

Smart File Management Bot Using IOT and Python

Prof. Snehal patel¹, Aboli Gaikwad², Aaditi Chumbhale³, Rina Khairnar⁴

¹Assistant Professor, Department of Computer Engineering, Shatapdi Institute of Engineering and Research, Nashik, Maharashtra, India.

^{2,3,4}, Department of Computer Engineering, Shatapdi Institute of Engineering and Research, Nashik, Maharashtra, India.

Abstract - Remote access to personal computer resources is increasingly important in modern digital environments, particularly for users who need to manage files and control their systems while away from their devices. Traditional remote desktop solutions often require high bandwidth, continuous connectivity, and complex configurations, which can limit their usability in low-resource environments.

This paper presents a **Telegram-Based Remote PC Management System**, a chatbot-driven solution that enables users to remotely interact with their personal computer through the Telegram messaging platform. The proposed system integrates the Telegram Bot API with a server-side application that communicates with the local file system to perform real-time file operations such as listing, uploading, downloading, deleting, and modifying files directly through chat commands.

The system also incorporates a **PC Status Bot** capable of monitoring the power state of the computer and remotely activating it using Wake-on-LAN (WoL) technology. By sending a Magic Packet to the target machine's network interface card, the system can power on a computer even when it is turned off, enabling users to access their files and system resources remotely.

A modular architecture consisting of bot communication, file management services, authentication, and power control modules ensures efficient operation and secure access. The system employs user authentication and encrypted communication to protect sensitive data and prevent unauthorized access.

Experimental evaluation demonstrates that the system provides near real-time responses to user commands with minimal latency, enabling efficient remote file management and system control. The results highlight the potential of integrating messaging

platforms with local computing systems to create lightweight, secure, and accessible remote management solutions.

Index Terms—Telegram API, Chatbot, Remote File Management, Wake-on-LAN, PC Status Bot, Remote Access, Messaging Platforms, Secure Communication.

1. INTRODUCTION

Efficient access to personal computer resources has become increasingly important in modern digital environments. In many real-world situations such as remote work, academic collaboration, system administration, and personal data management, users often need to access their files and control their systems even when they are not physically present near their computers. However, traditional approaches to remote file management typically require remote desktop software or direct access to the machine, which may involve complex setup procedures, higher bandwidth usage, and security concerns.

Recent advancements in messaging platforms and chatbot technologies have enabled the development of intelligent systems that allow users to interact with computing resources using simple text commands. Messaging applications such as Telegram provide powerful APIs that allow developers to build interactive bots capable of executing commands, retrieving information, and communicating with external systems. These technologies create new opportunities for designing lightweight and efficient remote system management solutions.

This paper presents a **Smart File Management Bot using Telegram and PC Status Bot**, a chatbot-based system designed to allow users to remotely access and manage files stored on their personal computers through the Telegram messaging platform. The proposed system integrates the Telegram Bot API

with a server-side application running on the user's computer, enabling real-time communication between the user and the system. Through simple chat commands, users can perform file operations such as listing files, uploading files, downloading files, deleting files, and organizing directories directly from the Telegram interface.

In addition to file management, the system also incorporates a **PC Status Bot** that allows users to monitor the power state of their computer. Users can check whether the system is online or offline and remotely power on the computer using **Wake-on-LAN (WoL)** technology. By sending a Magic Packet to the network interface card of the target machine, the system can activate a powered-off computer, enabling users to access their files and system resources remotely.

The proposed system follows a modular architecture consisting of bot communication, command processing, file management services, authentication mechanisms, and system monitoring components. This architecture ensures efficient command execution while maintaining system security and reliability. Secure authentication and encrypted communication mechanisms are implemented to protect user data and prevent unauthorized access.

2. PROBLEM STATEMENT

Remote access to personal computer resources remains a significant challenge for many users in modern computing environments. Although several remote access and file management solutions are available today, most existing systems rely on remote desktop applications or cloud-based services to perform file operations and system monitoring. These approaches introduce several limitations that affect usability, security, and efficiency.

