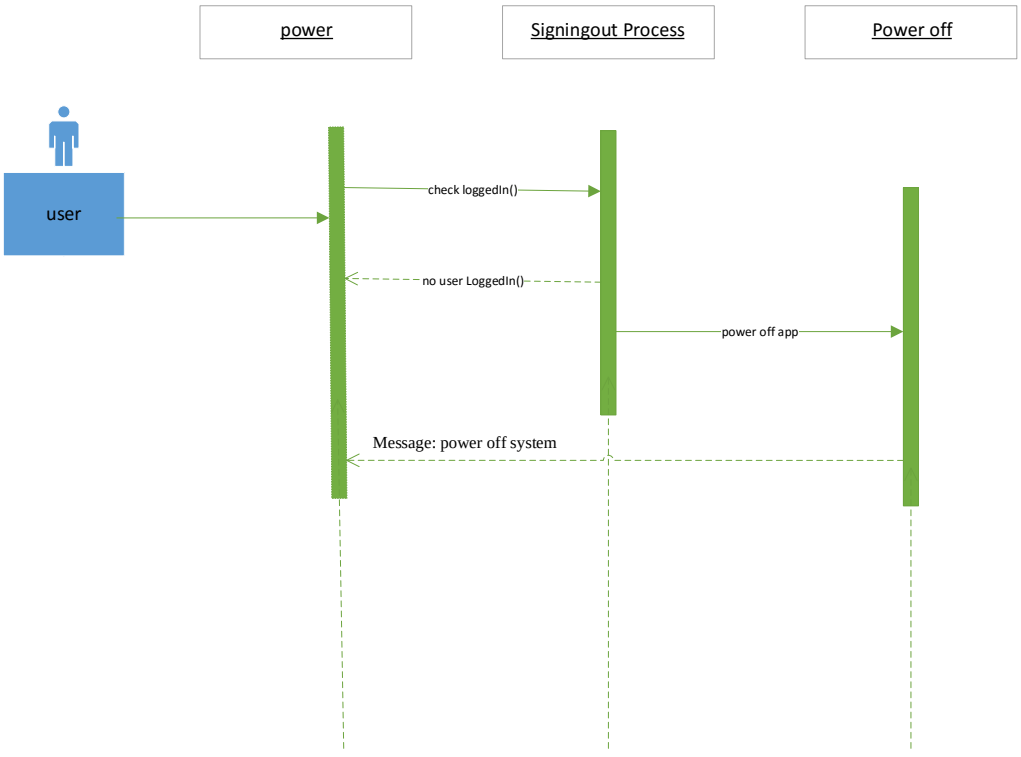
3.4.2.3 File Transfer:



3.4.2.4 Play Media:



3.4.2.5 Power Off:



CHAPTER # 4

USER INTERFACE

4.1 Introduction

This chapter comprises of **Graphical User Interface** (**GUI**) of AIO Desktop Remote Application that gives permission for user interaction with android devices through graphical based icons and visual clues such as secondary notation, in place of text-based user interfaces, written command labels and textual navigation.

For designing the visual composition the behavior of the user interface have an important role to play in the area of human–computer interaction. The visual languages introduced in the design is well-synchronized to the tasks to be performed.

4.1.1 GUI Work:

A GUI uses icons, windows and different menus to implement commands, such as viewing, and copying files. Even though various OSs are navigated and controlled through a cursor or by using keyboard shortcuts or arrow keys.

4.1.2 Benefits of GUI:

Dissimilar to a command line operating system like UNIX or MS-DOS, GUI operating systems are much user friendly because commands do not need to be learnt. Moreover, there is not requirement to know any programming languages. Hence it is the ease of usage, GUI OS have become the dominant OS in today's digital world.

4.1.3 Examples of a GUI:

- MS Windows
- Apple and Android OS
- Chrome OS
- Linux variants like Ubuntu

4.1.4 Desktop Remote App Interface:

User friendly platform to access some basic features of their personal computer and create a signal sharing medium between mobile and computer for different functionalities.

