

A
Project Report
On
“PC Controller Using Android Device”

Submitted by:

- | | |
|--|--------------------------------|
| 1. Atharv Milind Davale | (PRN: 2020032500183191) |
| 2. Abhijeet Balkrishna Surshetwar | (PRN 2020032500183392) |
| 3. Rushikesh Rajesh Waghule | (PRN:2020032500186525) |
| 4. Amrut Yogesh Virdhe | (PRN:2020032500185916) |
| 5. Digvijay Sambhaji Shinde | (PRN:2020032500185166) |

UNDER THE GUIDANCE OF

Mr. A.M. Dyade

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AT



**SHRI VITTHAL EDUCATION AND RESEARCH INSTITUTE
COLLEGE OF ENGINEERING, PANDHARPUR**

**AFFILIATED TO PUNYASHLOK AHILYADEVI HOLKAR SOLAPUR UNIVERSITY, SOLAPUR
2023 - 2024**



SVERI's COLLEGE OF ENGINEERING, PANDHARPUR

CERTIFICATE

This is to certify that the project report entitled "**PC Controller Using Android Device**" is submitted for partial fulfillment of Bachelor of Technology in Computer Science and Engineering as per requirement of Punyashlok Ahilyadevi Holkar Solapur University, Solapur for the academic year 2023-2024.

(Mr.A.M.Dyade)

Project Guide

(Mr. P. D. Mane)

Project Coordinator

(Dr. S. P. Pawar)

H.O.D. CSE

(Dr. B. P. Ronge)

Principal

External Examiner

Acknowledgement

We are pleased to acknowledge **Dr. S.P. Pawar** (HOD CSE) for her valuable guidance during the course of this project work. We extend our sincere thanks to **Mr. A.M.Dyade** who continuously helped us throughout the project and without his/her guidance, this project would have been an uphill task.

We are also grateful to other faculty members of the CSE Department and technical staff who cooperated with us regarding some issues.

I would also like to thank my parents and friends who helped me a lot in executing this project within the limited time frame.

SYNOPSIS

1. Synopsis of Project Work

(a) Relevance:

A project involving PC controller usage through Android devices offers a modernized approach to control systems compared to traditional methods. By digitizing the control process and storing data digitally, this system enhances management tasks over conventional paper-based methods. Just as blockchain presents new avenues for digital services, integrating Android devices as PC controllers offers innovative opportunities. However, current research often focuses on technical and legal aspects, neglecting the potential for advanced digital services. Utilizing blockchain technology in this context could mitigate fraud and expand access to controller functionalities. Through encrypted keys and tamper-proof personal IDs, the system ensures secure interactions. As electronic credibility services evolve, foundational tools like electronic signatures and authentication can pave the way for more sophisticated systems, particularly in projects like PC controller integration with Android devices.

(b) Present Theories and practices:

i. Agora:

The agora of wirelessly operating a computer, especially when its screen is projected onto a large display like a projector or television, encompasses a multitude of contexts, from classrooms to conference rooms, mobile workgroups, and even living rooms. In response to this demand, specialized devices have emerged. Wireless keyboards, utilizing Bluetooth or wireless USB mini-receivers, facilitate seamless communication between the keyboard and the computer, offering comfortable control without the constraints of cables.

ii. Wireless Control:

It is an Android App that accomplish live connection between computer and android device by using wifi for transmission using Socket programing. There are two sides like client and server, An Agent Creates server and wait for response from client and server

(c) Proposed work:

i. Scope of project:

The scope of the project involves the development of an Android application aimed at establishing a live connection between a computer and an Android device via Wi-Fi using Socket programming. The primary focus lies in creating a robust system comprising two distinct components: a server and a client. The server component, initiated by an agent within the Android app, is responsible for creating a server socket and listening for incoming connections from the client. Meanwhile, the client component connects to the server established by the agent and serves as the sender of data or commands to the server. Key functionalities include establishing and maintaining a reliable connection, transmitting data bidirectionally between the Android device and the computer, and ensuring security and stability throughout the communication process. The project encompasses tasks such as designing and implementing the user interface for the Android application, developing the server and client logic for socket communication, testing the application for functionality and performance, and refining the user experience based on feedback. Additionally, considerations regarding compatibility with various Android devices and operating systems, as well as potential scalability for future enhancements, are essential aspects of the project scope. Overall, the project aims to deliver a comprehensive solution that facilitates seamless and efficient communication between computers and Android devices over a Wi-Fi network.

ii. Objectives:

The project aims to develop an Android application that facilitates live communication between a computer and an Android device via Wi-Fi, utilizing socket programming for efficient data transmission. The primary objective is to create a robust client-server architecture within the app, allowing bidirectional communication between the Android device and the computer. This involves implementing a server component initiated by an agent within the Android app, which creates a server socket and listens for incoming connections from the client. Concurrently, the client component of the app connects to the server established by the agent, serving as the sender of data or commands to the server. Ensuring the stability, reliability, and security of the connection is paramount, with measures implemented to safeguard transmitted data and minimize downtime. Additionally, user interface design is crucial to providing an intuitive and user-friendly experience, enabling users to easily initiate and manage connections with the computer. Thorough testing, documentation, and ongoing support are essential aspects of the project to ensure optimal performance and user satisfaction.

The Main objectives are:

1. A Touch-screen mouse control with two onscreen mouse buttons.
2. Use in either portrait or landscape mode.
3. Works with any Wi-Fi network.
4. Keyboard typing capability.
5. Operating computer functionality and handling and modification capability
6. Support any Android operating system version.
7. Compatible with Mac, Windows and Linux operating systems.
8. Voice typing mechanism, on voice commands all the typing is done in desktop/laptop.
9. It can be used for showing live coding demo.

iii. Phase wise Proposed Work

Phase I	Phase II
Finalizing project topic	Coding
Requirement Collection	Organization support
Design the modules	Testing and Validation
Figma design	Deploying

2. Facilities Available:

- (a) Internet.
- (b) Required configured machine.
- (c) Required software

3. Approximate expenditure:

Cost estimation can be defined as the approximate judgment of the costs for a project. Cost estimation will never be an exact because there are too many variables involved in the calculation for a cost estimate, such as human, technical, environmental, and political. Furthermore, any process that involves a significant human factor can never be exact because humans are far too complex to be entirely predictable. Furthermore, software development for any fair-sized project will inevitably include a number of tasks that have complexities that are difficult to judge because of the complexity of software systems.

For college project, the approximate expenditure would likely be minimal, mainly comprising any costs associated with software tools or resources needed for development. Depending on the specific requirements of the project, such expenses could range from negligible to a few hundred dollars at most. For instance, if specialized software or licenses are required, students may opt for free or trial versions, open-source alternatives, or utilize resources provided by the college. Additionally, while testing and documentation are essential aspects of the project, students can minimize expenses by leveraging free testing tools and platforms, as well as creating documentation using readily available resources. Marketing and promotion expenses would typically be unnecessary for a college project, as the focus is on academic learning rather than commercial viability. Overall, the approximate expenditure for a college project would be modest, with students seeking cost-effective solutions to fulfill project requirements within budget constraints.

With the inclusion of the documentation and submission costs, the overall budget might now be adjusted to a range of around 1500 INR to 1800 INR, assuming the use of free resources and minimal hardware needs. If additional unexpected expenses are considered, setting a maximum budget of approximately 2000-2500 INR would be advisable to ensure all aspects of the project are covered without financial strain

ABSTRACT

From some years there has been a significant evolution in mobile or smartphone computing and communication devices like mobile phones, media players and many more. This project e is proposed to be able to perform most of the actions a normal computer keyboard and mouse can accomplish. Wireless presentation controller ensures good freedom of movement. but, most of such devices do not allow user full operation on the computer, like running the program, moving or closing an application window, etc This project proposed the design and implementation of converting smartphones into computer remote controllers by which user can wirelessly operate a computer. sWe proposed the system which can reduce the strain of sore moment with the use of computer. It enables the user to move cursor, click operation and applications, play with media such like forward, rewind, pause, run, and increase or reduce the volume of a media file, multi-touch scrolling, pinch gesture for zoom in and out. This could be achieved by useing the proposed application that occurs to the computer network via Wi-Fi which can connects both system with each other, then command from the mobile phone that remotely controls the computer.

With the rapid development of mobile terminals and wireless networks, the relationships between computers and mobile phones are more and more closely. Now, the mobile phones and computer become indispensable to modern society in the daily life. The information transmission and exchange between them has become increasingly frequent. The purpose of this research is to find a safe way of connection with low cost, low power consumption, convenient and flexible, so that we can finish the information exchange quickly and accurately between mobile phone and computer. Because of using a socket connection mode, the physical line media is not needed.

INDEX

<i>Title Page</i>	<i>i</i>
<i>Certificate</i>	<i>ii</i>
<i>Acknowledgement</i>	<i>iii</i>
<i>Synopsis</i>	<i>iv</i>
<i>Abstract</i>	<i>viii</i>
<i>Index</i>	<i>ix</i>
<i>Abbreviations</i>	<i>xi</i>
<i>List of Figures</i>	<i>xii</i>
<i>List of Tables</i>	<i>xiii</i>
1 INTRODUCTION	1
1.1 Introduction	1
1.2 Need of Work	2
1.3 Objectives.....	4
2 LITERATURE SURVEY	5
2.1 Existing System.....	5
2.2 Problem Definition.....	5
2.3 Preliminary Investigation	6
2.4 Proposed System.....	6
2.5 Advantageous of Proposed System	10
3 SYSTEM ANALYSIS AND DESIGN	12
3.1 Requirement Specification	12
3.1.1 Functional Requirements	12
3.1.2 Non Functional Requirements	14
3.2 System Architecture	16
3.2.1 System Architecure diagrams	16
3.2.1 Sequence diagram.....	17
3.2.2 Flowchart	18

3.3	Design and Test Steps/ Criteria	22
3.4	Testing Process	24
3.4.1	Testing Phases	21
3.4.2	Testing of socket programming.....	25
3.4.3	Testing of Gradle	26
4	METHODOLOGY/TECHNIQUES USED	28
4.1	Development Tools.....	28
4.2	Code design	30
4.2.1	Socket programming mechanism	32
4.2.2	Event handling server-side	33
4.2.3	Brightness control mechanism	35
4.3	User interface design	36
5	SYSTEM SECURITY MEASURES	40
5.1	Security threats	41
5.2	Security controls	41
6	EXPERIMENTAL RESULTS / OUTPUTS	42
7	CONCLUSION AND FUTURE SCOPE	47
	References	49
	Sponsorship Letter	51
	Paper Publication certificates	52
	Published Paper	55

Abbreviations:

1. **PC**: Personal Computer
2. **Android**: Operating system developed by Google for mobile devices
3. **USB**: Universal Serial Bus
4. **Wi-Fi**: Wireless Fidelity
5. **Bluetooth**: Wireless technology for exchanging data over short distances
6. **App**: Application
7. **UI**: User Interface
8. **API**: Application Programming Interface
9. **SDK**: Software Development Kit
10. **OS**: Operating System
11. **GUI**: Graphical User Interface
12. **LAN**: Local Area Network
13. **TCP/IP**: Transmission Control Protocol/Internet Protocol
14. **SSID**: Service Set Identifier
15. **IP**: Internet Protocol
16. **HTTP**: Hypertext Transfer Protocol
17. **HTTPS**: Hypertext Transfer Protocol Secure
18. **SSL**: Secure Sockets Layer
19. **AES**: Advanced Encryption Standard
20. **OTP**: One-Time Password

List of Figures

3.1 System Architecture.....	15
3.2 Sequence Diagram	17
3.3 Flowchart	18
3.4 Client Application flow diagram	19
3.5 Server Application flow diagram.....	20
3.6 Socket Programming Testing	26
3.7 Successfully build gradle.....	27
3.8 Lint configuration	31
4.1 Socket Programming module	38
4.2 Event Handling at server side	39
6.1 Home Page.....	42
6.2 Mouse-Pad	43
6.3 Keyboard Page.....	44
6.4 Multikey Page	45
6.5 Server-side output.....	47

List of Tables

3.1 Software Requirements	14
---------------------------------	----

Chapter 1

INTRODUCTION

1.1 Introduction

There are several situations where we want to wirelessly and comfortably operate a computer, where the computer screen is displayed onto a big screen through a projector or big screen television, such as classrooms, conference/meeting rooms, mobile, workgroup project environments and modern office environments, and even living rooms. Several specifically designed devices are available on the market for the goal of operating computers remotely and wirelessly. Wireless keyboard, uses each of two Bluetooth or wireless USB mini-receiver plugged into the USB port of computer for the communication among the keyboard and the computer. Some wireless keyboards have a touchpad for controlling the mouse pointer. Wireless presentation controller, as shown in Fig. 1b, allows user to operate his/her computer remotely for PowerPoint presentation over Bluetooth connection. It usually has several buttons, including mouse-left, mouse-right, next slide and previous slide buttons (for PowerPoint presentation), and even a small size rectangle touchpad for moving mouse cursor, etc.

In This research is mainly about the mobile phone how to control the remote computer. The system contains the following functions.

- 1) To simulate the mouse control. It includes moving the mouse, the mouse click on left and right key, simulation mouse pulley operation, dragging and dropping files and custom function of up and down on the volume key.
- 2) Keyboard input control. It includes sending any text in Chinese or English to a computer, backspace button simulation, receiving the information from cell phone under DOS and control of the key direction.
- 3) The game controller simulation. In the landscape phone mode, the left key is to control the direction and the right shooting button for operation is equivalent to the mouse button click.

1.2 Need of Work

1. **Convenience:** Users often find it more convenient to control their PCs using familiar mobile devices like smartphones or tablets, especially in scenarios where they need to interact with a large screen, such as presentations or media playback.
2. **Flexibility:** Utilizing an Android device as a controller provides flexibility in terms of mobility and ease of use, allowing users to operate their PCs from a distance or in different environments.
3. **Cost-Effectiveness:** Developing such a system can be cost-effective compared to purchasing specialized hardware for PC control. Most users already own Android devices, reducing the need for additional investments.
4. **Integration:** Integrating Android devices as controllers allows for seamless interaction between different devices within the same ecosystem, enhancing the overall user experience.
5. **Innovation:** By exploring the capabilities of Android devices and leveraging technologies like Wi-Fi and Bluetooth, developers can innovate and create new solutions for PC control, catering to evolving user needs and preferences.
6. **Accessibility:** For individuals with mobility impairments or disabilities, using an Android device as a PC controller can greatly improve accessibility, allowing them to operate their computers more comfortably and independently.
7. **Scalability:** As technology advances, the integration of Android devices as PC controllers offers a scalable solution that can adapt to evolving hardware and software standards, ensuring continued compatibility and functionality.
8. **Remote Management:** With the increasing trend towards remote work and virtual collaboration, a PC controller using an Android device enables users to remotely access and control their PCs, facilitating productivity and collaboration from anywhere.
9. **Customization:** Android devices offer extensive customization options through apps and settings, allowing users to tailor their control experience according to their preferences, such as customizing button layouts or gestures for specific tasks.
10. **Compatibility:** Android's widespread adoption across various devices and manufacturers ensures compatibility with a wide range of hardware configurations, making it a versatile platform for PC control solutions.
11. **Real-time Interaction:** Leveraging technologies like Wi-Fi and Bluetooth, a PC controller using an Android device enables real-time interaction with the PC, minimizing latency and ensuring a responsive user experience.
12. **Education and Learning:** In educational settings, such as classrooms or training environments, utilizing Android devices as PC controllers can facilitate interactive learning experiences, engaging students in hands-on activities and demonstrations.

13. **Remote Troubleshooting:** IT professionals and support staff can remotely troubleshoot and resolve issues on users' PCs using Android devices as controllers, reducing downtime and improving productivity in enterprise environments.
14. **Energy Efficiency:** Compared to traditional input devices like keyboards and mice, using an Android device as a PC controller consumes less power, contributing to energy efficiency and reducing environmental impact.
15. **Integration with Smart Home Systems:** For users with smart home setups, integrating Android devices as PC controllers allows for seamless integration with other connected devices and automation systems, enhancing overall home automation capabilities.
16. **Multi-Device Support:** Android apps can be designed to simultaneously control multiple PCs, which is especially useful in environments like digital labs, offices, or for individuals who manage several machines at home or remotely.
17. **User Experience Customization:** Developers can create highly intuitive and user-friendly interfaces that can be personalized for different user roles or needs, such as simplified controls for beginners or more complex functionalities for power users.
18. **Support for Special Features:** Integration of features like gyroscope, accelerometer, and touch input can provide innovative ways to interact with PCs, such as gesture-based controls or motion inputs which can simulate a more natural interaction paradigm compared to traditional mouse and keyboard setups.

Overall, the need for developing a PC controller using an Android device stems from the desire to enhance convenience, flexibility, and user experience in controlling PCs, while also providing a cost-effective and innovative solution

1.3 Objectives

The main objective of the project aims to deliver a comprehensive and versatile PC controller Android app that meets the diverse needs of its users across various usage scenarios and environments.:.

- **Touch-screen Mouse Control:** Develop a feature allowing users to control the computer's cursor using touch-screen gestures, including two onscreen mouse buttons for left and right-click actions.
- **Portrait/Landscape Mode Compatibility:** Ensure the app's interface is optimized for both portrait and landscape orientations to accommodate user preferences.
- **Wi-Fi Network Compatibility:** Ensure the app can connect and function over any Wi-Fi network, providing flexibility in usage environments.
- **Keyboard Typing Capability:** Implement a virtual keyboard feature that allows users to type on the computer using the Android device's touch-screen interface.
- **Computer Functionality Control:** Enable users to perform various computer operations, such as opening, handling, and modifying files or applications, directly from the Android device.
- **Android OS Version Compatibility:** Ensure the app is compatible with all versions of the Android operating system to maximize its accessibility to users.
- **Cross-Platform Compatibility:** Ensure the app can connect and function with computers running Mac, Windows, or Linux operating systems, providing versatility and broad compatibility.
- **Voice Typing Mechanism:** Implement a voice recognition feature that allows users to input text on the computer using voice commands issued from the Android device.
- **Live Coding Demo:** Enable the app to be used for live coding demonstrations, allowing users to showcase coding activities and projects directly from their Android device to the computer screen.
Power Management Controls: Provide controls for power management, such as shutdown, restart, sleep, and wake-on-LAN features, to enhance remote management capabilities.
- **Real-Time Notifications:** Send real-time alerts and notifications to the Android device about important PC events, such as system warnings, hardware malfunctions, or software updates.
- **Screen Mirroring:** Add a feature that enables screen mirroring, allowing users to view and interact with their PC screen directly from their Android device, enhancing the capabilities for presentations and detailed control.
- **Bandwidth Optimization:** Optimize the app to use minimal bandwidth to maintain performance even on slower network connections, ensuring reliable and efficient control.