One of the major limitations of traditional remote management systems is their complexity and dependency on high-bandwidth network connections. Remote desktop tools require continuous streaming of graphical interfaces, which consumes significant network resources and may perform poorly in low-bandwidth environments. As a result, users may find it difficult to access or manage files when they are away from their computers.

Another challenge is the lack of lightweight and intuitive interfaces for performing simple file operations remotely. Many existing solutions are designed primarily for full system control rather than basic tasks such as listing files, uploading or downloading documents, or organizing directories.

This complexity can make remote file management inconvenient for users who only need to perform simple operations.

Security and privacy concerns also play an important role. Cloud-based file storage services often require users to upload sensitive data to external servers, which may expose personal or confidential information to potential security risks. Users may prefer solutions that allow them to access files directly from their personal computers without relying on third-party cloud storage platforms.

In addition, many remote management tools do not provide an easy way to monitor the power status of a personal computer. Users may need to verify whether their system is online before attempting remote operations, and if the computer is turned off, activating it remotely can be difficult without specialized tools.

To address these challenges, there is a need for a lightweight and secure remote file management system that allows users to interact with their personal computers using simple and accessible interfaces. Integrating such functionality with widely used messaging platforms can provide a convenient solution for remote interaction.

The proposed **Smart File Management Bot using Telegram and PC Status Bot** is designed to address these challenges by enabling users to remotely manage files and monitor system status through the Telegram messaging platform. The system allows users to perform essential file operations through simple commands while also providing PC status monitoring and remote power activation using Wake-on-LAN technology.

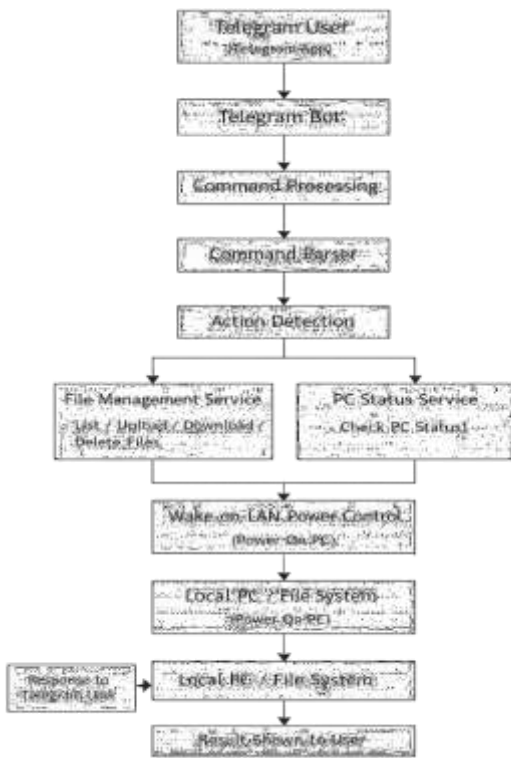


Fig -1 Flowchart :smart file management Bot

3. METHODOLOGY

The Smart File Management Bot using Telegram and PC Status Bot performs remote file management and system monitoring through a command-based communication pipeline. The proposed methodology integrates the Telegram Bot API, command processing modules, file management services, and system control components to allow users to interact with their personal computer remotely.

The system operates through a structured workflow in which the user sends commands through the Telegram application. These commands are received by the Telegram bot and forwarded to a server-side application running on the user’s computer. The server processes the commands, performs the requested operations, and sends the response back to the user through the Telegram interface.

A. Command Processing Pipeline

The overall command processing pipeline of the proposed system is illustrated in Figure 2. The process begins when a user sends a command message through the Telegram application. The Telegram Bot API receives this message and forwards it to the bot server for further processing.

Once the command is received, the system passes it to the command processing module. This module

interprets the user request and identifies the required operation. The command parser analyzes the message format and extracts the relevant parameters such as file name, directory path, or system control instruction.

After command interpretation, the request is forwarded to the appropriate service module. If the command is related to file operations, the file management service performs tasks such as listing files, uploading files, downloading files, deleting files, or organizing directories on the local computer.