4.1.5 Open App Interface:

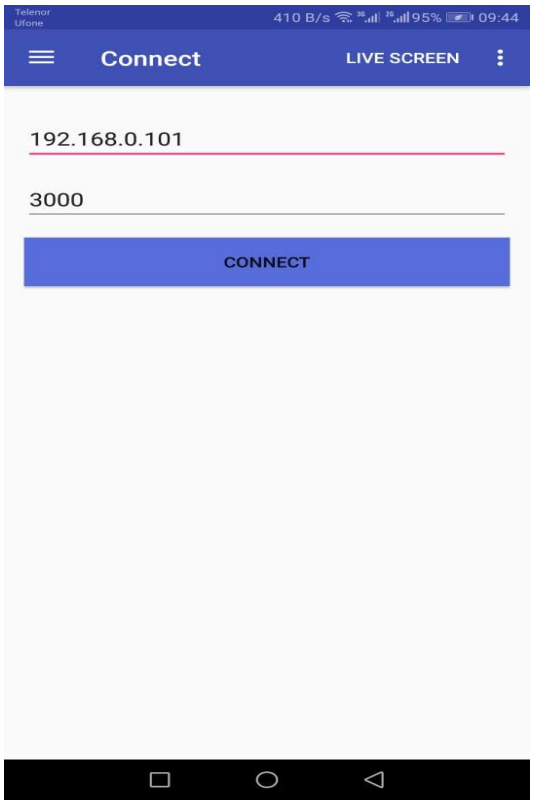


Figure 4.1 Initial Loading screen

4.1.6 Establish Connection Interfacce:

Figure 4.2 Establish Connection Interface

4.1.7 File Transfer Interface:

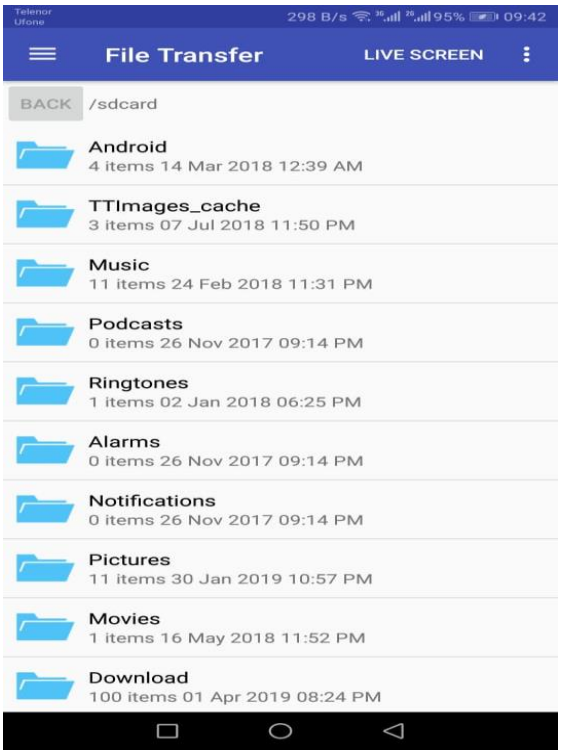


Figure 4.3 File Transfer Interface

4.1.8 File Download Interface:

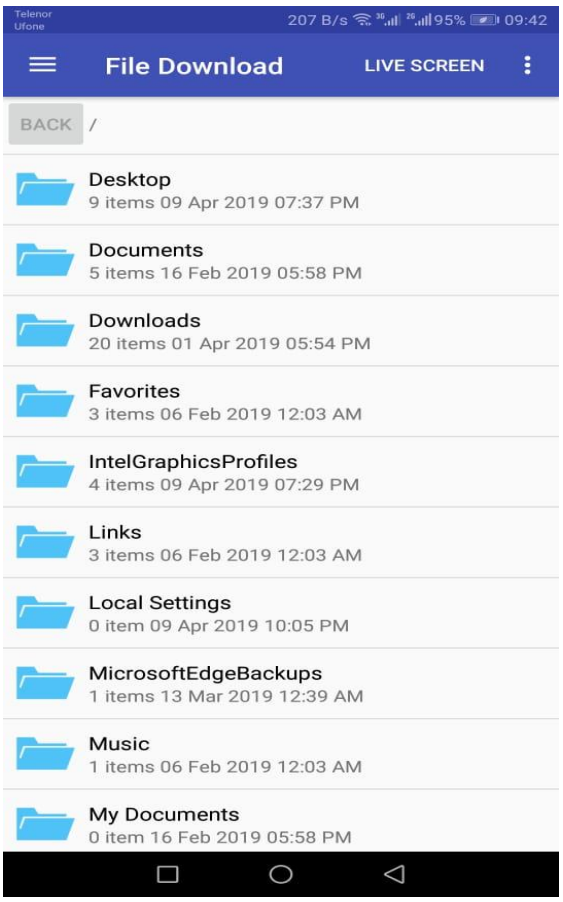


Figure 4.4 File Download Interface

4.1.9 Presentation Interface:

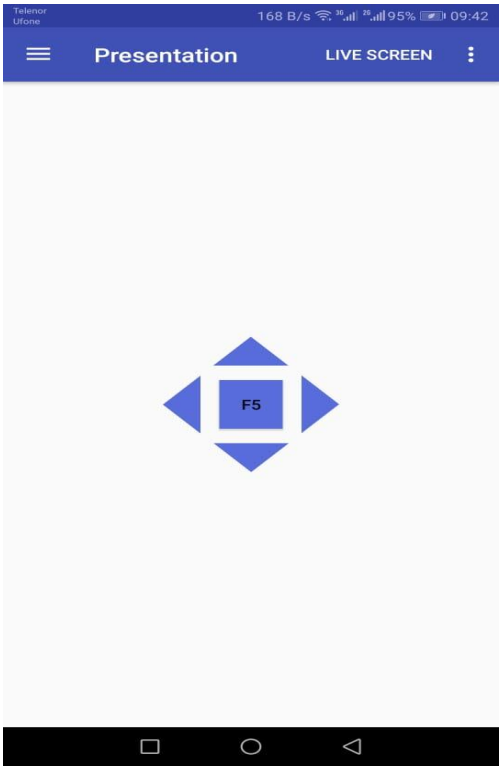


Figure 4.5 Presentation Interface

4.1.10 Power Off Interface:

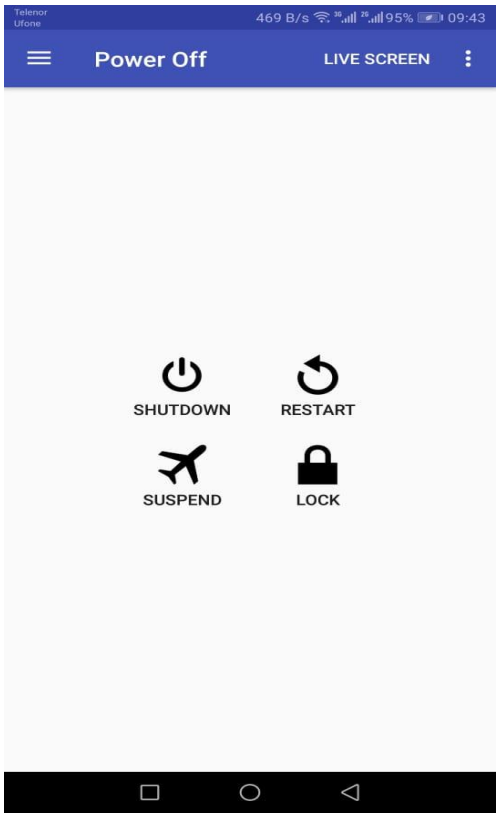


Figure 4.6 Power Off Interface

4.1.11 Live Screen Interface:

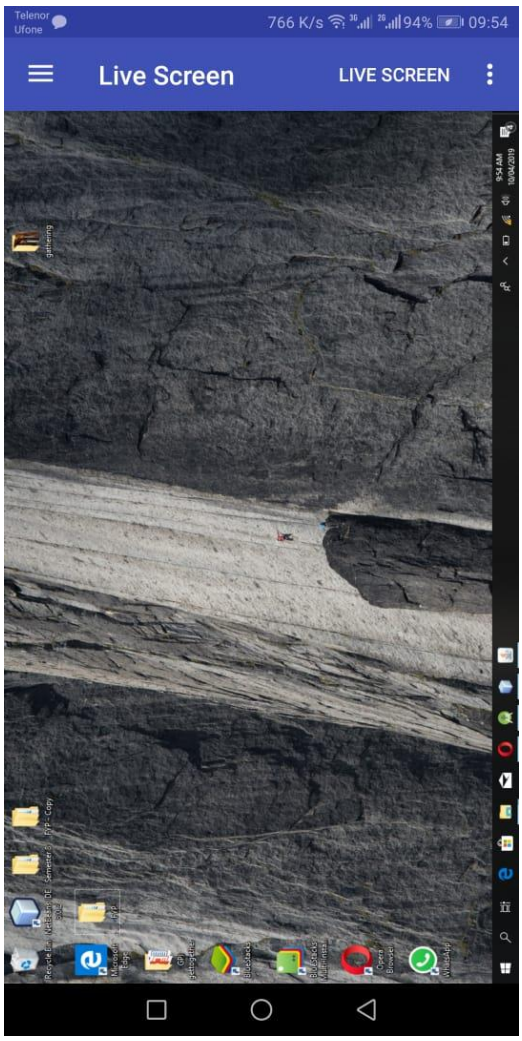


Figure 4.7 Live Screen Interface

CHAPTER # 5

QUALITY ASSURANCE

5.1 Introduction

Quality assurance(QA) discusses events which are implemented in a quality system so that the requirements for a product be fulfilled. It is the systematic approach for monitoring of processes and the related response loop that confers error anticipation. This can be compared with quality control, which is engrossed on process yields.

It concerns about the quality assurance and analyzing the quality testing in detail discussed under subsequent headings. The test cases of each module are discussed in detail, as to which components of the software to be tested and not to be tested and the testing. Other than that, it describes the audience for which the document is intended for. The document describes the appropriate methodologies and strategies used to plan and manage testing of the "AIO Desktop Remote Application".

5.2 Test Items

The system encompasses of multiple concepts that are combined together. It is very significant that every component of the system is methodically tested before it is delivered. The test items to be tested are mentioned below:

- **Database**