Chapter 2

LITERATURE SURVEY

2.1 Existing System

1. Agora:

The agora of wirelessly operating a computer, especially when its screen is projected onto a large display like a projector or television, encompasses a multitude of contexts, from classrooms to conference rooms, mobile workgroups, and even living rooms. In response to this demand, specialized devices have emerged. Wireless keyboards, utilizing Bluetooth or wireless USB mini-receivers, facilitate seamless communication between the keyboard and the computer, offering comfortable control without the constraints of cables.

2. Wireless Pc controller:

It is an Android App that accomplish live connection between computer and android device by using WIFI for transmission using Socket programing. There is are two sides like client and server, An Agent Creates server and wait for response from client and server.

2.2 Problem Definition

To create android app that establish connection between laptops, computers and android device using WIFI and allows use of android smartphones touchpad as mouse and phone as wireless keyboard to control laptops and computers Wirelessly.

2.3 Preliminary Investigation

The main aim of preliminary investigation is to identify the problem. First, need for the new or the enhanced system is established. Only after the recognition of need, then the proposed system is compared and then further analysis is possible. At this stage, we had to perceive the problem and opportunities, the existing system is studied and found out that there were few areas where we can integrate with other technology to make the system better than the existing system. It was analyzed that such proposed system would be possible to develop with given and it might turn out to be the feasible solution.

For the PC controller project using an Android device, integrating the Android app with existing PC control solutions presented significant challenges. The primary technical hurdle was establishing a reliable and secure communication protocol between the Android devices and different operating systems like Windows, macOS, and Linux. Throughout the development process, we encountered several unit-level issues, including the development of a responsive user interface that adapts to both portrait and landscape modes without lag, and ensuring compatibility across various versions of Android and PC operating systems.

2.4 Proposed System

Several studies have been done on using computer technologies to improve elections. These studies tell about the risks of adopting electronic voting system, because of the software challenges, insider threats, network vulnerabilities, and the challenges of auditing. We are going to provide solution to modern voting system such that it should overcome all the problems occurring in the current voting system. Introducing a blockchain based E- voting system would bring a fraud less, non- alterable, and transparent e-voting system. During critical times of Covid-19 people can vote online from their homes. Hence there is no standing in queues for casting votes. It will save a lot of time as well as social distancing is maintained. As data added in blockchain is more secure hence application of blockchain in voting system would be a boon to modern election systems.

Modular Design

Every project that is to be implemented is not implemented directly. The project is divided into multiple sub modules for convenience in development of the project.

The functionality of the proposed application is divided into number of sub modules. Our project basically includes two modules they can be stated as

A. Client-side Module

This module is an android containing user interface and have various keyboard buttons and mouse controls to control PC wirelessly

B. Server-side Module

This module contains a java program which hold all necessary libraries for connectivity and contains java socket programming which initiates the socket for connection and waits for input from client side and handles all the input form clients and perform necessary action on computer

2.4.1 System Design

System design is a process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. In our case we basically focus on two major particulars i.e., Design Goals and Modularization.

A. Design Goals

The design goals for the PC controller Android app aim to create a user-friendly and versatile interface that seamlessly integrates with existing PC systems. Key objectives include developing a responsive touch-screen interface for intuitive navigation and control, ensuring compatibility with various PC operating systems, and optimizing performance for both portrait and landscape orientations. Security is paramount, with encryption protocols implemented to safeguard data transmission between the Android device and PC. Additionally, customization options are prioritized, allowing users to tailor the app to their preferences through customizable layouts, gestures, and control panel configurations. Overall, the design goals focus on delivering a comprehensive solution that enhances convenience, flexibility, and productivity in PC control scenarios.

B. Modularization Details:

The project has been divided into two modules in which for each functionality we have designated modules. Any software comprises of many systems which contains several sub-systems and those sub-systems further contains their sub-systems. So, designing a complete system in one go comprising of each and every required functionality is a hectic work and the process can have many errors because of its vast size. Effective modular design can be achieved if the partitioned modules are separately solvable, modifiable as well as compliable. Following are the project modules:

A. Client-side Module (Android):

- User Interface (UI): This component manages the presentation layer of the Android app, including layouts, buttons, and controls for keyboard input and mouse emulation. It provides an intuitive interface for users to interact with and control the PC wirelessly.
- Input Handling: Responsible for interpreting user inputs such as touch-screen gestures, keyboard presses, and mouse movements. It converts these inputs into commands to be sent to the server-side module for execution on the PC.
- Network Communication: Manages communication with the server-side module over Wi-Fi or Bluetooth. It establishes connections, sends control commands, and receives feedback from the server to update the UI accordingly.
- Security Features: Implements security measures such as encryption and authentication to ensure secure communication between the Android device and the server-side module, protecting against unauthorized access or tampering.

B. Server-side Module (Java Program):

- Socket Initialization: This component initializes the server-side socket for communication with the client-side Android app. It listens for incoming connections from Android devices and establishes communication channels to handle incoming commands.
- Input Handling: Receives input commands from the client-side Android app and processes them to perform actions on the PC. It interprets commands related to keyboard inputs, mouse movements, and other control functions and executes corresponding actions.
- PC Control Logic: Contains the core logic for controlling various aspects of the PC based on the received commands. It interacts with the PC's operating system to perform actions such as simulating keyboard inputs, moving the mouse cursor, or executing system commands.
- Dependency Management: Manages dependencies and libraries required for connectivity and PC control functionalities. It ensures that the server-side module has access to necessary resources and libraries for smooth operation.
- Error Handling and Logging: Implements error handling mechanisms to gracefully handle exceptions and errors that may occur during communication or PC control. It logs relevant information for debugging and troubleshooting purposes.

2.4.2 Modules Implemented.

The PC controller using android device basically includes multiple steps in its development. As previously stated, that a project is not implemented as a whole it is divided into multiple modules for convenience in the development of the project. In our project we have considered five modules that are implemented.

A. User Interface Module (Android App):

- This module focuses on creating an intuitive and user-friendly interface for controlling the PC wirelessly from an Android device.
- It includes features such as virtual keyboard buttons, mouse controls, and touch-screen gestures for navigation and interaction.
- The UI should be optimized for both portrait and landscape orientations to ensure seamless usability in various device configurations.

B. Communication Module (Android App):

- Responsible for establishing and managing communication channels between the Android device and the PC.
- Implements protocols such as Wi-Fi Direct, Bluetooth, or network sockets for reliable data transmission.
- Handles the exchange of control commands and feedback messages between the client-side Android app and the server-side PC controller.

C. PC Control Logic Module (PC Server Program):

- This module contains the core logic for interpreting control commands received from the Android device and executing corresponding actions on the PC.
- It includes functionalities for simulating keyboard inputs, mouse movements, system commands (e.g., shutdown, restart), and media playback control.
- Utilizes Java or another suitable programming language to interact with the PC's operating system and execute control actions.

D. Security Module (Both Client and Server):

- Ensures secure communication between the Android device and the PC to prevent unauthorized access or tampering.
- Implements encryption and authentication mechanisms to protect sensitive data transmitted over the network.
- Includes features such as user authentication, session management, and data encryption to safeguard the integrity and confidentiality of communication
- .

E. Configuration and Customization Module (Android App):

- Allows users to customize their control experience by adjusting settings such as control sensitivity, button layouts, and gesture recognition preferences.
- Provides options for configuring network settings, connection parameters, and PC control preferences within the Android app.
- Enables users to save and manage multiple configurations for different PC setups or usage scenarios.

F. Testing and Debugging Module (Both Client and Server):

- Includes unit tests, integration tests, and UI tests to verify the functionality, performance, and reliability of each module.
- Facilitates debugging and troubleshooting of issues encountered during development and testing phases.
- Provides logging and error handling mechanisms to capture and analyse runtime errors, exceptions, and unexpected behaviours.

2.5 Advantageous of Proposed System

Smartphones are part of almost everything we do – business, entertainment, knowledge, social networking, etc. In most cases, they must have access to the Internet and our data. But lately, the Smartphone's are being connected even to a computer. We cannot store all our data on our smartphones. Sometimes we need to access the information like important data (word file, image, and video) that we need an urgent basis. Where we sometimes out of the station and in that case by using this software we can take that data by simply login to an android client where the list of machines will be displayed and we can take important data from those machines and copy-paste in our android phone. We can do many things like rename, delete, rename desktop file or folder and even we can change the desktop file data.

The advent of PC controllers using Android devices presents a paradigm shift in the way users interact with their computers, offering several distinct advantages over traditional methods. Firstly, the convenience factor is unparalleled, as users can now seamlessly control their PCs wirelessly from anywhere within the vicinity of the Wi-Fi network, eliminating the need for cumbersome wired connections and physical proximity to the computer. This newfound mobility is especially beneficial in scenarios such as presentations, where users can navigate their PC's interface effortlessly while moving around the room, enhancing engagement and interaction. Additionally, the versatility of Android devices allows for a highly customizable control experience, with options to tailor the interface, input methods, and control gestures to suit individual preferences. Furthermore, the integration of advanced features like touch-screen gestures, voice commands, and multimedia controls enhances user productivity and efficiency, enabling seamless multitasking and fluid interaction with various applications and media content. Moreover, the seamless integration with existing smart home ecosystems further enhances the utility of PC controllers using Android devices, allowing users to consolidate control of their PC alongside other connected devices and automation systems, thereby streamlining their digital workflows and enhancing overall user experience. Overall, the transition to PC controllers using Android devices represents a significant leap forward in terms of convenience, flexibility, and functionality, offering users a more intuitive and immersive computing experience compared to traditional input methods.

The main advantages of PC controller using android device:

1. Eliminate need of hardware: The use of wireless PC control android app eliminate need of carry wireless hardware such as mouse keyboard.
2. Accessibility: You can easily connect app with less efforts and faster setup. No need to pair and connect device to PC and laptop every time.
3. Real Time performance: Our app is using WIFI based communication that makes low latency and almost 0 latency to perform actions which provides smoother experience to user.
4. Integration with Existing System: The project can be integrated with existing system no need to setup new hardware or systems.
5. User Friendly Interface: The development of an intuitive user interface ensures that the system is accessible and easy to use for individuals.
6. Enhanced Mobility: With the ability to control the PC wirelessly from anywhere within the Wi-Fi network range, users gain greater mobility and flexibility in their computing activities. This is particularly useful in scenarios where users need to move around while controlling the PC, such as during presentations or while giving demonstrations.
7. Multi-Device Compatibility: The PC controller Android app can be installed and used on a wide range of Android devices, including smartphones and tablets, providing flexibility and compatibility across different devices. This allows users to control their PC using their preferred device without being limited to specific hardware.
8. Remote Access: Users can remotely access and control their PC from anywhere with an internet connection, offering convenience and flexibility, especially for remote workers or individuals on the go. This feature enables users to troubleshoot issues, access files, or perform tasks on their PC remotely, enhancing productivity and efficiency.
9. Customizable Controls: The PC controller Android app offers customizable controls and settings, allowing users to tailor their control experience according to their preferences. Users can customize button layouts, assign shortcuts, and adjust sensitivity settings to optimize their control experience for different tasks and applications.

Chapter 3

SYSTEM ANALYSIS AND DESIGN

System analysis is the process of gathering and interpreting facts, diagnosing problems and using the information to recommend improvements on the system. System analysis is a problem solving activity that requires intensive communication between the system users and system developers.

System analysis or study is an important phase of any system development process. The system is viewed as a whole, the inputs are identified and the system is subjected to close study to identify the problem areas. The solutions are given as a proposal. The proposal is reviewed on user request and suitable changes are made. This loop ends as soon as the user is satisfied with the proposal.

3.1 Requirement Specification

The Software requirement specification is a specification for a specific software product, program, or set of applications that perform particular functions in a specific environment. It serves several goals depending on who is writing it. First, the SRS could be written by the client of a system. Second, the SRS could be written by a developer of the system. It is complete specification and description of requirements of software that needs to be fulfilled for successful development of software system. These requirements can be functional as well as non-requirements depending upon type of requirement. The interaction between different customers and contractor is done because it is necessary to fully understand needs of customers.

3.1.1 Functional Requirements

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirement

A. Software requirements

The Software requirements required for our project are stated as below

- 1) **Android Studio IDE:** Android Studio is the official integrated development environment (IDE) for Android app development. It provides a comprehensive set of tools for designing, building, and testing Android applications, including features for code editing, debugging, and performance analysis.
- 2) **Virtual Device (Emulator):** Android Studio includes a built-in emulator that allows developers to test their apps on virtual Android devices. The emulator simulates various device configurations and Android versions, providing a convenient way to test the PC controller app across different screen sizes and resolutions.
- 3) **Visual studio code:** Visual Studio Code is a freeware source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.
- 4) **Android device:** Android device with Android version 6 and above for testing.
- 5) **Windows 10 OS:** Windows 10 is a series of operating systems developed by Microsoft and released as part of its Windows NT family of operating systems
- 6) **JDK (Java Development Kit) / JVM (Java Virtual Machine):** Android app development primarily involves programming in Java or Kotlin languages. Therefore, installing JDK, which includes the JVM, is necessary for compiling Java code and running Android applications on devices or emulators.
- 7) **Git (Version Control):** Git is a widely used version control system for tracking changes in software development projects. Integrating Git with Android Studio or Visual Studio Code allows for efficient collaboration, code sharing, and version management among team members working on the PC controller app.
- 8) **Wireless Debugging Tools:** To facilitate debugging and testing of the PC controller app on physical Android devices, developers may need to set up wireless debugging tools such as ADB (Android Debug Bridge) over Wi-Fi. This allows for remote debugging without the need for a USB connection between the device and the development machine.

Software	Type	Version
Android Studio IDE	Integrated Development Environment (IDE)	4.2.1
Virtual Device (Emulator)	Emulation Software	Android 6 and above
Visual studio code	Source-Code Editor	Visual Studio Code 1.56.2
Android device	Smartphone or Tablet running Android OS	Android 6 and above
Windows 10 OS	Operating System	Windows 10 version 21H1
JDK (Java Development Kit)	software Development Kit	JDK 11, JDK 8
Git (Version Control)	Version Control System	Git 2.31.1
Wireless Debugging Tools	Debugging Utilities	ADB (Android Debug Bridge) over Wi-Fi, compatible

Table 3.1 Software requirements

3.1.2 Non-Functional Requirements

These are basically the quality constraints that the system must satisfy according to the project idea. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements or non-functional requirements.

A. Performance Expectations:

- 1) **Response Time:** The system should exhibit low latency, with users able to establish a connection and initiate control actions within a few seconds. This includes importing the necessary configurations and establishing communication between the Android device and the PC.
- 2) **Responsiveness:** The app should respond promptly to user inputs, with minimal delay between actions performed on the Android device and corresponding actions executed on the PC.

B. Performance Modes:

- 1) **Normal Operation:** During regular usage, such as navigating the PC's interface or controlling media playback, the system should respond swiftly, with actions executed in real-time or near real-time.
- 2) **High Load Situations:** In scenarios where multiple users simultaneously access the PC controller app or perform resource-intensive tasks, the system should maintain acceptable performance levels without significant degradation in responsiveness.

C. Security Requirements:

The security requirements for the PC controller using an Android device encompass several key aspects to ensure the integrity, confidentiality, and privacy of user data and interactions. This includes implementing robust encryption protocols for secure communication between the Android device and the PC, establishing strong authentication mechanisms to verify user identities, enforcing access controls to restrict unauthorized actions, and employing secure storage mechanisms to safeguard sensitive information. Additionally, measures such as protection against malicious attacks, adherence to secure coding practices, regular security updates, and compliance with relevant regulations are essential to mitigate potential risks and maintain the trust and confidence of users. By addressing these security requirements comprehensively, the PC controller app can provide a secure and reliable platform for users to control their PCs remotely with confidence in the protection of their data and privacy.

D. Reliability:

Reliability in the PC controller using an Android device entails ensuring consistent and uninterrupted functionality of the system. This involves maintaining the system's integrity through regular maintenance and bug fixes, ensuring accurate transaction handling, and ensuring that voting statuses are promptly updated and reflected accurately. Additionally, the system should be resilient to network interruptions and web3 connection errors, ensuring seamless operation during voting sessions. A user-friendly interface and guidance for first-time users contribute to a reliable user experience, ensuring that the system functions flawlessly and reliably throughout its operation

E. Usability:

- The system will have a minimal and simple User Interface.
- To guide the users for the first time using it, there will be a guidance related to the usage of the system.

3.2 System Architecture

System architecture is a conceptual model that defines the structure and behavior of a system's components and subsystems. It defines the system's boundaries, components, data flow, and communication channels. System architecture can be categorized into centralized and decentralized architectural organizations.