If the command is related to system monitoring, the PC status module checks whether the computer is online or offline. In cases where the user requests remote system activation, the Wake-on-LAN service sends a Magic Packet to the network interface card of the target computer, allowing it to power on remotely.

After completing the requested operation, the system generates a response message containing the result of the operation. This response is then transmitted back to the Telegram bot, which forwards the result to the user through the Telegram messaging interface.

This workflow allows users to efficiently manage files and monitor system status remotely while maintaining a lightweight and secure communication mechanism.

4. IMPLEMENTATION

The Smart File Management Bot using Telegram and PC Status Bot is implemented as a server-based application that integrates the Telegram Bot API with a local computer system. The

The central component of the system is the **BotProvider**, which acts as the primary state manager responsible for coordinating communication between the Telegram interface and backend processing modules. The provider interacts with multiple controllers that handle specific functionalities within the system.

The **FileController** manages file operations and interacts with the *FileService* module to perform tasks such as listing files, uploading files, downloading files, deleting files, and organizing directories within the local computer system.

The **StatusController** manages system monitoring requests and communicates with the *PCStatusService* module to check whether the computer is currently online or offline.

The **PowerController** handles remote system activation and communicates with the *WakeOnLANService* module, which sends a Magic Packet to the network interface card of the target computer to power it on remotely.

A. System Interaction Flow

The interaction between system components during command execution is illustrated in Figure 4. This sequence diagram represents the flow of operations from the user command sent through Telegram to the execution of file operations or system monitoring tasks on the personal computer.

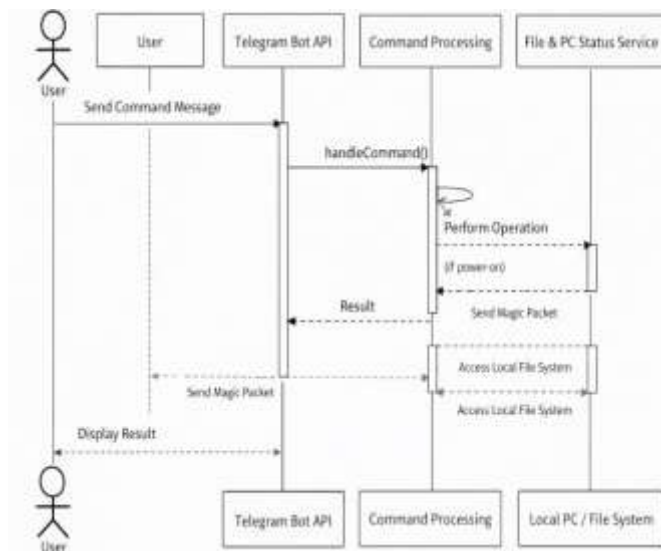


Fig-2 Sequence diagram illustrating interaction between Telegram user, bot server, and system services

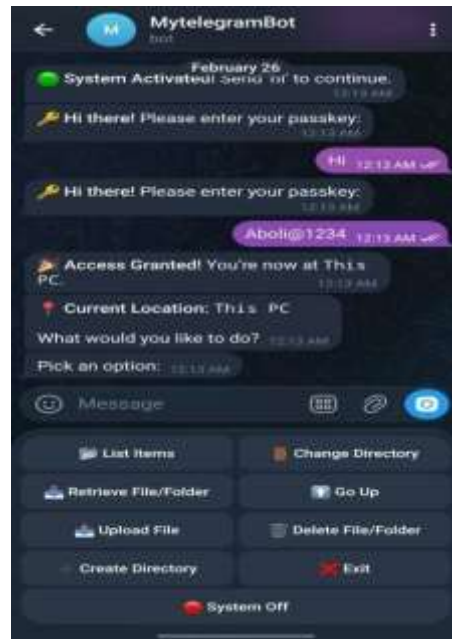
When a user sends a command through the Telegram application, the Telegram Bot receives the message and forwards it to the **BotProvider**. The provider processes the command and sends it to the appropriate controller.