  The database is tested to make sure that all the values are correctly read, updates or deleted.
- **All the Modules**
  All the modules were completely tested to make sure that its all functional and non-functional requirements are met.
- **All the components**
  To ensure that it is working properly.

5.3 Features To Be Tested

It was made sure that all the features are tested and no feature was left behind. Following are the features:

- Open App
- Connect
- Files Transfer

- Files Download
- Image Viewer
- Media Player
- Presentation
- Power Off
- Live Screen
- Touchpad Control

5.4 Features Not to be Tested

We will not test the appearance of the system as it is not the functional need of the interface (Theme Testing).

5.5 Test Approaches

Manual Testing will be followed which comprises testing the software manually, not actually using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. Each Unit will be tested separately and then will be integrated with others, therefore, Unit Testing and Integration testing will be followed. For each unit, Black box Testing is done and for combined units Acceptance Testing is done. Test Approaches used in test execution are as follow:

- **Usability Testing**

  The overall flow of system is tested to check its usability by the Stakeholders and developers. Furthermore, all interfaces and interaction between these were carefully tested to see the ease of use.
- **Unit Testing**

  The Module/Unit testing phase is supported out to ensure that all the functional requirements of each module works correctly. This phase is repeated while testing each module.
- **Integration Testing**

  All the modules were tested for the way they interact with one another. Components of modules must be working efficiently with one another. The components of modules which are dependent on eachother must work without fault.