3.2.1 System Architecture diagram

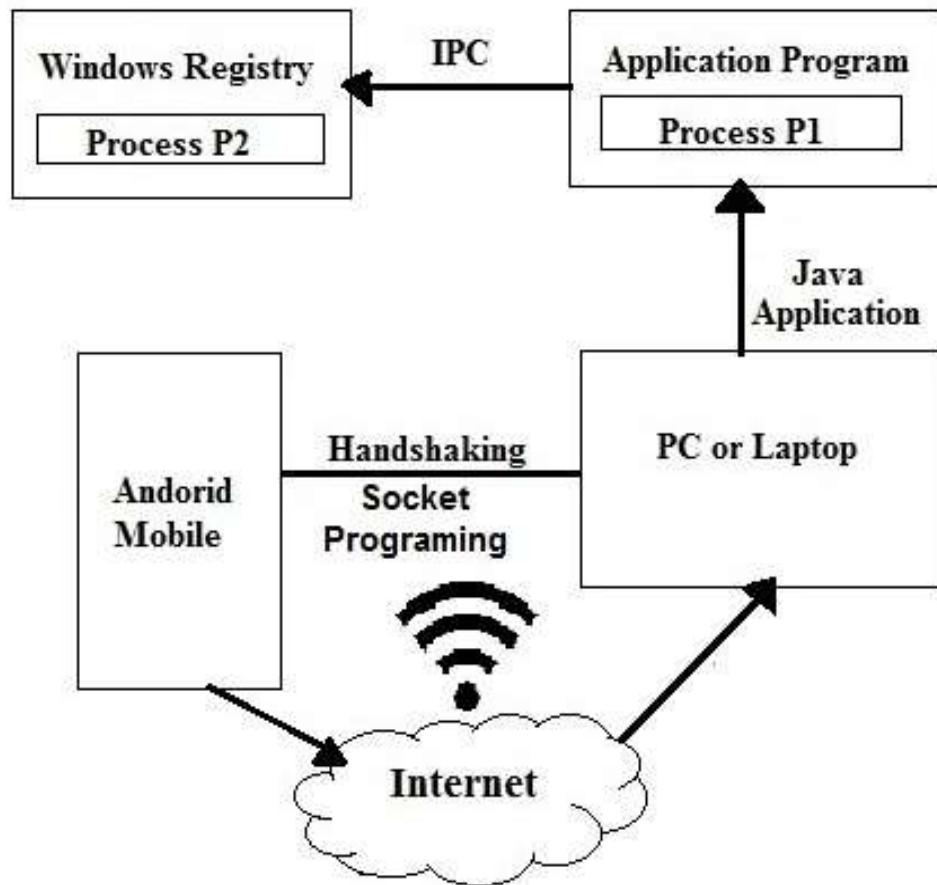


Fig 3.1. System Architecture

1. Start Point:

The application starts with the initiation of the Windows Registry to store configuration settings and options.

2. IPC Application Program:

The application interfacing with an IPC (Inter-Process Communication) Application Program, which enables communication between various processes or applications on the system.

3. Java Application:

The use of Java programming language in the application, indicating Java-based development and functionality.

4. Handshaking with PC or Laptop:

Establishment of a handshake process with a PC or laptop, possibly for synchronization, data exchange, or control interactions.

5. Android Socket:

Integration of Android Socket functionality, suggesting communication capabilities between the application and an Android device via sockets.

6. Mobile Programming:

Inclusion of mobile programming aspects, indicating the application is designed to run on mobile devices and tailored for mobile platforms.

7. Endpoint:

The endpoint signifies the conclusion or final processing stage of the application flow, possibly indicating the completion of a transaction or data exchange

3.2.2 SEQUENCE DIAGRAM

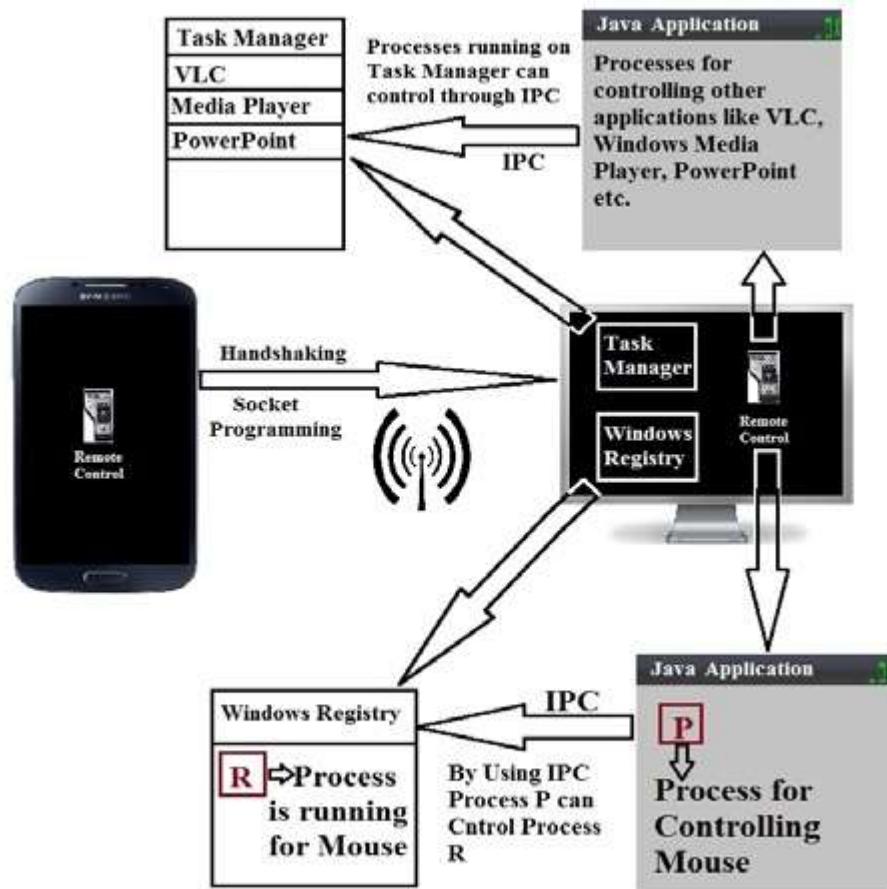


Fig.3.2. SEQUENCE DIAGRAM

3.2.3 FLOWCHART

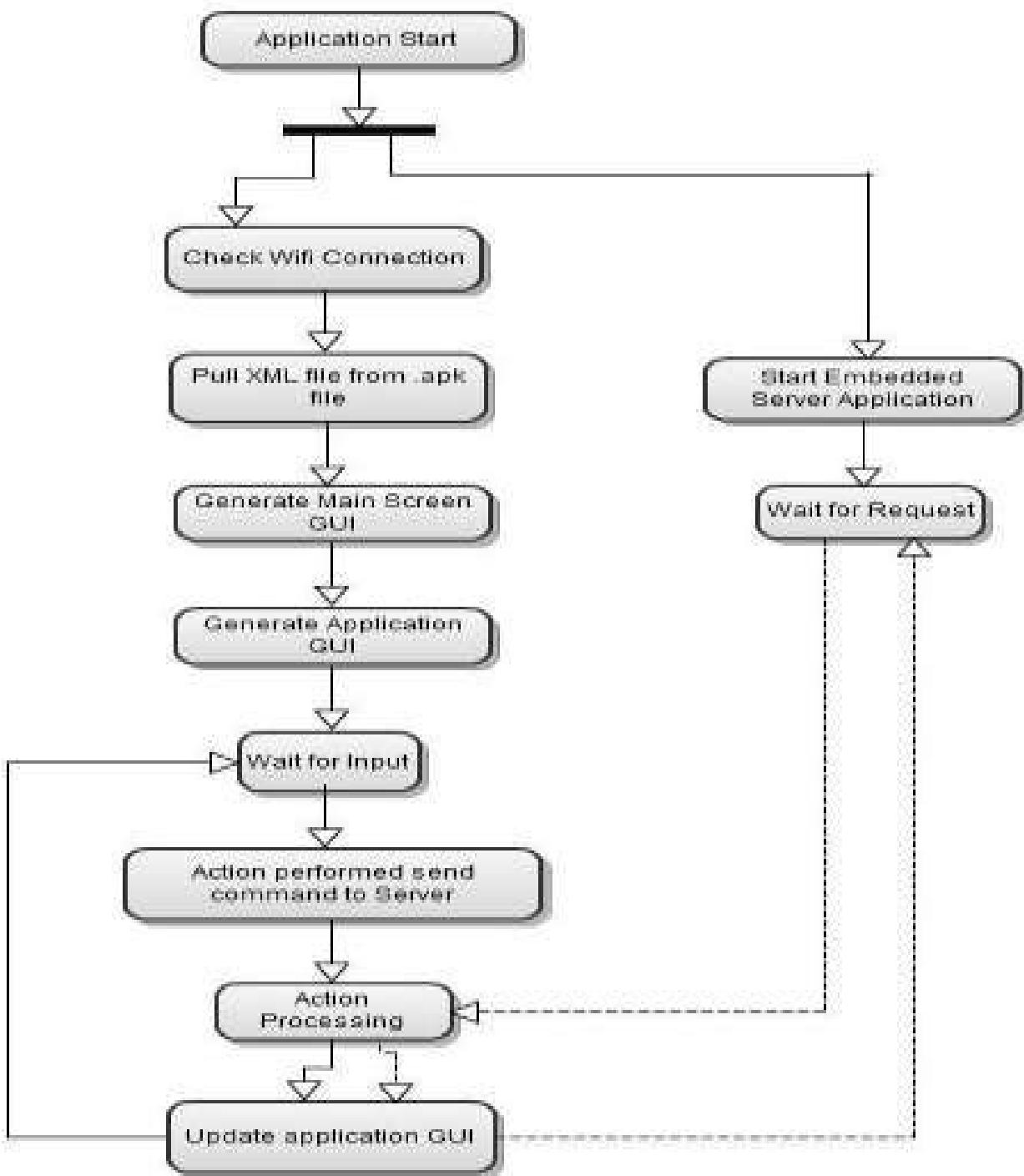


FIG.3.3. FLOWCHART

This fig 3.3 shows the overall program flow for this application. When the remote control is run, and it follows the path shown on the figure below. After application starts, the embedded java application server is runs in parallel. Sound notification is implemented in the proposed application so as to let users being aware that their IP address. so it has been validated and the user can proceed with the application.

1. When we start android application which is on server side it first check for Wi-Fi connection whether Wi-Fi is turned on or not and after that if condition get satisfied then it pull xml file and generated main user interface on application.
2. After that it waits for user input which is custom i/p address, when user enters the custom i/p address it send command to server and checks for connection with that i/p address.
3. After successful connection it updated GUI of android application
4. For server side it is straight forward process when we start java program or server application it makes system ready for connection i.e. it reserve and socket for connection and wait for request from client side.

3.2.4. Client Application flow diagram

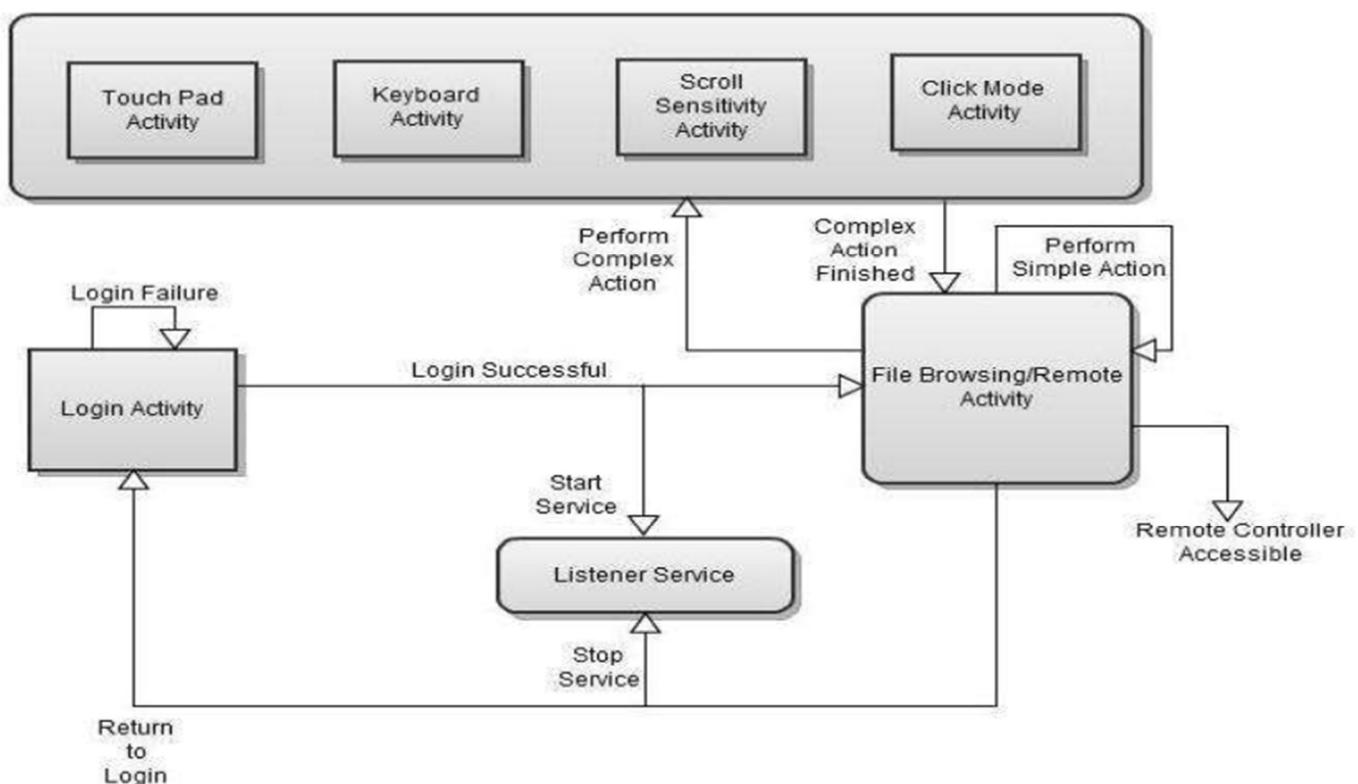


Fig 3.4 Client Application flow diagram

There are specially two ways to deliver an application on Android: as a client-side application developed. By using the Android SDK and installed on user devices as an .apk or as a web application developed using web standards and accessed through a web browser. Here in this project the client application was implemented. By based on the platform using Android Software Development Kit, this application is written using the Java Programming Language and is designed to run on a customized Linux-based virtual machine called Dalvik. The developed client application is an .apk extension the interested users can download from the developer and installed directly to their mobile phone device. Fig. 2 shows the program flow diagram of the client application. First, the application program starts and check the Wi-Fi connection. The embedded web server is runs in parallel once the application starts. The application pick the xml file from the .apk file and generate the main screen and moves ahead to generate also the application Graphical Us er Interface. The application GUI awaits an input from the user and when the action has been performed, it sends a command to the Web Server which can processes the action sent from the GUI. Then the Web Server resends back the action and that has been processed to the Application GUI which then updates the input back to the user as expected.

3.2.5: Server application flow diagram

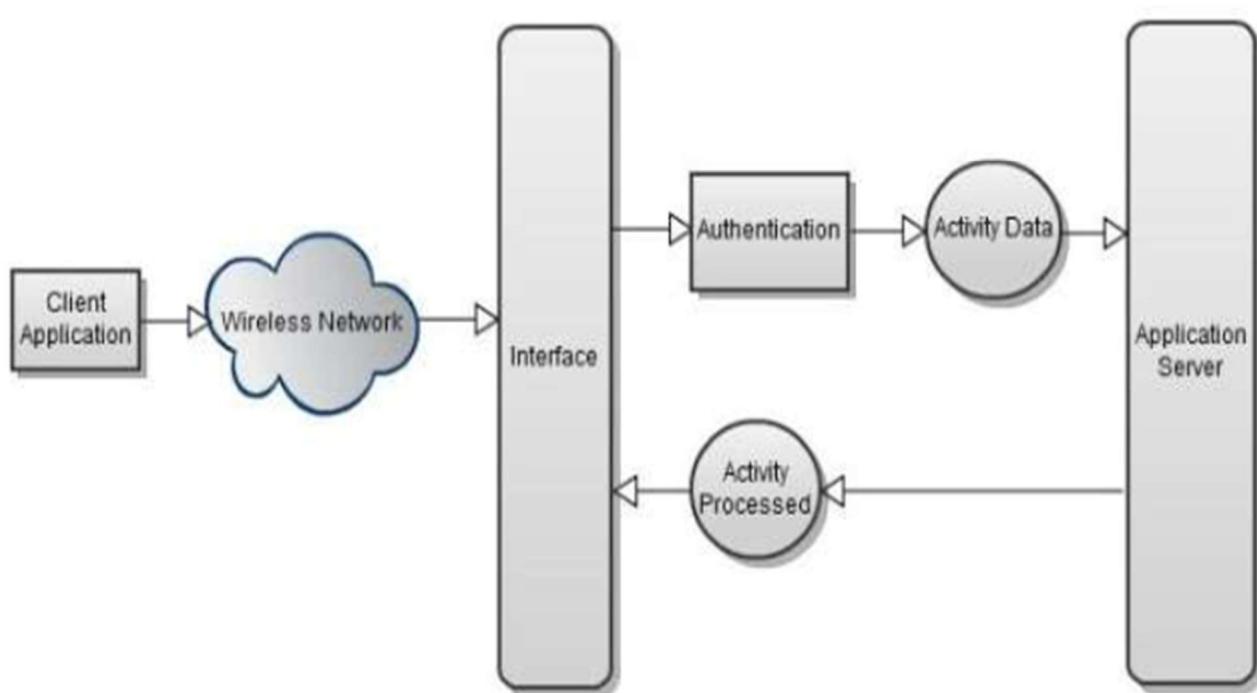


Fig 3.5 Server flow diagram

Now a day mobile phones are having more computing power, running a server on mobile phones can do a lot of more help in running the device. In the project the server works as a pair to the client. For without the server and that transmits the message to the listener, the application itself would not be able to function. To create a typical web server, and one has to create a server socket and listen to the desired port. Then accept the connection and finally process the request and send the response to the client. For the server side of this application, a java server application which is required for listening to client activities. In this project the server is a private java application server is hosted by the application's machine. The application's machine can also be any machine where Java Runtime Environment (JRE) is installed. Fig. 3 shows the program flow diagram of the server application

In addition to hosting the server application on the device, leveraging the computing power of mobile phones offers several advantages. Firstly, it enables efficient utilization of resources, as the device can handle server tasks without the need for additional hardware. This reduces dependency on external servers and enhances the overall reliability and availability of the system. Moreover, running a server on the mobile phone enables seamless integration with other functionalities and sensors of the device, allowing for enhanced features such as real-time data processing, location-based services, and sensor data monitoring. Additionally, hosting the server locally on the device enhances data privacy and security by minimizing data transmission over external networks, reducing the risk of interception or unauthorized access. This approach also promotes scalability, as the system can easily accommodate increasing user demands without relying on external infrastructure. Overall, leveraging mobile phones as servers offers a cost-effective, flexible, and reliable solution for running server applications in various contexts.