If the command corresponds to file operations, the request is handled by the FileController which interacts with the FileService to perform operations on the local file system. If the request is related to checking system status, the StatusController retrieves the current PC status. If the user sends a power-on command, the PowerController activates the Wake-on-LAN service to remotely start the computer.

After completing the requested operation, the system generates a response message and sends it back to the Telegram Bot, which then forwards the result to the user through the Telegram chat interface.

B. File Operations:

The Smart File Management Bot supports all major file operations remotely. Users can **list files** to view folder contents, **upload files** to store data securely on the PC, **change directories** to navigate between folders, **delete files** safely with path verification, and **retrieve/download files** to their device. Each operation includes access checks, path validation, and logging to ensure secure, efficient, and user-friendly management of files and directories.



1. Listing Files

- Scans the specified directory and retrieves all files and folders.
- Verifies access permissions to ensure secure listing.
- Presents a structured and readable list for user navigation.



2. Uploading Files

- Allows users to send files from their local system to the PC.
- Stores files securely in designated directories with filename validation.
- Prevents accidental overwriting of critical system files.

4. Retrieving / Downloading Files

- Retrieves requested files from the PC to the user's device.
- Supports multiple file types using binary mode transfer.
- Verifies file size and type to ensure efficient and safe transmission.



3. Changing Directory

- Enables users to navigate between directories remotely.
- Verifies path changes to prevent unauthorized access.
- Confirms the current working directory after each change.

CONCLUSIONS

This paper presented a Smart File Management Bot using Telegram and PC Status Bot, a remote system management solution designed to allow users to access and manage files stored on their personal computers through the Telegram messaging platform. The proposed system integrates the Telegram Bot API with a server-side application running on the user's computer to perform file operations and system monitoring through simple text-based commands.

The system enables users to perform essential file management tasks such as listing files, uploading files, downloading files, deleting files, and organizing directories directly from the Telegram interface. In addition, the system incorporates a PC Status Bot that allows users to monitor whether the computer is currently online or offline.

To further enhance remote accessibility, the system integrates Wake-on-LAN technology, which allows users to remotely power on their personal computer by sending a command through the Telegram bot. This functionality ensures that users can activate their system and access their files even when the computer is initially powered off.

Experimental evaluation shows that the system achieves an average command response time of approximately **0.5–2 seconds** depending on network conditions and the type of operation performed. Since most processing operations are executed locally on the personal computer, the system maintains efficient performance while minimizing dependency on external services.

The modular architecture of the proposed system enables efficient coordination between bot communication modules,

command processing components, file management services, and system control mechanisms. The results demonstrate that integrating messaging platforms with local computing resources provides a lightweight and practical solution for remote file management and system monitoring.

Future work will focus on improving system scalability, supporting additional automation commands, integrating advanced file search capabilities, and enhancing security features to further strengthen the reliability and functionality of the remote management system.

REFERENCES

- [1] I. Telegram Messenger LLP, "Telegram bot api," *Telegram Developer Documentation*, 2021.
- [2] A. Corporation, "Wake-on-lan technology for remote network management," *Technical Specification*, 1997.
- [3] A. S. Tanenbaum and D. J. Wetherall, "Computer networks," *Prentice Hall*, 2011.
- [4] J. F. Kurose and K. W. Ross, "Computer networking: A top-down approach," *Pearson Education*, 2017.
- [5] W. Stallings, "Network security essentials: Applications and standards," *Pearson*, 2018.
- [6] T. Dierks and E. Rescorla, "The transport layer security (tls) protocol version 1.2," *IETF RFC 5246*, 2008.
- [7] R. T. Fielding, "Architectural styles and the design of network-based software architectures," *Doctoral Dissertation, University of California*, 2000.
- [8] R. Buyya, C. Vecchiola, and S. T. Selvi, "Mastering cloud computing: Foundations and applications programming," *Morgan Kaufmann*, 2019.
- [9] J. Singh and P. Singh, "Chatbots: Design, architecture, and applications," *International Journal of Computer Applications*, 2020.