5.6 Item Pass/Fall Criteria

All the details of the test cases are specified in section Test Deliverables. Following the principles, a test item would be refereed as pass or fail.

- All pre-conditions are met
- All I/Os are passed out as stated
- When the result are same as what is specified in output => Pass
- When the system doesn't work or results not same as specified output => Fail.

5.7 Suspension Criteria and Resumption Requirement

If any bugs are found that can be evaluated and then fixed by the developers then there is no need to start the testing process from the beginning. But, when major bugs will block some test cases as they are interdependent, then the testing must be paused.

Major Error or bugs will be removed and then testing will be started again.

5.8 Test Deliverables

Following are the test cases:

User Interface Testing:

Test Case Number	01
Test Case Name	Open AIO Desktop Remote App.
Description	Testing Application whether it runs on cell.
Testing Technique	Unit testing, Black Box Testing
Preconditions	Application must be installed on android cell.
Input Values	Click on Application icon
Steps	• Open android cell. • Go To AIO Desktop Remote App app icon. • Click on the app icon.
Expected output	AIO Desktop Remote App will open.
Actual output	Application opened.
Status	Test case passed successfully.

Table 5.1 Open AIO Desktop Remote App

Test Case Number	02
Test Case Name	Connect
Description	User will connect to a system.
Testing Technique	Unit testing, Black Box Testing
Preconditions	User must have opened AIO Desktop Remote App.
Input Values	Click on “Connect” Button giving port number and ip address.
Steps	• Open AIO Desktop Remote App. • Then Click “Connect”
Expected output	User will have to enter credentials on the new screen.
Actual output	Connection established.
Status	Test case passed successfully.

Table 5.2 User establishes Connection

Test Case Number	03
Test Case Name	File Transfer
Description	User will transfer the desired file from mob to PC.
Testing Technique	Unit testing, Black Box Testing
Preconditions	User must have established connection.
Input Values	Select file to be transferred to PC.
Steps	• Open AIO Desktop Remote App.
Expected output	File transferred to PC.
Actual output	File transferred to PC.
Status	Test case passed successfully.

Table 5.3 User transfers files

Test Case Number	04
Test Case Name	File Download
Description	User will download the desired file from PC to mob.
Testing Technique	Unit testing, Black Box Testing
Preconditions	User must have an established connection.
Input Values	Select file to be transferred to mob..
Steps	• Open AIO Desktop Remote App. • Click on the File Download Button
Expected output	File transferred to mob.
Actual output	File transferred.
Status	Test case passed successfully.

Table 5.4 User downloads file

Test Case Number	05
Test Case Name	Presentation
Description	User will be able to control the presentation slides.
Testing Technique	Unit testing, Black Box Testing
Preconditions	User must have an established connection.
Input Values	Click on Presentation button.
Steps	• Open AIO Remote Desktop App. • After Login, Click on the Presentation.
Expected output	Presentation will be controlled by App.
Actual output	Slides will be controlled.
Status	Test case passed successfully.

Table 5.5 User controls presentation

Test Case Number	06
Test Case Name	Power Off
Description	User will be able to control power of PC.
Testing Technique	Unit testing, Black Box Testing
Preconditions	User must have an established connection.
Input Values	Click on "Power Off" button.
Steps	• Open AIO Remote Desktop App. • Click Menu button • Then Click "Power Off"
Expected output	Location will be shared.
Actual output	Location will be shared.
Status	Test case passed successfully.

Table 5.6 Power Off

Test Case Number	07
Test Case Name	Live Screen
Description	User will be able to access live screen of PC.
Testing Technique	Unit testing, Black Box Testing
Preconditions	User must have an established connection.
Input Values	Click on “Live Screen” button.
Steps	• Open AIO Desktop Remote App. • Click on “Live Screen” button on the main screen.
Expected output	Live screen of desktop computer on mobile.
Actual output	Live Screen available.
Status	Test case passed successfully.

Table 5.7 Live Screen

5.9 Responsibilities, Staffing and Training needs

- **Responsibilities:**

The developers are responsible for all components testing and integration testing tasks. Developers are the testing team that are testing the system against the requirements.

- **Staffing and Training Needs:**

For testing the basic knowledge of different techniques and strategies is required to be known.

Techniques such as Black Box testing, integration testing should be known to developers.

All the developers actively take part in the development process, furthermore they will test each other’s work and testing of the project concurrently.

5.10 Schedule

Important Dates:

- Unit Testing and integration testing will be finished by the start of April as will Development process
- Acceptance Testing will be performed right after the Development process completed i.e in first week of May.

5.11 Risk and Contingencies

- **Programmatic Risks:**

Due to the constraints of the project, limitations will be there in the scope of the project in case of a programmatic risk. Programmatic Risk can also be caused due to external events.

Training: Training employees to deal with situations that might occur.

- **Operational Risks:**

The cause of Operational risk are No communication in the group or failure to address priority conflicts. It will be eradicated by arranging regular meetings and keeping time lines to meet the goals of the project.