3.3 Design and Test Steps / Criteria

A. General Test Cases

1. Verify that the user can successfully connect the Android device to the PC controller app.
2. Confirm that all control functions, including mouse movements, keyboard inputs, and media controls, are functioning correctly across different categories.
3. Check that the connection between the Android device and the PC remains stable during prolonged usage.
4. Verify that the user interface elements, such as buttons and menus, are displayed correctly and are responsive to user interactions.
5. Test the compatibility of the PC controller app with various Android devices, ensuring consistent performance across different hardware configurations.
6. Confirm that the app functions smoothly in both portrait and landscape orientations.
7. Verify that the app does not consume excessive battery or system resources during operation.
8. Test the app's compatibility with different PC operating systems, including Windows, macOS, and Linux.
9. Ensure that the app's performance remains consistent under different network conditions, including Wi-Fi and mobile data.
10. Check for any security vulnerabilities or privacy concerns in the app's communication protocols and data transmission methods.
11. Validate that the app maintains data integrity and does not corrupt or lose user input during transmission to the PC.
12. Test the app's ability to handle interruptions, such as incoming calls or notifications, without disrupting its functionality.
13. Verify that the app's user interface is intuitive and easy to navigate, with clear instructions provided for setup and usage.
14. Test the app's error handling capabilities, ensuring that it gracefully handles unexpected errors or exceptions without crashing or freezing.
15. Confirm that the app complies with relevant industry standards and guidelines for user interface design, accessibility, and performance.

B. Server Side - Test cases

1. Verify that the server application is running and listening on the designated port.
2. Test the server's ability to establish connections with multiple client devices simultaneously.
3. Confirm that the server accurately receives and processes requests from connected client devices.
4. Validate the server's ability to handle various types of requests, including mouse movements, keyboard inputs, and media controls.
5. Test the server's responsiveness to client commands, ensuring timely execution of actions on the PC.
6. Verify that the server correctly relays responses back to the respective client devices.
7. Check the server's error handling capabilities, ensuring that it gracefully handles invalid or unexpected requests without crashing.
8. Test the server's performance under load by simulating a high volume of concurrent client connections and requests.
9. Validate the server's security measures, such as encryption and authentication, to protect against unauthorized access or data breaches.
10. Ensure that the server complies with relevant industry standards and protocols for network communication and data transmission.

3.4 Testing Process

The testing phase of any project is one of the most important phase in project development. In our case we use Mocha framework for testing purpose. Testing basically can be divided into unit testing and integration testing.

3.4.1 Testing Phases:

A. Unit testing:

This is the first and the most important level of testing. Its need begins from the moment a programmer develops a unit of code. Every unit is tested for various scenarios. Detecting and fixing bugs during early stages of the Software helps reduce costly fixes later on. It is much more economical to find and eliminate the bugs during early stages of application building process. Hence, Unit Testing is the most important of all the testing levels. As the software project progresses ahead it becomes more and more costly to find and fix the bugs.

In the unit testing we basically create a test plan. Then we verify our test cases in the testing. Then after primary focus is on creation of scripts to run the test cases wherever applicable. Once the code is ready execution of the test cases is performed. Fixing of the bugs if present and re testing of the code is executed after executing test cases. Finally repetition of the test cycle is performed until the Unit is free from all types of bugs.

B. Integration Testing

Integration strategy stands for how individual modules will be combined during Integration testing. The individual modules can be combined in one go, or they can be joined one by one. A decision on how to put the pieces together is called the Integration Strategy.

We have used bottom-up integration approach to integrate test our application. In Bottom Up Integration, we move from the bottom to top i.e. the components below are first written and these are integrated first. The integration happens from bottom to top.

3.4.2 Testing of socket programming

Socket programming test commands are specific commands or scripts used to test the functionality and performance of socket-based applications. These commands are typically executed from the command line interface (CLI) or through scripts. Here are some commonly used socket programming test commands:

1. **Net cat (NC):**
 - Net cat is a versatile networking utility that can be used to establish TCP or UDP connections, send data, and listen for incoming connections. It is commonly used for testing socket connections and debugging network applications.
 - Example: **nc -vz <host> <port>** - Tests if a connection can be established to the specified host and port.
2. **Telnet:**
 - Telnet is a network protocol used to establish a command-line connection to a remote server. It can be used to manually interact with TCP-based services and test socket connections.
 - Example: **telnet <host> <port>** - Initiates a telnet session to the specified host and port.
3. **Ping:**
 - The ping command is used to test the reachability of a host on an IP network. While not specifically for socket programming, it can be used to test network connectivity and identify potential issues.
 - Example: **ping <host>** - Sends ICMP echo requests to the specified host to test network connectivity.

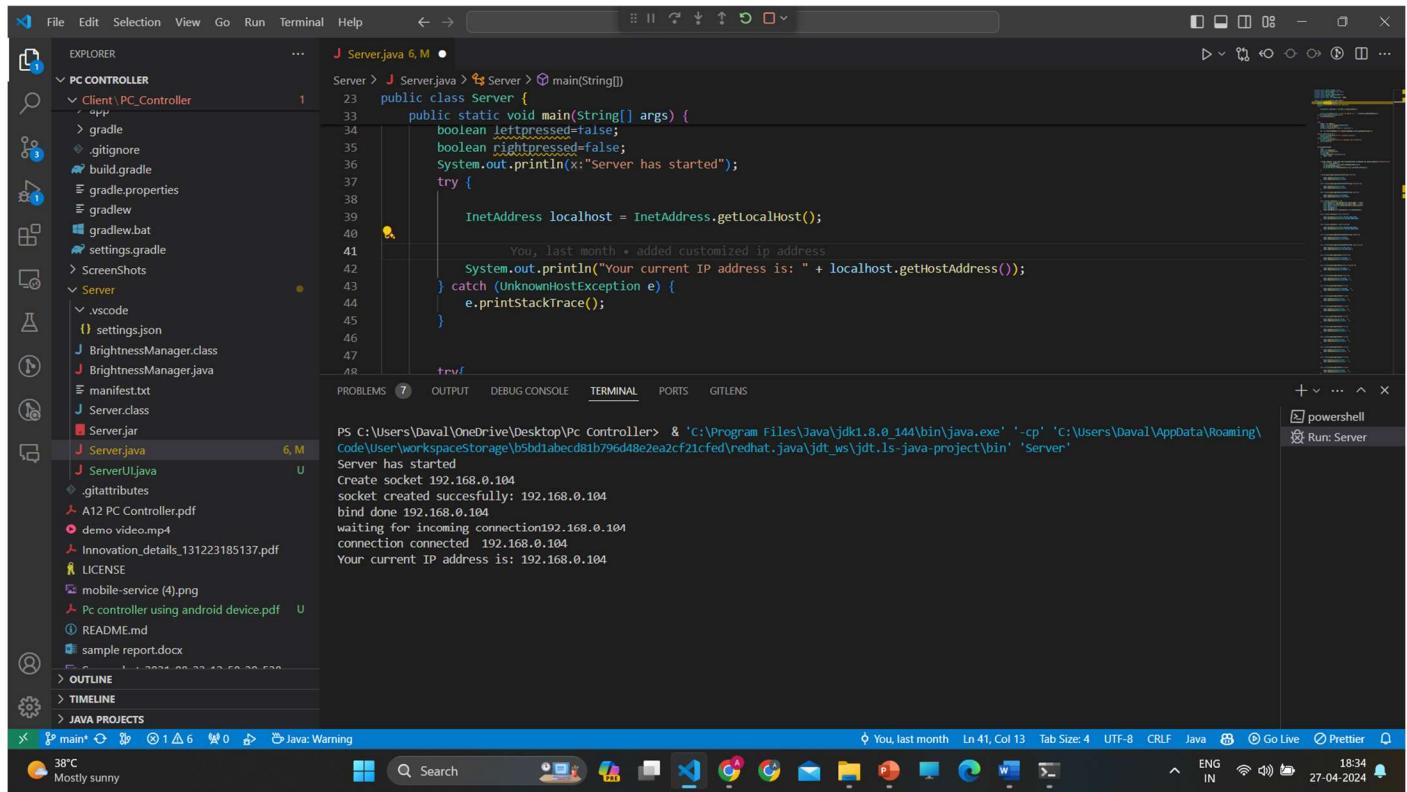


Fig.3.6 Socket Programming testing

3.4.3 Testing of Gradle

Testing the Gradle build process in Android Studio involves ensuring that your project builds successfully, dependencies are resolved correctly, and the application can be deployed and run on a device or emulator. Here's an overview of how to test Gradle builds in Android Studio:

1. Build Project:

Click on the "Build" menu in Android Studio.

Select "Build Project" (or press Ctrl + F9) to initiate the Gradle build process.

Monitor the "Build" tool window for any build errors or warnings.

2. Run Gradle Tasks:

Open the Gradle tool window by clicking on "View" > "Tool Windows" > "Gradle".

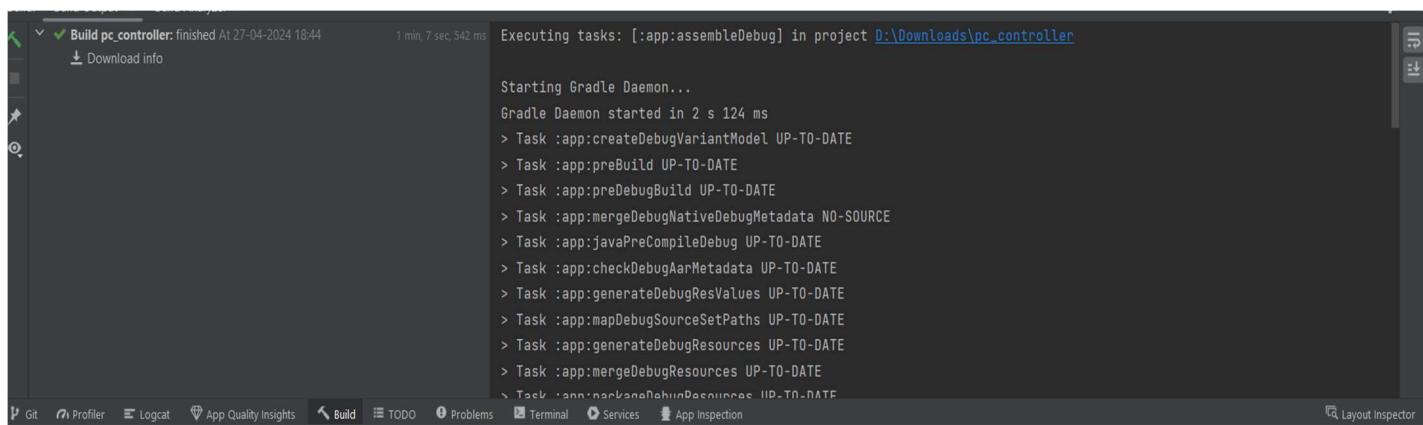
Expand your project and navigate to the "Tasks" node.

Double-click on individual Gradle tasks to run them, such as "assembleDebug" or "assembleRelease".

3. Check Console Output:

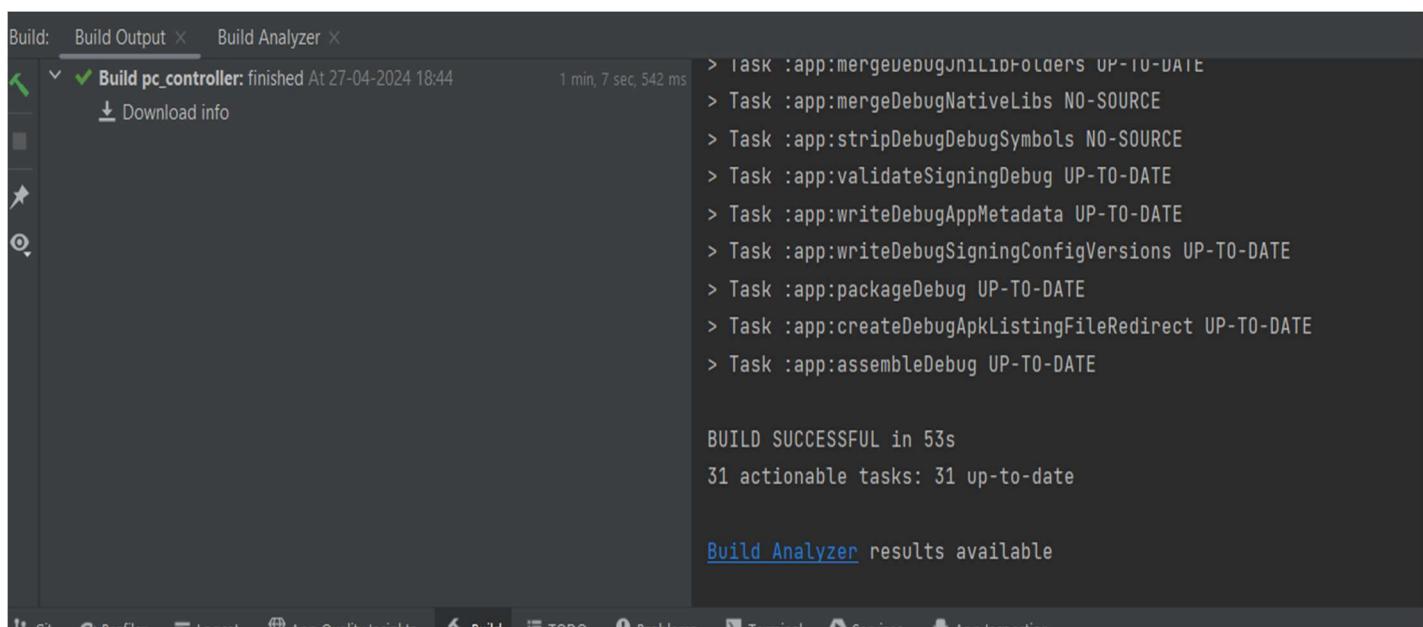
Review the output in the "Run" or "Debug" console windows for any error messages or stack traces.

Pay attention to any warnings or messages related to dependency resolution, resource processing, or compilation.



The screenshot shows the Android Studio interface with the 'Build' tab selected. A green checkmark indicates a successful build named 'Build pc_controller' completed at 27-04-2024 18:44. The build took 1 min, 7 sec, 542 ms. Below the status bar, the toolbar includes Git, Profiler, Logcat, App Quality Insights, Build, TODO, Problems, Terminal, Services, and App Inspection. The main pane displays the command-line output of the build process:

```
Executing tasks: [:app:assembleDebug] in project D:\Downloads\pc_controller
Starting Gradle Daemon...
Gradle Daemon started in 2 s 124 ms
> Task :app:createDebugVariantModel UP-TO-DATE
> Task :app:preBuild UP-TO-DATE
> Task :app:preDebugBuild UP-TO-DATE
> Task :app:mergeDebugNativeDebugMetadata NO-SOURCE
> Task :app:javaPreCompileDebug UP-TO-DATE
> Task :app:checkDebugAarMetadata UP-TO-DATE
> Task :app:generateDebugResValues UP-TO-DATE
> Task :app:mapDebugSourceSetPaths UP-TO-DATE
> Task :app:generateDebugResources UP-TO-DATE
> Task :app:mergeDebugResources UP-TO-DATE
> Task :app:processDebugManifest UP-TO-DATE
```



This screenshot shows the 'Build Output' tab in the Android Studio interface. It displays the same successful build information as the previous screenshot. The build output pane shows the detailed tasks executed:

```
> Task :app:mergeDebugJniLibFolders UP-TO-DATE
> Task :app:mergeDebugNativeLibs NO-SOURCE
> Task :app:stripDebugDebugSymbols NO-SOURCE
> Task :app:validateSigningDebug UP-TO-DATE
> Task :app:writeDebugAppMetadata UP-TO-DATE
> Task :app:writeDebugSigningConfigVersions UP-TO-DATE
> Task :app:packageDebug UP-TO-DATE
> Task :app:createDebugApkListingFileRedirect UP-TO-DATE
> Task :app:assembleDebug UP-TO-DATE
```

Below the tasks, the message 'BUILD SUCCESSFUL in 53s' is displayed, along with '31 actionable tasks: 31 up-to-date'. At the bottom, a link to 'Build Analyzer results available' is shown.

FIG 3.7 Successfully build gradle

Chapter 4

METHODOLOGY/ TECHNIQUES USED

4.1 Development Tools

The entire development process has been subdivided into multiple parts like

1. Client Side (Android application)
2. Server Side.

The client-side development involves development of android app and user interface to control pc using button and touchpad. While server-side development involves the development of java program or application which creates server and make computer system to establish connection by reserving socket. For coding purpose, we use Visual studio code which is an Integrated Development Environment to compile and run codes.

1.Client-Side (Android Application)

1. Front End Development:

For the front-end development of the Android app, XML (eXtensible Markup Language) was primarily utilized. XML is a markup language similar to HTML but is more versatile and allows for the hierarchical structuring of data. XML was used to define the layout and structure of the app's user interface, including views, layouts, and resources.

1. In addition to XML, the Android app may have also utilized Java for implementing logic and functionality. Java is the primary programming language used for Android app development and is used to handle user interactions, process data, and perform other tasks within the app.
2. Third-party libraries may have been integrated into the Android app to enhance its functionality or provide additional features. These libraries could include:
 3. Retrofit: A type-safe HTTP client for Android and Java that simplifies the process of making network requests and handling responses.
 4. Glide: A fast and efficient image loading library for Android that simplifies the process of loading and caching images from remote sources.
 5. Material Components for Android: A set of customizable and ready-to-use UI components based on

Google's Material Design guidelines, which can be easily integrated into Android apps to provide a consistent and polished user interface.

6. Firebase: A mobile and web application development platform provided by Google, which offers various services such as real-time database, authentication, cloud storage, and more, making it easier to build and scale apps.
7. Development tools such as Android Studio IDE would have been used for coding the front end of the Android app. Android Studio provides a comprehensive environment for Android app development, offering features such as code editing, debugging, and testing, as well as integration with the Android SDK and other development tools.
8. XML for Layout: XML files were used to define the layout and structure of each screen or activity within the Android app. XML allows for the arrangement of various UI elements such as buttons, text fields, images, and more in a hierarchical manner.
9. View Binding: Android's view binding feature may have been utilized to interact with XML layout elements more efficiently and safely. View binding generates a binding class for each XML layout file, allowing developers to access views directly without the need for `findViewById()` calls.
10. Responsive Design: The app's front-end design likely incorporated principles of responsive design to ensure compatibility with various screen sizes and orientations. Layouts and UI elements were adjusted dynamically to provide a consistent and user-friendly experience across different devices.
11. Material Design Guidelines: The app may have adhered to Google's Material Design guidelines, which provide best practices for creating visually appealing and intuitive user interfaces. Material Design emphasizes elements such as consistent typography, elevation, and motion for a cohesive user experience.