- **Technical risks:**

Technical risks generally lead to failure of functionality and performance. They will be eradicated by keeping the once defined requirements constant.

- **Schedule Risk:**

When the project tasks and schedule release risks are not addressed properly then Project schedule get slip and lingers on. The project might get delayed from its expected schedule so in order to avoid this we will need to increase the hours/day that the project is being worked on.

CHAPTER # 6

USER MANUAL

6.1 Introduction

This user manual guide will provide required description for all the platforms and tools which are used in the development of this project and the procedure to install them. This manual will summarize the way to install All in One (AIO) Desktop Remote Application on android device and required IDEs on the computer system.

6.2 Installation Guide

6.2.1 Application Installation

Keeping in view the financial and project scope limitations this application will not be available on a public source or any website or on Google Play Store but will only be installed through android studio and an apk file. This application will be installed through USB Debugging Mode in developer's options on an android device.

- For android studio to recognize your device as a target for deploying debuggable APKs, you must first enable USB debugging mode in the developer options.

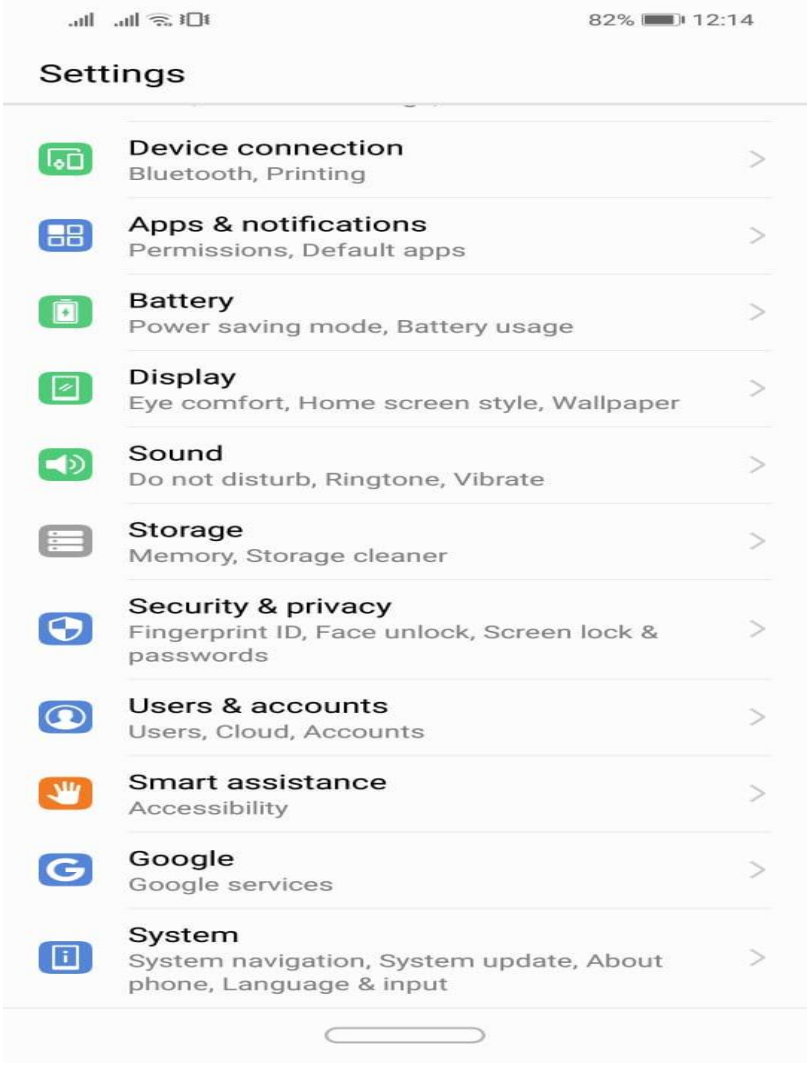


Figure 6.1 Application Installation Process

- Depending on the version of Android you're using, proceed as follows:
- On Android 8.0 and higher, go to Settings > System > About phone and tap Build number seven times.
- On Android 4.2 through 7.1.2, go to Settings > About phone and tap Build number 7 times.

Figure 6.2 Application Installation Process

- Now return to the main Settings menu to find Developer options at the bottom.

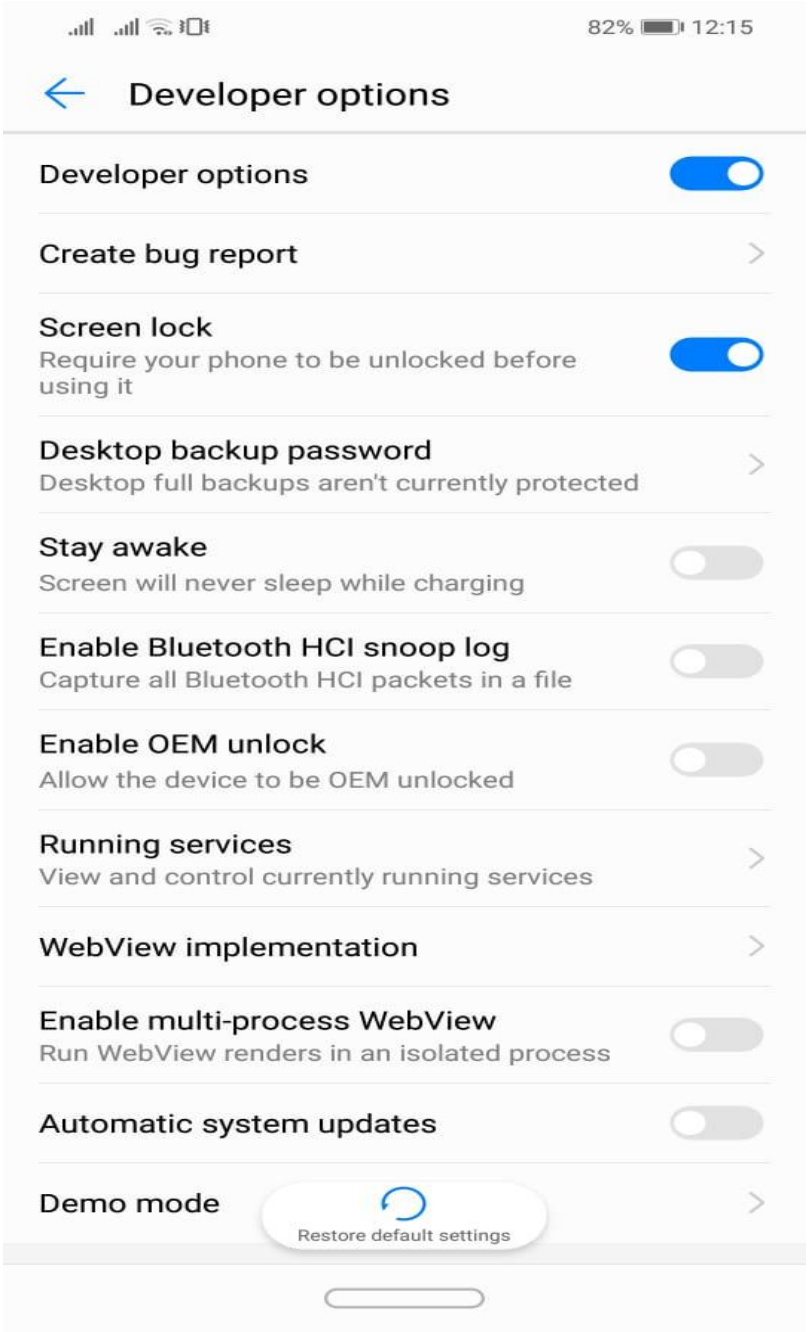


Figure 6.3 Application Installation Process

- Go to the Developer options menu
- Scroll down and enable USB debugging mode

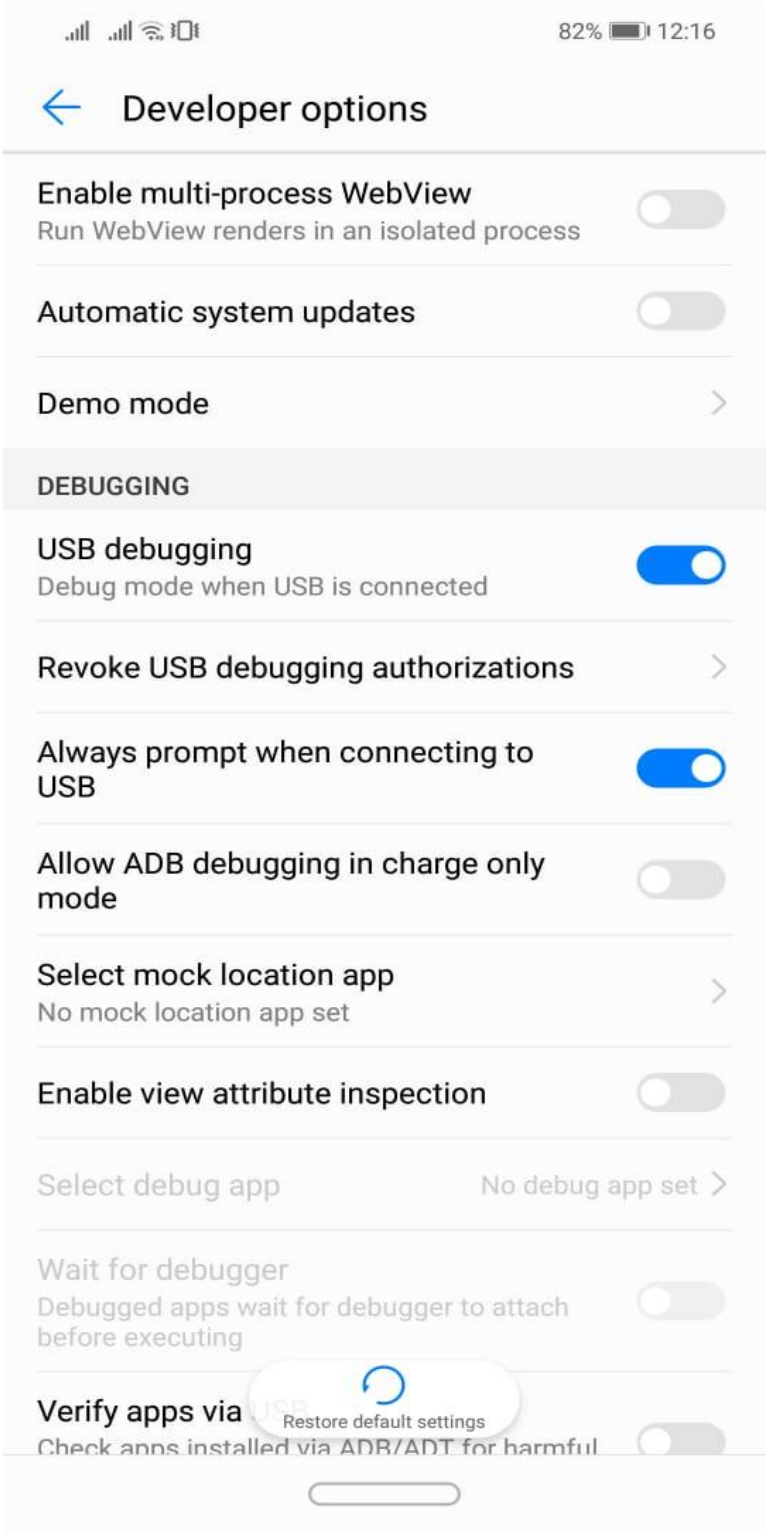


Figure 6.4 Application Installation Process

- Now connect mobile with the system running android studio
- On android device select transfer files on pop up menu