Development tools such as Android Studio IDE would have been used for coding the front end of the Android app. Android Studio provides a comprehensive environment for Android app development, offering features such as code editing, debugging, and testing, as well as integration with the Android SDK and other development tools

2. Backend Development:

In the backend development of the Android app for the PC controller, several technologies and tools were utilized to facilitate communication between the Android device and the computer. Here are the key components and technologies used:

1. Java: Java programming language was utilized to implement the backend logic of the Android app. Java provides a robust and platform-independent environment for developing backend services and handling communication between different components of the application.
2. Socket Programming: Socket programming was employed to establish communication channels between the Android device and the computer. Sockets allow bidirectional data transfer between devices over a network, enabling real-time communication and interaction.
3. Inter-process Communication (IPC): Inter-process communication mechanisms were implemented to facilitate communication between the Android app and the computer's operating system. IPC allows different processes or applications to exchange data and coordinate their actions.
4. Android Debug Bridge (ADB) Client: The Android Debug Bridge (ADB) client was used to establish a connection between the Android device and the computer for debugging and testing purposes. ADB provides a command-line interface for communicating with Android devices and executing various commands.

4.2 Code Design

The code design is one of the most important concepts in any application development or software development procedure. A good code must exhibit certain properties like simplicity, readability, modularity, layering, design, efficiency, elegance, and clarity. Whenever a certain application is to be developed it is divided into multiple phases. These phases are written separately so as to avoid further debugging cost or avoid future problems. In our case we divide code development in certain phases for better convenience. The code design of various phases used in our project can be stated as

We added ESLint configuration to maintain code design:

1. **Lint Rules:** Lint comes with a set of built-in rules that analyze Android code for potential issues, such as performance bottlenecks, usability problems, and coding style violations. These rules cover various aspects of Android development, including XML layout files, Java code, and resource files.
2. **Rule Severity Levels:** Lint rules can have different severity levels, including Error, Warning, Information, and Ignore. Severity levels determine how lint issues are reported and whether they affect the build process. Errors and warnings are typically more severe and can prevent the app from being built successfully.
3. **Custom Rules:** In addition to built-in rules, developers can define custom lint rules to enforce project-specific coding standards and best practices. Custom rules are written using the lint API and can target specific code patterns or conventions that are important for the project.

4. **Lint Configuration Files:** Lint configurations are typically stored in XML files named **lint.xml** or **lint-options.xml**. These configuration files specify which lint rules to enable or disable, as well as the severity levels for each rule. Configuration files can be placed at the project level, module level, or even at the individual file level.
5. **Scope:** Lint configurations can be scoped to apply to specific modules, build variants, or flavors within a multi-module Android project. Scoping allows developers to tailor lint checks to different parts of the project and apply different configurations based on project requirements.
6. **Suppression Annotations:** Developers can suppress lint warnings and errors using annotations such as **@SuppressLint** or **@SuppressWarnings**. These annotations can be applied to specific code elements, such as methods, classes, or individual lines of code, to suppress lint checks for those elements.
7. **Integration with Gradle:** Lint configurations are typically applied and managed through the project's Gradle build scripts. Gradle provides tasks for running lint checks, generating lint reports, and configuring lint options. Developers can specify lint options and configuration files in the **lintOptions** block of the Gradle build script.
8. **Continuous Integration:** Lint configurations are often integrated into continuous integration (CI) pipelines to automate code analysis and enforce quality standards. CI systems such as Jenkins or GitLab CI can execute lint checks as part of the build process and report lint issues in the build logs or as artifacts.

```
pp > .eslintrc.js > [?] <unknown> > ⚡ extends
    ritesh-jadhav-pst, 2 weeks ago | 3 authors (shubham-pethe-pst and others)
1  module.exports = {
2    ...env: {
3      browser: true,
4      es6: true,
5    },
6    extends: [
7      "airbnb", shubham-pethe-pst, 4 weeks ago • Formated code with eslint
8      "plugin:@typescript-eslint/recommended",
9      "plugin:react/recommended",
10     "plugin:jsx-a11y/recommended",
11     "prettier",
12     "plugin:prettier/recommended",
13   ],
14   parser: "@typescript-eslint/parser",
15   parserOptions: {
16     ecmaFeatures: {
17       jsx: true,
18     },
19     ecmaVersion: "latest",
20     sourceType: "module",
21   },
22   plugins: ["@typescript-eslint", "react", "jsx-a11y", "prettier"],
23   rules: {
24     "prettier/prettier": [
25       "error",
26     ],
27   },
28 }
```

Fig3.8 Lint configuration

4.2.1. Socket Programming mechanism

Java offers some features like networking support, multithreading, multiplatform and portability that converts it in a suitable platform to be implemented in a distributed system. However, it has a problem of performance due to lack of efficient communication middleware, thus penalizing sync speed. Sockets are a low-level programming interface to manage networked communications in most of the network protocols. The Java Sockets can be used in every system that has Java Virtual Machine, but they are limited to the TCP/IP protocol. In systems with highspeed networks this communication system is not appropriate because it does not make use of the benefits of the environment. To create this communication system, the Android application will have a Server Socket running and awaiting client requests. To manage clients in a parallel manner, it will implement a system of threads that will serve each client concurrently, making use of the shared resources. On the other hand, clients will have Java client sockets that will open communication with the server and will exchange messages with the server until the connection is closed. Because the Java communication mechanism consists of the exchange of binary data; an interface of communication will be implemented. This mechanism consists of defining a set of objects that contains a subset of objects representing both requests and res

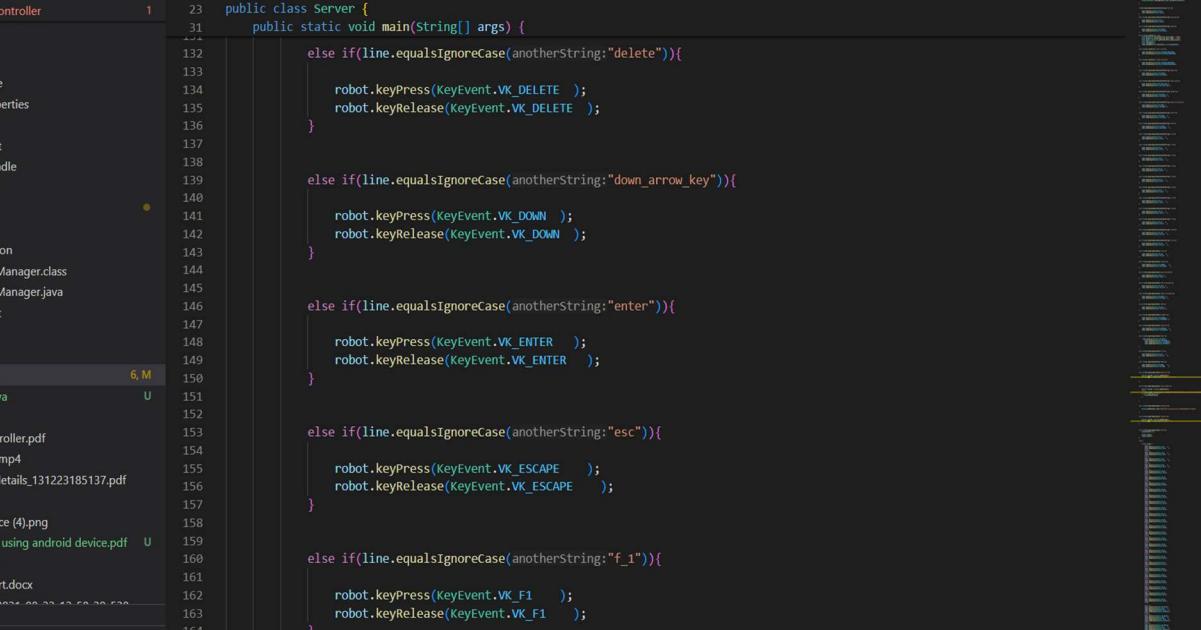
- **Socket Programming module:**

```
Server > J Server.java > Server > main(String[])
You, last month | 2 authors (Atharv Davale and others)
23 public class Server {
24     private static ServerSocket server = null;
25     private static Socket client = null;
26     private static BufferedReader in = null;
27     private static String line;
28     private static boolean isConnected=true;
29     private static Robot robot;
30     private static final int SERVER_PORT = 8080;
31     Run | Debug
32     public static void main(String[] args) {
33         boolean leftpressed=false;
34         boolean rightpressed=false;
35         System.out.println("Server has started");
36         try {
37             InetAddress localhost = InetAddress.getLocalHost();
38             System.out.println("Your current IP address is: " + localhost.getHostAddress());
39         } catch (UnknownHostException e) {
40             e.printStackTrace(); Atharv Davale, 5 months ago • Initial commit
41         }
42         try{
43             robot = new Robot();
44             server = new ServerSocket(SERVER_PORT);
45             client = server.accept();
46             System.out.println("Client is connected ");
47
48             in = new BufferedReader(new InputStreamReader(client.getInputStream()));
49
50         }catch (IOException e) {
51             System.out.println("Error in opening Socket");
52             System.exit(-1);
53         }catch (AWTException e) {
54             System.out.println("Error in creating robot instance");
55             System.exit(-1);
56         }
57     }
58 }
```

4.2.2 Event Handling at server side

The response of the user is received in the form of string from the connection that we made with socket programming so that response is handled by using if and else conditional statements and written the code for corresponding actions in corresponding if/else blocks.

```
Server.java 6, M
Server > J Serverjava > Server > main(String[])
23  public class Server {
31      public static void main(String[] args) {
32          ...
33
34          else if(line.contains(s:"left_click")){
35
36              robot.mousePress(InputEvent.BUTTON1_DOWN_MASK);
37              robot.mouseRelease(InputEvent.BUTTON1_DOWN_MASK);
38          }
39
40
41          else if(line.contains(s:"right_click")){
42
43              robot.mousePress(InputEvent.BUTTON3_DOWN_MASK);
44              robot.mouseRelease(InputEvent.BUTTON3_DOWN_MASK);
45          }
46
47
48          else if(line.equalsIgnoreCase(anotherString:"space")){
49
50              robot.keyPress(KeyEvent.VK_SPACE);
51              robot.keyRelease(KeyEvent.VK_SPACE);
52          }
53
54
55          else if(line.equalsIgnoreCase(anotherString:"caps_lock")){
56
57              robot.keyPress(KeyEvent.VK_CAPS_LOCK );
58              robot.keyRelease(KeyEvent.VK_CAPS_LOCK );
59          }
60
61
62          else if(line.equalsIgnoreCase(anotherString:"delete")){
63
64              robot.keyPress(KeyEvent.VK_DELETE );
65              robot.keyRelease(KeyEvent.VK_DELETE );
66          }
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
```



A screenshot of a Java application interface. The top bar shows "File Edit Selection View Go Run Terminal Help". The left sidebar has sections for "EXPLORER", "OUTLINE", "TIMELINE", and "JAVA PROJECTS". The main area displays a Java code editor for "Server.java" with line numbers 1 through 165. The code handles various key presses based on input strings. A status bar at the bottom shows "Atharv Davale, 5 months ago" and "Java: Warning". The bottom navigation bar includes icons for file operations like "main", "New", "Open", "Save", "Run", "Terminal", and "Help".

```
public class Server {
    public static void main(String[] args) {
        else if(line.equalsIgnoreCase(anotherString:"delete")){
            robot.keyPress(KeyEvent.VK_DELETE );
            robot.keyRelease(KeyEvent.VK_DELETE );
        }

        else if(line.equalsIgnoreCase(anotherString:"down_arrow_key")){
            robot.keyPress(KeyEvent.VK_DOWN );
            robot.keyRelease(KeyEvent.VK_DOWN );
        }

        else if(line.equalsIgnoreCase(anotherString:"enter")){
            robot.keyPress(KeyEvent.VK_ENTER );
            robot.keyRelease(KeyEvent.VK_ENTER );
        }

        else if(line.equalsIgnoreCase(anotherString:"esc")){
            robot.keyPress(KeyEvent.VK_ESCAPE );
            robot.keyRelease(KeyEvent.VK_ESCAPE );
        }

        else if(line.equalsIgnoreCase(anotherString:"f_1")){
            robot.keyPress(KeyEvent.VK_F1 );
            robot.keyRelease(KeyEvent.VK_F1 );
        }
    }
}
```

And for handling keyboard buttons like from A to Z we used switch case

The screenshot shows a Java development environment with the following details:

- File Explorer (Left):** Shows the project structure under "PC_CONTROLLER".
 - Client_Pc_Controller:** Contains .app, gradle, .gitignore, build.gradle, gradle.properties, gradlew, gradlew.bat, settings.gradle, ScreenShots, .vscode (with settings.json), and files BrightnessManager.class, BrightnessManager.java, manifest.txt, Server.class, and Server.jar.
 - Server:** Contains .vscode (with settings.json), files Server.java, ServerUI.java, and .gitattributes, and documents A12 PC Controller.pdf, demo video.mp4, Innovation_details_131223185137.pdf, LICENSE, mobile-service (4).png, and Pc controller using android device.pdf.
- Code Editor (Center):** Displays the content of **Server.java**. The code uses Robot API to handle key events (A through I).

```
public class Server {
    public static void main(String[] args) {
        switch (line) {
            case "a":
                robot.keyPress(KeyEvent.VK_A);
                robot.keyRelease(KeyEvent.VK_A);
            break;
            case "b":
                robot.keyPress(KeyEvent.VK_B);
                robot.keyRelease(KeyEvent.VK_B);
            break;
            case "c":
                robot.keyPress(KeyEvent.VK_C);
                robot.keyRelease(KeyEvent.VK_C);
            break;
            case "d":
                robot.keyPress(KeyEvent.VK_D);
                robot.keyRelease(KeyEvent.VK_D);
            break;
            case "e":
                robot.keyPress(KeyEvent.VK_E);
                robot.keyRelease(KeyEvent.VK_E);
            break;
            case "f": Atharv Davale, 5 months ago • Initial commit
                robot.keyPress(KeyEvent.VK_F);
                robot.keyRelease(KeyEvent.VK_F);
            break;
            case "g":
                robot.keyPress(KeyEvent.VK_G);
                robot.keyRelease(KeyEvent.VK_G);
            break;
            case "h":
                robot.keyPress(KeyEvent.VK_H);
                robot.keyRelease(KeyEvent.VK_H);
            break;
            case "i":
        }
    }
}
```
- Bottom Status Bar:** Shows the current file is "main", Java Warning, Atharv Davale, 5 months ago, Line 385, Col 31, Tab Size: 4, UTF-8, CRLF, Java, Go Live, and Prettier.
- Bottom Icons:** Includes icons for search, file operations, and system status.

4.2.3 Brightness control

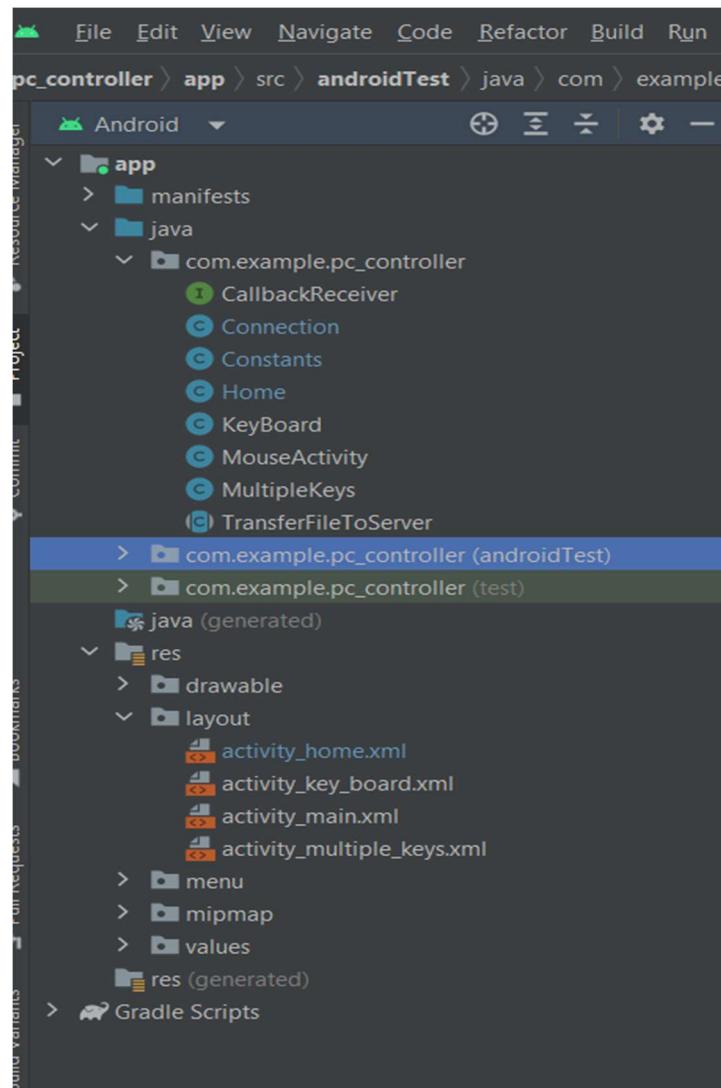
The BrightnessManager class in the provided Java file is responsible for controlling the brightness of a monitor using PowerShell commands in a Windows environment. The setBrightness method takes an integer parameter brightness, which represents the desired brightness level. Inside the method, a PowerShell command string is constructed based on the specified brightness value. This command string includes PowerShell commands to set the monitor brightness using the WmiMonitorBrightnessMethods class from the root\wmi namespace. The constructed PowerShell command is executed using the Runtime.getRuntime().exec(command) method, which initiates a new process to run the PowerShell command.

```
Server.java 6 M J BrightnessManager.java 1 X
server > J BrightnessManager.java > BrightnessManager > setBrightness(int)
You, last month | 2 authors (Atharv Davale and others)
1 import java.io.BufferedReader;
2 import java.io.IOException;
3 import java.io.InputStreamReader;
You, last month | 2 authors (Atharv Davale and others)
4 public class BrightnessManager {
5     public static void setBrightness(int brightness)
6         throws IOException {
7
8         String s = String.format(format:"$brightness = %d;", brightness)
9             + "$delay = 0;"
10            + "$myMonitor = Get-WmiObject -Namespace root\\wmi -Class WmiMonitorBrightnessMethods;"
11            + "$myMonitor.wmisetbrightness($delay, $brightness);"
12        String command = "powershell.exe " + s;
13
14        Process powerShellProcess = Runtime.getRuntime().exec(command);
15
16        powerShellProcess.getOutputStream().close(); Atharv Davale, 5 months ago • Initial commit
17
18
19        String line;
20
21        BufferedReader stderr = new BufferedReader(new InputStreamReader(
22            powerShellProcess.getErrorStream()));
23        line = stderr.readLine();
24        if (line != null)
25        {
26            System.err.println(x:"Standard Error:");
27            do
28            {
29                System.err.println(line);
30            } while ((line = stderr.readLine()) != null);
31
32        }
33        stderr.close();
34
35    }
}
q
φ Atharv Davale, 5 months ago Ln 16, Col 53 Spaces: 4
```

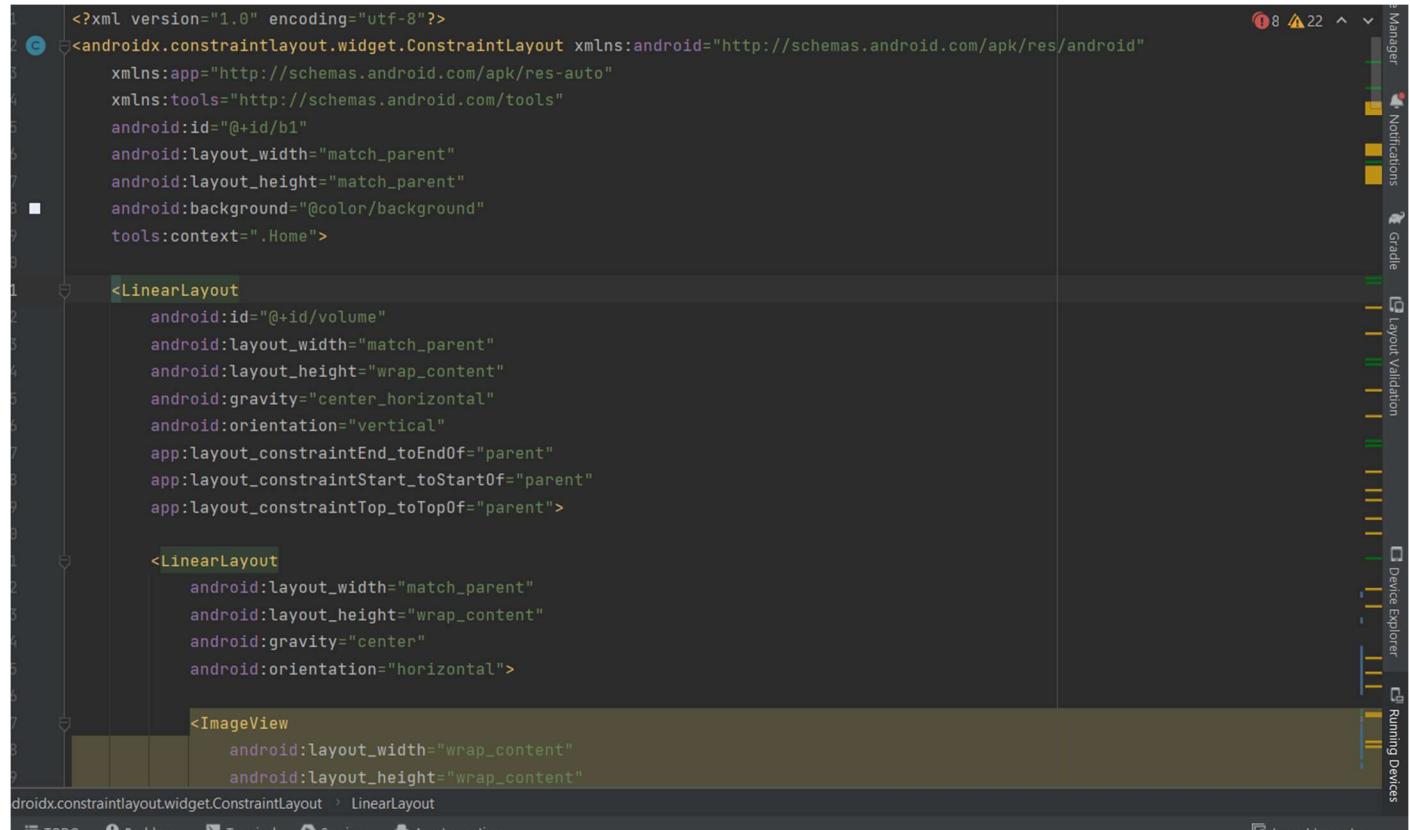
4.3. User interface design:

In the Android app for the PC controller, the user interface design focuses on providing intuitive controls and easy navigation for the user to interact with the app and control their PC remotely. The user interface (UI) elements are designed using XML layout files, which define the structure and appearance of each screen or activity in the app. Here's a brief overview of the user interface design for the Android app

We have separate file structure for each component . Breaking down the user interface components into separate files is a common practice in software development that significantly enhances code readability and maintainability. By organizing each component into its own file, developers can isolate the functionality and styling specific to that component, making it easier to understand and modify. This approach also promotes code reuse, as components can be easily referenced and imported into different parts of the application without duplicating code. Additionally, separating components into individual files simplifies collaboration among team members, as each developer can focus on a specific component without being overwhelmed by the complexity of the entire application. Overall, this modular approach to organizing UI components streamlines development workflows, reduces code complexity, and facilitates better code comprehension for everyone involved in the project.



Home Page component(activity_home.xml)



```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/b1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/background"
    tools:context=".Home">

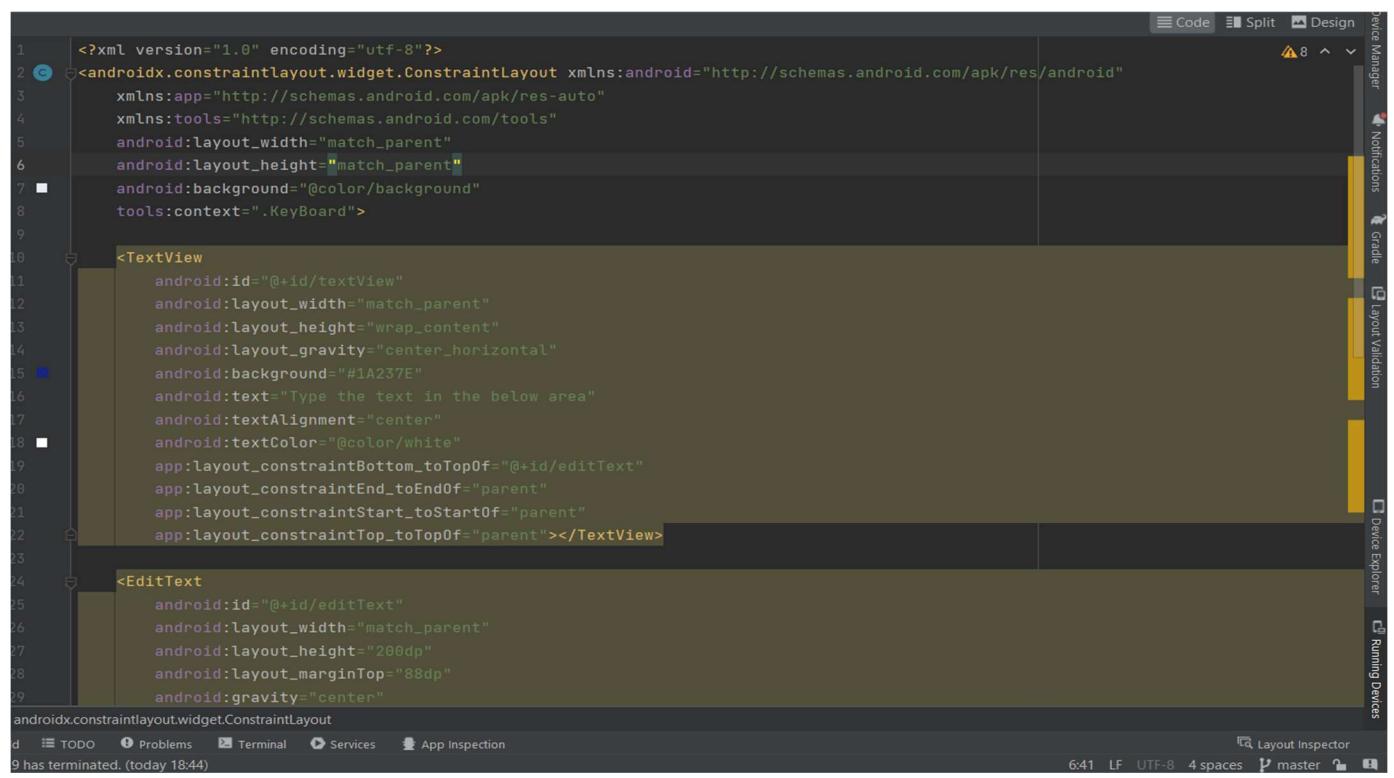
    <LinearLayout
        android:id="@+id/volume"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:orientation="vertical"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:orientation="horizontal">

                <ImageView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:background="#FFFF00"/> 
            
        

```

MousePad Component(activity_Mousepad.xml)



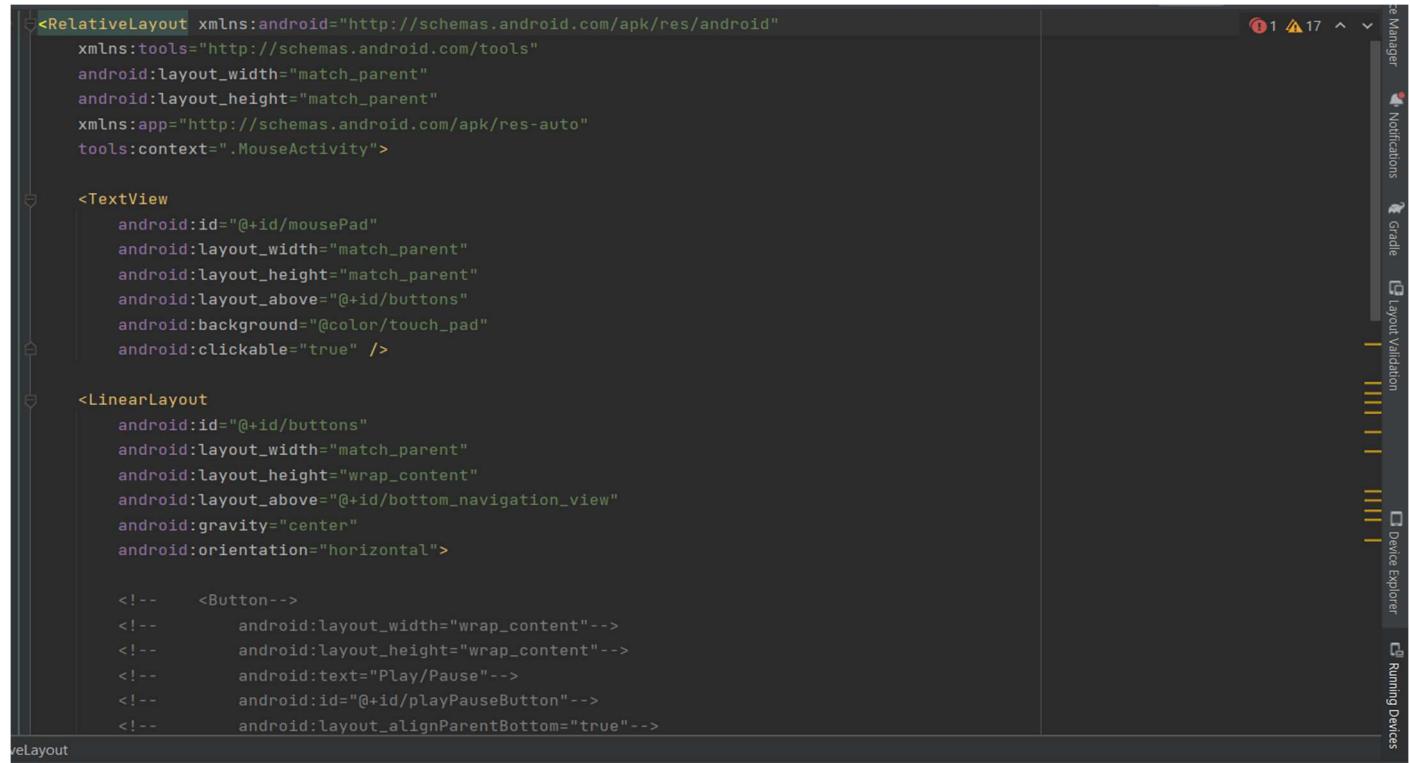
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/background"
    tools:context=".KeyBoard">

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:background="#1A237E"
        android:text="Type the text in the below area"
        android:textAlignment="center"
        android:textColor="@color/white"
        app:layout_constraintBottom_toTopOf="@+id/editText"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"></TextView>

    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:layout_marginTop="88dp"
        android:gravity="center"
        android:text="Type the text in the below area" />

```

Keyboard Component (activity_Keyboard.xml)



The screenshot shows the Android Studio interface with the XML code for the activity_Keyboard.xml file. The code defines a RelativeLayout containing a TextView and a LinearLayout. The TextView has an id of @+id/mousePad and is positioned above the LinearLayout. The LinearLayout has an id of @+id/buttons and contains several button definitions, each starting with a '<!--' tag. The code uses various Android XML attributes like android:id, android:layout_width, android:layout_height, and android:gravity. The right side of the screen shows the Android Studio navigation bar with icons for Device Manager, Notifications, Gradle, Layout Validation, Device Explorer, and Running Devices.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/mousePad"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_above="@+id/buttons"
        android:background="@color/touch_pad"
        android:clickable="true" />

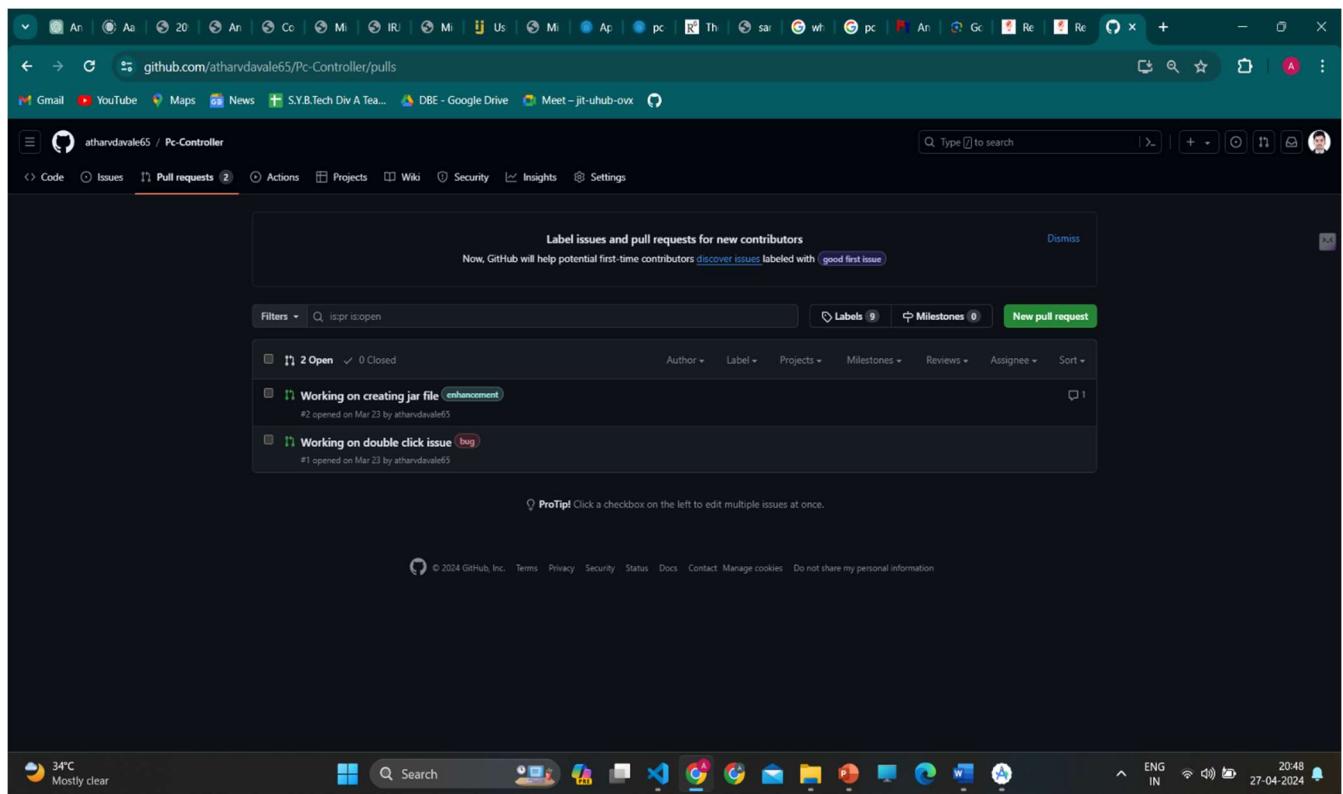
    <LinearLayout
        android:id="@+id/buttons"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_above="@+id/bottom_navigation_view"
        android:gravity="center"
        android:orientation="horizontal">

        <!-- <Button-->
        <!--     android:layout_width="wrap_content"-->
        <!--     android:layout_height="wrap_content"-->
        <!--     android:text="Play/Pause"-->
        <!--     android:id="@+id/playPauseButton"-->
        <!--     android:layout_alignParentBottom="true"-->
    </LinearLayout>
</RelativeLayout>
```

Version Control

For version control we are using git and github. we have common a common repository and every one creates a separate branch for his work and after completion of his work he push the code and make new pull request for his changes then everyone from team review that branch and test that branch .If everything looks good and changes meets to requirements then approve that Pull request .If everyone approves that pull request then team leader merge that branch into our main master branch.