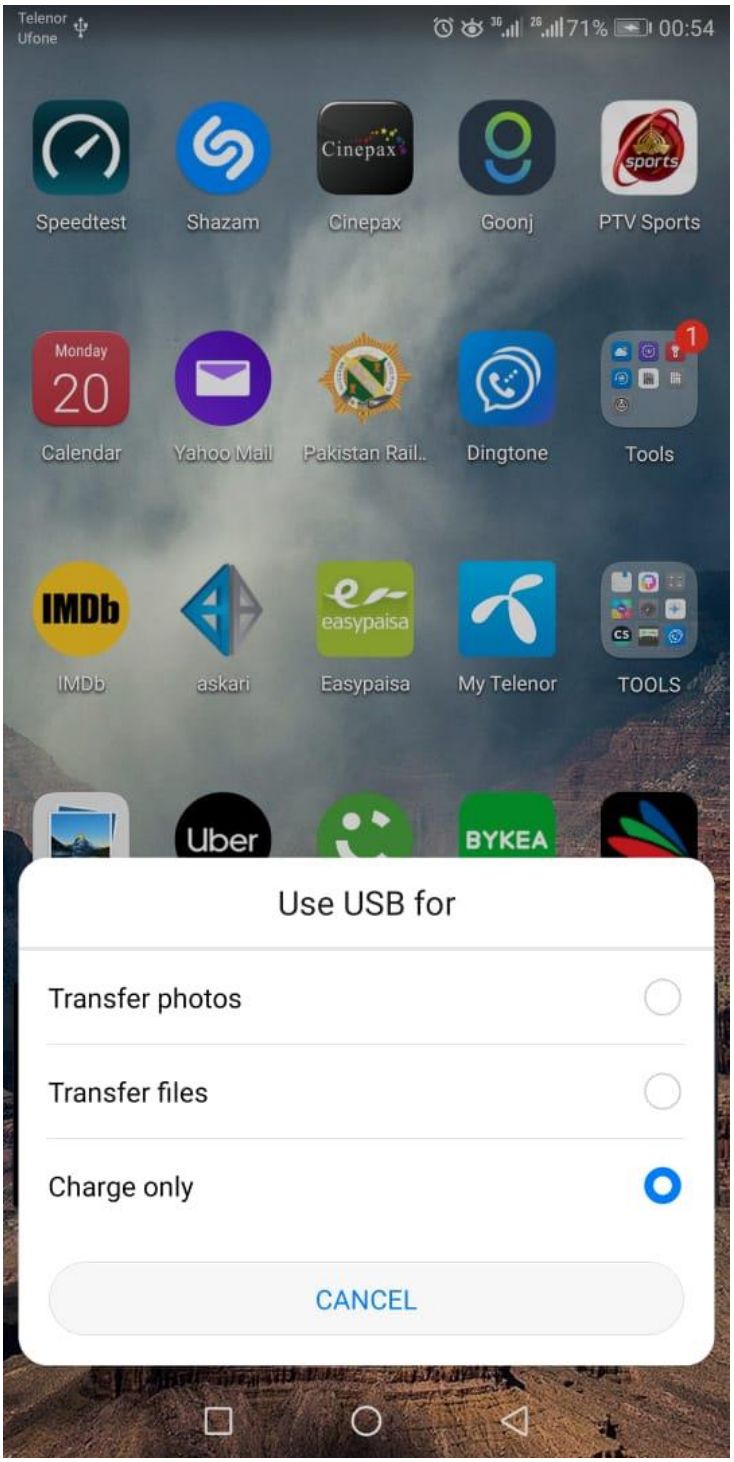


Figure 6.5 Application Installation Process

6.2.2 Desktop IDE Installation

- Install Netbeans 8.0.2 or above/ Install .exe file of desktop software

6.3 Hardware/Software Requirements

6.3.1 Hardware Requirements

- Mobile phone, tablet with android Operating System with an active internet connection.