GitHub is a widely used version control platform that provides developers with a centralized repository to store, manage, and collaborate on their codebase. By leveraging Git, GitHub enables teams to track changes, coordinate work, and maintain a complete history of their project's development. Developers can create branches to work on new features or bug fixes independently, and then merge their changes back into the main branch once they are ready. GitHub's user-friendly interface and collaboration features, such as pull requests and code reviews, streamline the development process and foster collaboration among team members. Additionally, GitHub offers robust integration with popular development tools and services, making it an essential tool for modern software development workflows. Overall, GitHub empowers developers to efficiently manage and collaborate on their projects while ensuring version control and code quality.



Chapter 5

SYSTEM SECURITY MEASURES

Security is about risk management, so it is important to start with an understanding of the risk associated with the blockchain solutions. The specific risks of a blockchain solution depends on the type of blockchain being used. Let's take a look at the various types of blockchains with decreasing level of risks and increasing levels of security:

- 1) **Secure Communication:** Implement secure communication protocols, such as HTTPS or SSL/TLS, to encrypt data transmitted between the Android device and the controlled PC. This helps prevent eavesdropping and Man-in-the-Middle (MITM) attacks.
- 2) **Authentication:** Require users to authenticate themselves before accessing the PC controller app. Implement strong authentication mechanisms, such as password-based authentication, biometric authentication (e.g., fingerprint or face recognition), or two-factor authentication (2FA), to verify the identity of users.
- 3) **Authorization:** Implement role-based access control (RBAC) to restrict access to specific features or functionalities within the PC controller app based on the user's role or privileges. Ensure that users can only perform actions that they are authorized to perform.
- 4) **Secure Storage:** Store sensitive information, such as user credentials or authentication tokens, securely on the Android device using encryption techniques. Avoid storing sensitive data in plaintext or insecure locations that could be accessed by malicious actors.
- 5) **Input Validation:** Validate user input and command parameters to prevent injection attacks, such as SQL injection or command injection, which could be used to execute arbitrary commands on the controlled PC.
- 6) **Code Signing:** Sign the Android app's code with a digital certificate to verify its authenticity and integrity. This helps prevent tampering with the app's code and ensures that users are downloading legitimate software from trusted sources.
- 7) **Secure Configuration:** Configure the PC controller app and the controlled PC securely by disabling unnecessary services, ports, or features that could pose security risks. Keep both the Android device and the PC up to date with security patches and updates.
- 8) **Logging and Monitoring:** Implement logging and monitoring mechanisms to track and audit user activities within the PC controller app. Monitor for suspicious or unauthorized behavior and promptly respond to security incidents.

9) Secure Development Practices: Follow secure coding practices and guidelines when developing the PC controller app to minimize the risk of vulnerabilities and security flaws. Regularly conduct security assessments, code reviews, and penetration testing to identify and address potential security issues.

5.1 Security Threats

When developing a PC controller using an Android device, it's crucial to be aware of potential security threats that could compromise the integrity and confidentiality of the controlled PC and the data transmitted between the Android device and the PC. One of the primary security threats is unauthorized access, where malicious actors attempt to gain unauthorized control of the PC or intercept sensitive information transmitted between the Android device and the PC. This can occur through various means, such as exploiting vulnerabilities in the PC controller app or the controlled PC's operating system, intercepting unencrypted communication channels, or bypassing authentication mechanisms. Another significant security threat is data interception or eavesdropping, where attackers monitor and capture sensitive data, such as login credentials or command instructions, transmitted between the Android device and the PC. Additionally, the PC controller app may be vulnerable to various forms of attacks, including code injection, cross-site scripting (XSS), or denial-of-service (DoS) attacks, which could compromise the app's functionality or allow attackers to execute arbitrary code on the controlled PC. It's essential to implement robust security measures, such as encryption, authentication, input validation, and secure communication protocols, to mitigate these threats and protect the integrity and confidentiality of the PC controller system. Regular security assessments, code reviews, and penetration testing can help identify and address potential vulnerabilities before they can be exploited by attackers. Additionally, user education and awareness about security best practices can help mitigate the risks associated with social engineering attacks or user negligence. By proactively addressing security threats and implementing appropriate countermeasures, developers can enhance the overall security posture of the PC controller system and ensure the safety of both the users and the controlled PC.

5.2 Security Controls

- API security best practices are used to safeguard API-based transactions.
- Data classification are adopted for the approach to safeguard data/information.
- The appropriate endorsement policies are defined and endorsed based on business contracts.
- Secrets-store for both application and privileged access is leveraged.

Chapter 6

EXPERIMENTAL RESULTS/ OUTPUTS

- User interface:

Home Page:

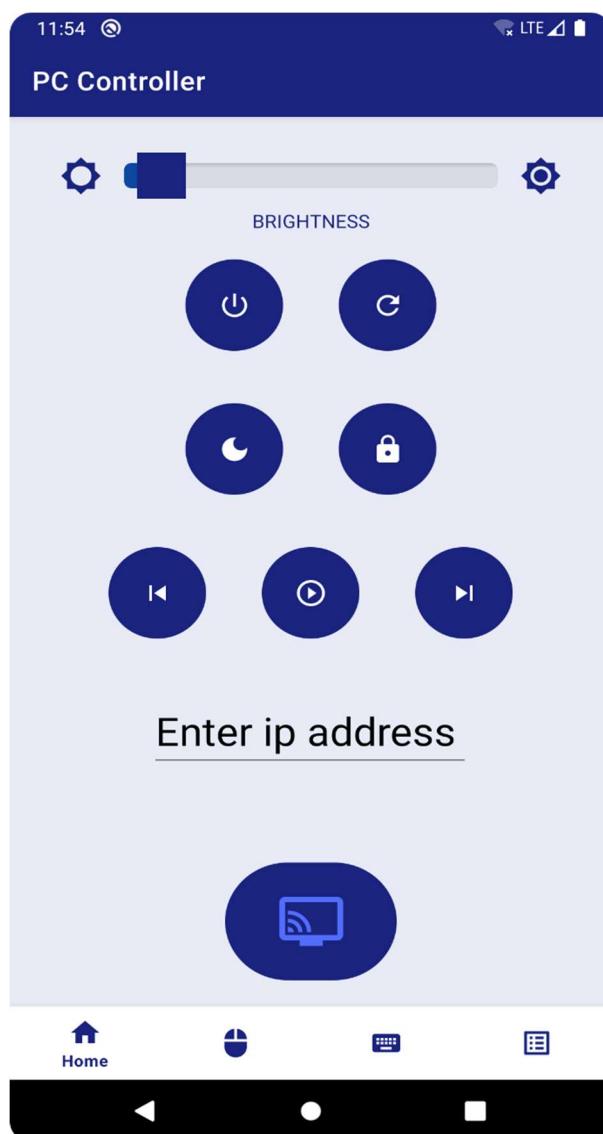


FIG.6.1 Home Page

Mouse-Pad: • Castin



FIG.6.2 Mouse-Pad page

In the above figure user can use there touch screen as mousepad can control cursor by using there mobile phone screen and also there are two buttons provide which works as right click and left click of the mouse

Keyboard:

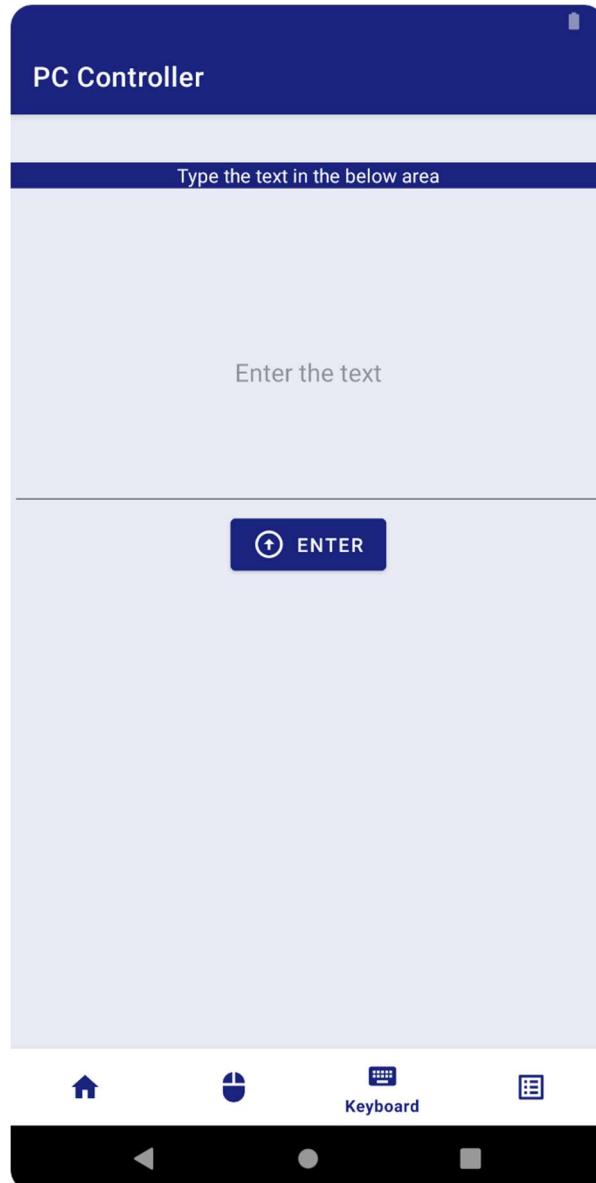


FIG.6.3. keyboard page

This is keyboard page where a textfield is provided in that user can type any thing by using there mobile phones keyboard and whatever the user types is reflected in you computer also

Multi-keys Page:

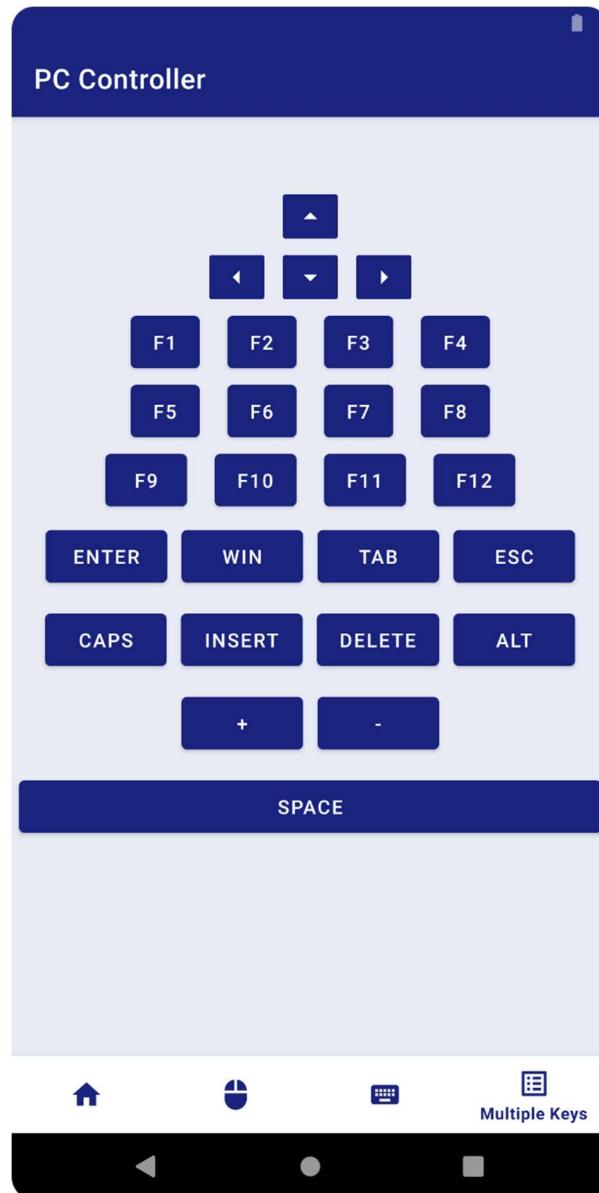


FIG.6.4. Multi keys page

This page contains all the keys that a computer keyboard have you can use that this to perform actions on the your computer.

Server-side output:

```
PS C:\Users\David\OneDrive\Desktop\Pc Controller> & 'C:\Program Files\Java\jdk1.8.0_144\bin\java.exe'  
a\Roaming\Code\User\workspaceStorage\b5bd1abecd81b796d48e2ea2cf21cfed\redhat.java\jdt_ws\jdt.ls-java-pr  
Server has started  
Your current IP address is: 192.168.0.104
```

FIG.6.5 Server-side output

This is server side output means output on the computer when you start application as we can see in above fig when you start the application it creates server and prints message as server has started and generates a custom ip address which that we have to enter in client side application to connect to the server.

Chapter 7

CONCLUSIONS AND FUTURE SCOPE

CONCLUSION:

We are developing a desktop application which acts as server on PC/Laptop & mobile application which acts as client. This allows user to use Smart phone's keypad & mouse pad to operate/control Laptop/PC. Also it will allow user to play games on their PCs through Smart phone using sensor feature as well as directional key games. Thus we are implementing a system that provides ease and convenience. And will also allow the user to make full utilization of his/her Smart phone.

This project explores the possibility of controlling the computer remotely using an Android phone device. The proposed prototype is able to control a lot of operations a normal computer keyboard and mouse would perform. It practically turns a mobile phone into a wireless keyboard and mouse using a wireless network via a portable mobile device running under an Android Platform Operating System. It helps mobile phone users on facilitating their work in study life, home life or working life, where the use of the prototype helps in easing the device control. It is proven that this project would relieve a pain in the neck and also the normal back ache due to constantly sitting at a particular place. With the help of this prototype, these stressful moments will be minimized as users will be having a very relaxed position as intended. This is a convenient application for simple operations and for manipulating such computer without the keyboard and mouse been connected.

FUTURE SCOPE:

Our group will be creating a Remote Access application for Android phones. This application will run on a mobile platform (Android). Which provides services for data transfer, file transfer and it has ability to view the remote location on android phones. For wireless connection Bluetooth as well as Wi-Fi are embedded to it.

A] TV can be controlled by mobile. In future TV can be controlled by a computer (already started in some areas).

B] The whole desktop will be controlled by an android phone where we can use internet explorer, windows media player, word operation, and games of desktop clients through an android phone.

C] As a continuation of work in this application, we would include the encryption algorithm to prevent data leakage. We will also put efforts for displaying the screen of the target PC on the android phone itself for better visualization.

References

- [1] *Android*. <http://www.android.com> Retrieved March 1st, 2011.
- [2] *What is Android* <http://developer.android.com/guide/basics/whatis-android.html> Retrieved March 1st, 2011
- [3] *Lingyan Bi, Weining Wang, Haobin Zhong, Wenxuan Liu, "Design and Application of Remote Control System Using Mobile Phone with JNI Interface"*, *The 2008 International Conference of Embedded Software and Systems Symposia (ICESS2008)*, 2008, pp.416-419
- [4] *Michael Spreitzenbarth, "Tools and Processes for Forensic Analyses of smartphones and Mobile Malware"*, *6. GI FG SIDARGraduierten-Workshop ueber Reaktive Sicherheit (SPRING)*, March 22th, 2011
- [5] *Xinfang Lee, Chunghuang Yang, Shihjen Chen, Jainshing Wu, "Design and Implementation of Forensic System in Android Smart Phone"*, *The 5th Joint Workshop on Information Security*, 2009
- [6] *Enck, W., Ongtang, M., McDaniel, P., "Understanding Android Security"*, *Security & Privacy*, IEEE, Jan.-Feb. 2009, Volume 7, Issue 1, pp.50-57
- [7] *Adam, Skurski, Bartłomiej Swiercz, "VNC-based Remote Control for Symbian OS smartphones"*, *MIXDES (Mixed Design of Integrated Circuits and Systems) 2009*, June 25-27, 2009
- [8] *Samsung, "KiesManual"*, <http://www.samsung.com/es/support/mobilesoftwaremanual/mobilesoftwaremanual.do?page=MOBILE.SOFTWARE.MANUAL>, Retrieved March 1st, 2011.
- [9] *T. Richardson, Q. Stafford-Fraser, K. Wood and A. Hooper, "Virtual networking computing"*, *Internet Computing*, Vol. 2, No. 1pp.33-38, 1998
- [10] *Android Developers, "AndroidDebugBridge"*, <http://developer.android.com/guide/developing/tools/adb.html>, Retrieved March 1st, 2011
- [11] *Android Developers, Android SDK*. <http://developer.android.com/sdk/> Retrieved March 1st, 2011.
- [12] *Android Developers, "USB Host and Accessory"*, <http://developer.android.com/guide/topics/usb/index.html>, Retrieved June 1st, 2011
- [13] *Dean Jezard, Johnny Makkar, David Holding-Parsons, Google Android Whitepaper, TigerSpike*, 2008,
- [14] *Nielsen, "U.S. Smartphone Market: Who's the Most Wanted?"*, <http://blog.nielsen.com/nielsenwire/?p=27418>, April 26th, 2011
- [15] *T. Richardson, "The RFB Protocol"*, Tech. rep., RealVNC Ltd, 2007
- [16] *Cynthia Taylor, Joseph Pasquale, "Improving Video Performance In VNC Under High Latency Conditions"*, *Collaborative Technologies and Systems (CTS), 2010 International Symposium on*, 17-21 May 2010, pp.26-35
- [17] *A. P. Rajsekhar, Socket Programming in Java*,
- [18] *Asaf Shabtai, Yuval Fledel, Uri Kanonov, Yuval Elovici and Shlomi Dolev, A State-of-the-Art*

Review of Security Mechanisms, Cornell University, 2009,

[19] Portokalidis, G., Homburg, P., Anagnostakis, K., Bos , H., “Paranoid Android: Versatile Protection For Smartphones ”, In Proceedings of the 26th Annual Computer Security Applications Conference (December 2010), pp. 347-356.