6.3.2 Software Requirements

- Android Operating System: 3.1, API level 12, NDK 6 or above

6.3.3 Main Features of the Application

- Live screening of desktop system.
- Data transfer between mobile and desktop system.
- Controlling power point presentations
- Cursor and keyboard control of desktop system.
- Power control of desktop system.
-

6.4 Operating Manual of Application

6.4.1 Following steps to be followed:

- Open application by clicking on the icon.

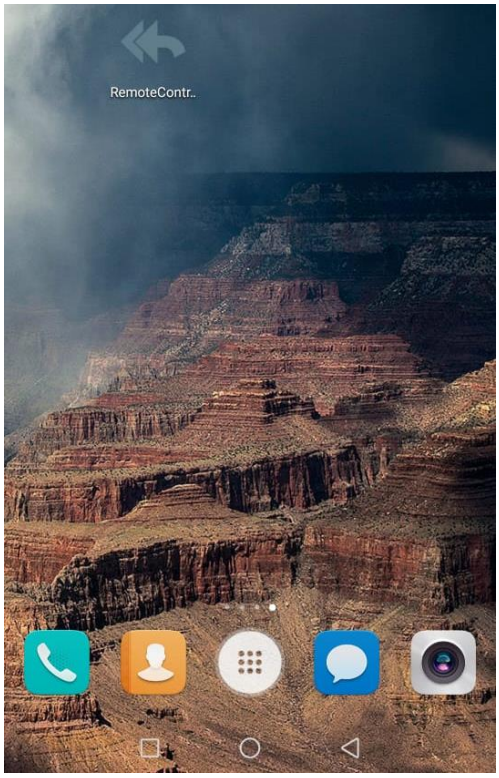


Figure 6.6 Application Operation

- Enter IP address and port number on Connect menu.

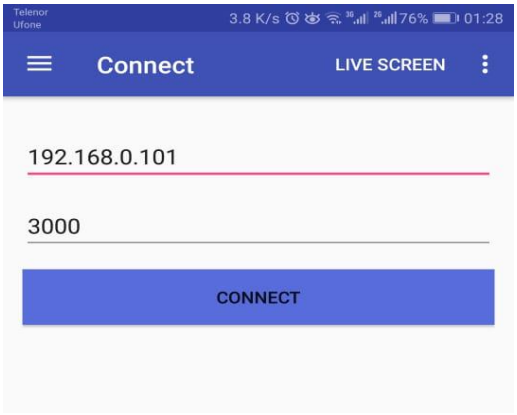


Figure 6.7 Application Operation

- Slide right on screen and select the required option.

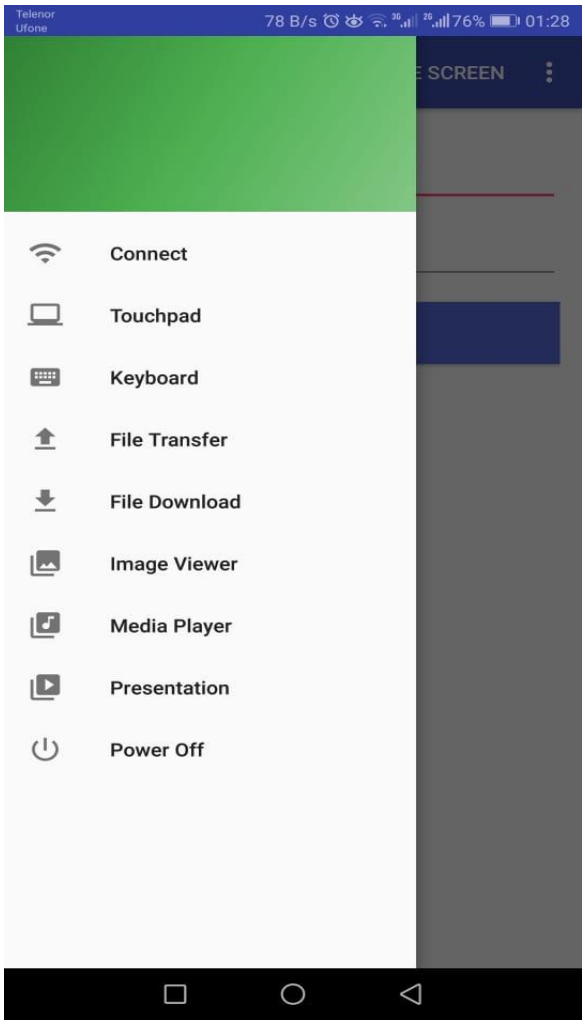


Figure 6.8 Application Operation

6.1 REFRENCES

- www.w3schools.com
- https://developer.android.com/
- https://netbeans.org/kb/articles/learn-java.html
- https://netbeans.org/kb/docs/java/gui-functionality.html
- https://www.homeandlearn.co.uk/java/the_netbeans_software.html

Appendix A - Project Timeline

Development Phase	180 Days						Duration (Days)
	0 – 30 Days	31 – 60 Days	61 – 90 Days	91 – 120 Days	121 – 150 Days	151 – 180 Days	
Requirement Gathering							18
Analysis							12
Design							70
Coding							90
Testing							28
Implementation Deployment							20
Documentation							170 (Parallel)
Total Time (Days)	30	30	30	30	30	30	180