[20] William Enck, Understanding Android Security, IEEE Security & Privacy, 2009,

[21] Siliang Li, Gang Tan, Finding Bugs in Exceptional Situations of JNI Programs, Lehigh University, 2009,

[22] Jose Pereira, Droid VNC Server Application,

[23] Alexandre Thiel, AndroidScreencast,

[24] Liu, Y., Anshus, O.J., “Improving the Performance of VNC for High-Resolution Display Walls ”, International Symposium on Collaborative Technologies and Systems, 2009. CTS '09, 18-22 May, 2009, pp. 376-383.

SPONSORSHIP LETTER:

TechnoWings

IT SOLUTIONS

Plot No 64, Raghvendra Nagar, Near Airport, Hotgi Road,
Solapur, Maharashtra pin 413003 Mobile -7588216108

Date: 13/01/2024

To Whomsoever It May Concern

This is to certify that the student of Computer Science and Engineering Department from SVERI's College of Engineering, Pandharpur are allowed to develop software module for our TechnoWings IT Solutions, under the title of "**PC Control using android device**".

The Details of operation will be provided as per the requirements.

The students are as follows:

Atharv milind Davale
Amrut yogesh virdhe
Abhijeet Balkrishna surshetwar
Rushikesh Rajesh waghule
Digvijay sambhaji shinde

Regards,



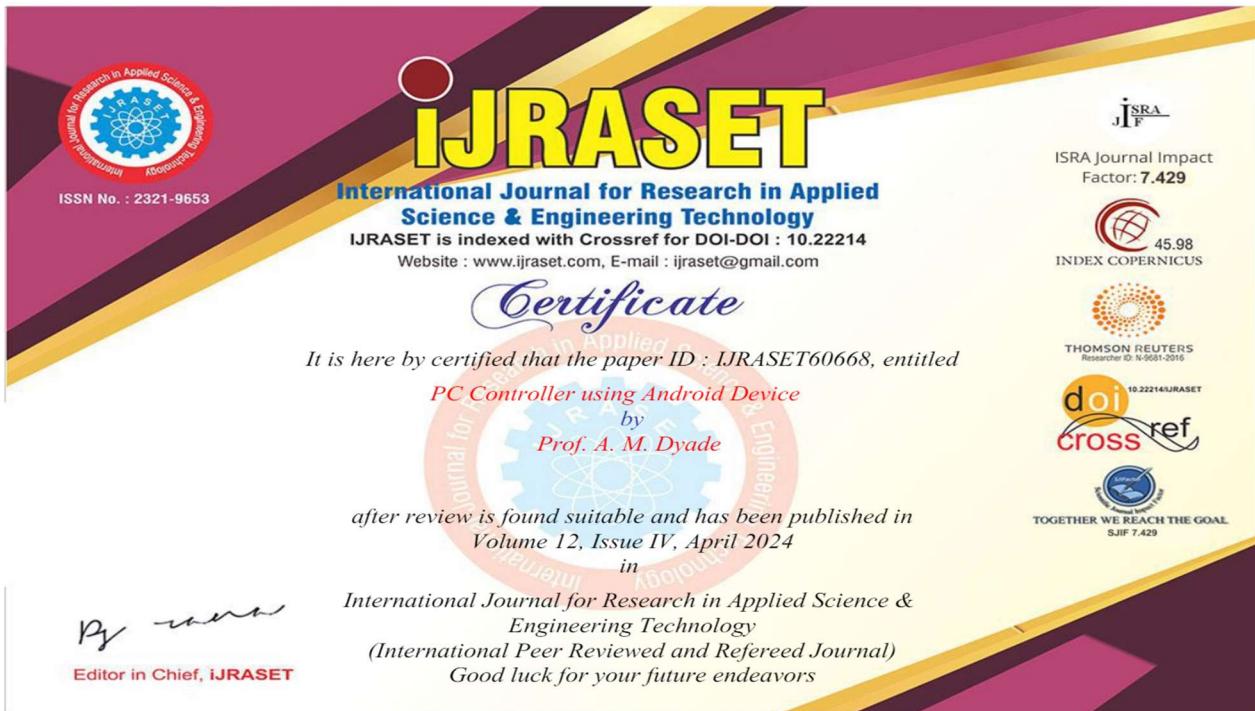
Bahubali.M.Kandale

TechnoWings
IT SOLUTIONS



+91 7588216108 ; +91 9326076449
technowings1@gmail.com
www.technowingsindia.com

Paper Publication Certificates:







ISSN No. : 2321-9653

IJRASET

International Journal for Research in Applied
Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

JIF

ISRA Journal Impact
Factor: 7.429



45.98
INDEX COPERNICUS



THOMSON REUTERS



doi 10.22214/IJRASET



TOGETHER WE REACH THE GOAL.
SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET60668, entitled

PC Controller using Android Device

by

Mr. Abhijeet Surshetwar

*after review is found suitable and has been published in
Volume 12, Issue IV, April 2024*

in

*International Journal for Research in Applied Science &
Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors*

By _____

Editor in Chief, IJRASET

PC Controller using Android Device

Prof.A.M.Dyade¹, Mr Atharv Davale², Mr Rushikesh Waghule³, Mr Digvijay Shinde⁴, Mr Amrut Virdhe⁵, Mr. Abhijeet Surshetwar⁶

¹Assistant Professor, ^{2, 3, 4, 5, 6}Student, Department of Computer Science and Engineering SVERI's College of Engineering, Punyashlok Ahilyadevi Holkar Solapur Vidyapeeth., Gopalpur, Pandharpur, Maharashtra, India

Abstract: From some years there has been a significant evolution in mobile or smartphone computing and communication devices like mobile phones, media players and many more. This project e is proposed to be able to perform most of the actions a normal computer keyboard and mouse can accomplish. Wireless presentation controller ensures good freedom of movement. but, most of such devices do not allow user full operation on the computer, like running the program, moving or closing an application window, etc. This project proposed the design and implementation of converting smartphones into computer remote controllers by which user can wirelessly operate a computer. We proposed the system which can reduce the strain of sore moment with the use of computer. It enables the user to move cursor, click operation and applications, play with media such like forward, rewind, pause, run, and increase or reduce the volume of a media file, multi-touch scrolling, pinch gesture for zoom in and out. This could be achieved by using the proposed application that occurs to the computer network via Wi-Fi which can connects both system with each other, then command from the mobile phone that remotely controls the computer.

I. INTRODUCTION

There are several situations where we want to wirelessly and comfortably operate a computer, where the computer screen is displayed onto a big screen through a projector or big-screen television, such as classrooms, conference/meeting rooms, mobile, workgroup project environments and modern office environments, and even living rooms. Several specifically designed devices are available on the market for the goal of operating computers remotely and wirelessly. Wireless keyboard, uses each of two Bluetooth or wireless USB mini-receiver plugged into the USB port of computer for the communication among the keyboard and the computer. Some wireless keyboards have a touchpad for controlling the mouse pointer. Wireless presentation controller, as shown in Fig. 1b, allows user to operate his/her computer remotely for PowerPoint presentation over Bluetooth connection. It usually has several buttons, including mouse-left, mouse-right, next slide and previous slide buttons (for PowerPoint presentation), and even a small size rectangle touchpad for moving mouse cursor.

II. LITERATURE REVIEW

The advent of sophisticated mobile platforms, particularly Android, has significantly altered the landscape of remote device management and control. Among the myriad contributions to this evolving domain, several key studies stand out for their innovative approaches and foundational principles. For instance, Lingyan Bi et al. [1] introduced a groundbreaking method to design an Android-based Remote Control System utilizing the Java Native Interface (JNI), markedly enhancing user convenience by bridging the gap between native Android applications and lower-level system operations. Concurrently, Michael Spreitzenbarth et al. [2] embarked on an exploratory journey into Smartphone Mobile Malware, unveiling novel forensic analysis techniques that underscore the critical need for secure mobile computing environments. Further extending the discourse on Android system security, Xinfang Lee et al. [3] detailed an Android-based Forensic System, paving the way for advanced methodologies in digital forensic investigations on mobile platforms.

Moreover, the work of Enck, W et al. [4] presents a robust framework for a secure Android Remote controlling mechanism, emphasizing the importance of safeguarding remote transactions executed from dispersed locales. Complementing these studies, T. Richardson et al. [5] explored the application of internet-based Android applications, illustrating the practical implications of leveraging internet computing to enhance the functionality and reach of mobile devices. This body of research collectively highlights the transformative impact of Android in reshaping not only the functionality and security paradigms of mobile devices but also their integration into the broader ecosystem of personal computing.

The proliferation of smartphones has not only redefined the boundaries of what mobile devices can achieve but has also introduced novel challenges and opportunities in the integration of these devices with traditional computing systems.

As technological advancements continue to equip mobile devices with capabilities once exclusive to PC architectures, the imperative to seamlessly integrate and control these devices from afar has never been more pronounced. Among the solutions emerging in this space, Virtual Networking Computing (VNC) systems have gained prominence for their ability to offer remote visualization.

III. SYSTEM ARCHITECTURE

In the proposed system there is a client-server architecture. In which android mobile phone is a client and PC/Laptop is a server. Client sends request to the server for connection establishment via wireless communication (Wi-Fi). Connection is established using handshaking. After connection establishment ports are assigned and sockets are established at both ends for communication. Desktop application is in Java and mobile application is in android. Windows registry is hierarchical database which stores some configuration setting options and some processes which helps to control mouse movements. When the user control mouse from android mobile then in background the process of desktop application will communicate with mouse process which is in windows registry. And this This Is Another Level 2 Heading

IV. INTERPROCESS COMMUNICATION

In computing, InterProcess Communication (IPC) is a set of methods for the exchange of data among multiple threads in one or more processes. Processes may be running on one or more computers connected by a network. IPC methods are divided into methods for message passing, synchronization, shared memory, and remote procedure calls (RPC). The method of IPC used may vary based on the bandwidth and latency of communication between the threads, and the type of data being communicated. There are several reasons providing an environment that allows process cooperation:

- 1) Information sharing
- 2) Computational speedup
- 3) Modularity
- 4) Convenience
- 5) Privilege separation

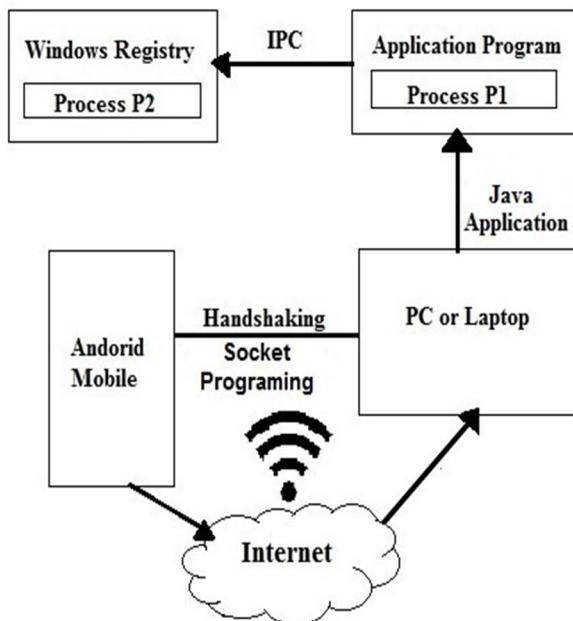


Fig. 1. This is a figure caption. It appears directly underneath the figure

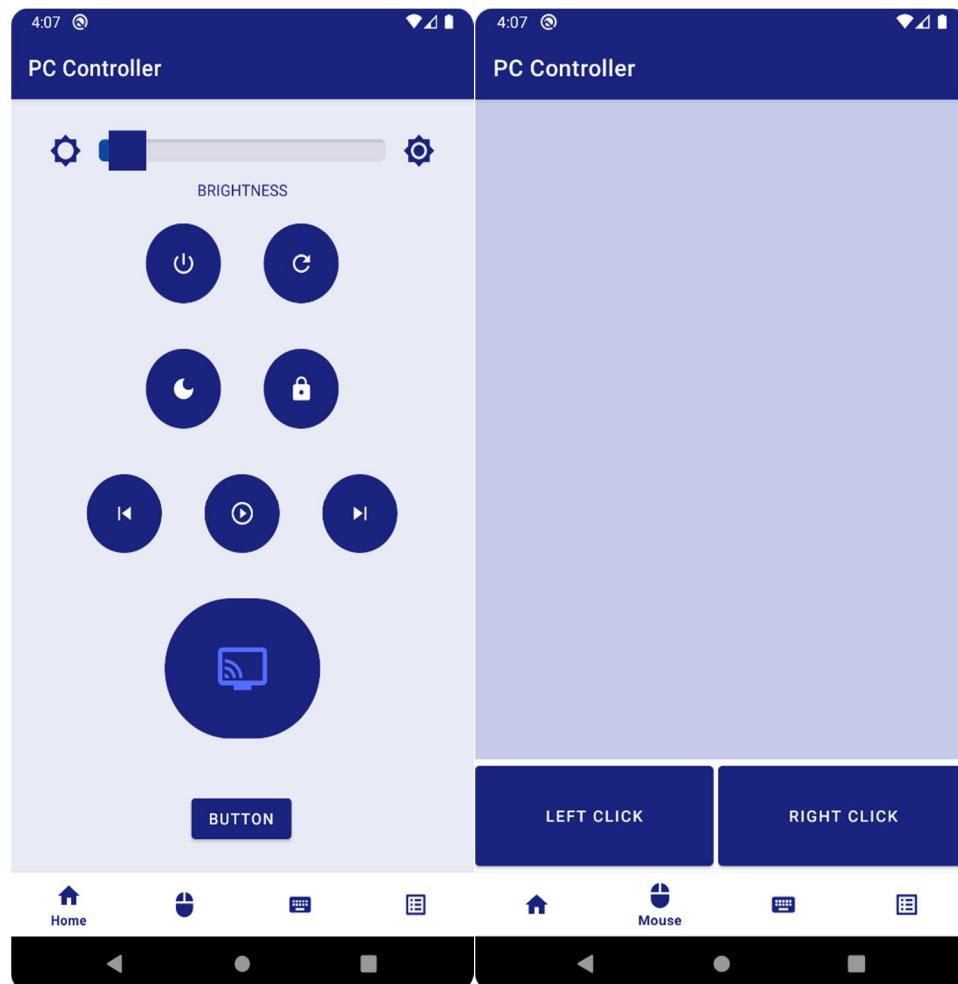
IPC may also be referred to as inter-thread communication and inter-application communication. InterProcess Communication (IPC) is the transfer of data among processes. For example, a web browser may request a web page from a web server, which then sends HTML data. This transfer of data usually uses sockets in a telephone-like connection and data exchange techniques between processes. References

V. SOCKET PROGRAMMING

A socket is a bidirectional communication device that can be used to communicate with another process on the same machine or with a process running on other machines. Sockets are the only InterProcess Communication that permit communication between processes on different devices such as here in this project the devices are laptop and android mobile. Normally, a server runs on a specific computer and has a socket that is bound to a specific port number. The server just waits, listening to the socket for a client to make a connection request. The client knows the hostname of the machine on which the server is running and the port number on which the server is listening. To make a connection request, the client tries to rendezvous with the server on the server's machine and port. The client also needs to identify itself to the server so it binds to a local port number that it will use during this connection. This is usually assigned by the system. A client's connection request is sent to server if everything goes well, the server accepts the connection. Upon acceptance, the server gets a new socket bound to the same local port and also has its remote endpoint set to the address and port of the client. It needs a new socket so that it can continue to listen to the original socket for connection requests while tending to the needs of the connected client. The connection is made on the client side, if the connection is accepted, a socket is successfully created and the client can use the socket to communicate with the server

VI. REMOTE CONTROL ARCHITECTURE PROPOSAL

The architecture proposed in this paper consists of a remote control architecture of mobile devices on the Android platform based on a client / server model oriented to services. The server layer is performing the services of mobile device management and accepts the connection from different clients. The client layer, available from a remote device, performs the interaction between the control equipment and the monitored device. As can be observed in Figure 1, the architecture offers several types of connection to different clients in order to allow the remote control to all the users. Below, the features that will implement the architecture will be listed and will be determined the chosen solution will be determined.





REFERENCES

- [1] Android. <http://www.android.com> Retrieved March 1st, 2011.
- [2] What is <http://developer.android.com/guide/basics/whatis-android.html> Retrieved March 1st, 2011
- [3] Lingyan Bi, Weining Wang, Haobin Zhong, Wenxuan Liu, "Design and Application of Remote Control System Using MobilePhone with JNI Interface", The 2008 International Conference of Embedded Software and Systems Symposia (ICESS2008), 2008,pp.416-419
- [4] Michael Spreitzenbarth, "Tools and Processes for Forensic Analyses of smartphones and Mobile Malware", 6. GI FG SIDARGraduierten-Workshop ueber Reaktive Sicherheit (SPRING),March 22th, 2011
- [5] Xinfang Lee, Chunghuang Yang, Shihjen Chen, Jainshing Wu,"Design and Implementation of Forensic System in AndroidSmart Phone", The 5th Joint Workshop on Information Security, 2009
- [6] Enck, W., Ongtang, M., McDaniel, P., "Understanding AndroidSecurity", Security & Privacy, IEEE, Jan.-Feb. 2009, Volume 7,Issue 1, pp.50-57
- [7] Adam, Skurski, Bartłomiej Swiercz, "VNC-based Remote Control for Symbian OS smartphones", MIXDES (Mixed Design of Integrated Circuits and Systems) 2009, June 25-27, 2009
- [8] Samsung, "Kies Manual", <http://www.samsung.com/es/support/mobilesoftwaremanual/mobilesoftwaremanual.do?page=MOBILE.SOFTWARE.MANUAL>, Retrieved March 1st, 2011.
- [9] T. Richardson, Q. Stafford-Fraser, K. Wood and A. Hooper, "Virtual networking computing", Internet Computing, Vol. 2, No. 1pp.33-38, 1998
- [10] Android Developers, "Android Debug Bridge",<http://developer.android.com/guide/developing/tools/adb.html>,Retrieved March 1st, 2011
- [11] Android Developers, Android SDK.<http://developer.android.com/sdk/> Retrieved March 1st,2011.
- [12] Android Developers, "USB Host and Accessory",<http://developer.android.com/guide/topics/usb/index.html>,Retrieved June 1st, 2011
- [13] Dean Jezard, Johnny Makkar, David Holding-Parsons, Google