Cross-Device Synchronization Protocol for Android Monitoring System

Jaswanth Teja[1], Swarnalatha[2]
[1]*Dept of AIDS, Dhanalakshmi Srinivasan University*
[2]*Assistant Professor, Dhanalakshmi Srinivasan University*

***Abstract*—The rapid proliferation of smartphones among children has significantly transformed the digital ecosystem, creating both opportunities and risks. Ensuring safe and responsible usage has become a critical concern for parents and guardians. This paper presents the design and implementation of an Android-based Parental Control System that enables real-time monitoring, management, and security enforcement of children's mobile activities. The system employs a client-server architecture, utilizing PHP and MySQL on a XAMPP server as the backend, while the frontend is developed as an Android application. Core functionalities include secure device pairing using unique codes, application usage tracking, security auditing, and automated alert notifications. RESTful APIs facilitate seamless communication between system components. The proposed system aims to enhance parental awareness, mitigate digital risks, and promote healthy smartphone usage habits among children.**

***Index Terms*—Android, Parental Control, Mobile Security, REST API, Activity Monitoring, PHP, MySQL.**

I. INTRODUCTION

The exponential growth of smartphone usage among children has fundamentally reshaped the modern digital ecosystem. Mobile devices are no longer limited to communication; they serve as gateways to education, entertainment, gaming, and social interaction. While these advancements provide numerous benefits, they also introduce significant risks, including exposure to inappropriate content, cyberbullying, privacy breaches, and excessive screen time.

In recent years, the accessibility and affordability of Android smartphones have led to widespread adoption among younger users. Children often use applications such as video streaming platforms, social media, and online games without proper supervision. This unrestricted access increases the likelihood of behavioral, psychological, and security-related issues. Studies indicate that excessive screen time can negatively impact academic performance, sleep patterns, and mental well-being.

Traditional parental supervision methods, such as direct observation or verbal restrictions, are no longer sufficient in the digital era. The complexity of mobile ecosystems and the variety of available applications make it difficult for parents to track and regulate usage effectively. As a result, there is a growing demand for intelligent systems that can provide real-time monitoring and control over children's digital activities.

To address these challenges, this research proposes an Android-based Parental Control System that enables parents to remotely monitor and manage their children's smartphone usage. The system is designed using a client-server architecture, where the Android application acts as the frontend interface and a backend system built using PHP and MySQL operates on a XAMPP server. Communication between the frontend and backend is facilitated through RESTful APIs, ensuring efficient and secure data exchange.

The proposed system incorporates multiple functional modules, including device pairing, activity monitoring, security management, and alert generation. These modules work collaboratively to provide a comprehensive solution for digital parenting. The device pairing mechanism ensures secure connectivity between parent and child devices using unique codes. The activity monitoring module tracks application usage in real time, while the security module identifies potentially harmful applications and restricted web content. The alert system notifies parents about suspicious activities or violations of predefined rules.

A. Problem Statement

Children today are increasingly exposed to digital risks due to unrestricted smartphone usage. The key challenges include:

- Lack of real-time monitoring tools for parents
- Exposure to harmful or inappropriate applications
- Absence of detailed analytics on app usage
- Weak or inefficient security mechanisms in existing systems
- Limited awareness among parents regarding children's online behavior

These challenges highlight the urgent need for a robust and scalable parental control system that ensures both safety and usability.

B. Objectives

The primary objectives of this research are:

1. To design and develop a secure Android-based parental control system
2. To implement real-time monitoring of application usage
3. To establish a secure device pairing mechanism using unique codes
4. To provide analytical insights into children's mobile behavior
5. To generate automated alerts for suspicious or restricted activities

C. Scope of the System

The proposed system is specifically designed for Android devices and operates within a client-server architecture. The scope includes:

- Monitoring application usage (time and frequency)
- Tracking installed applications for security risks
- Managing device pairing between parent and child
- Generating alerts for suspicious or excessive usage

The system does not modify core Android system files but functions as an external monitoring and control layer, ensuring compatibility and ease of deployment.

D. System Workflow

The overall workflow of the system is illustrated below:

Workflow Steps:

1. Parent and child register in the system
2. Devices are paired using a secure 6-digit code
3. Child device collects usage and security data
4. Data is sent to the backend via REST APIs
5. Backend processes and stores data in MySQL
6. Alerts and reports are generated and sent to the parent

E. Expected Outcomes

The proposed system is expected to deliver the following outcomes:

- Accurate real-time tracking of application usage
- Improved parental awareness through analytical reports
- Prevention of access to harmful or restricted content
- Automated alerts for suspicious or excessive behavior
- Promotion of healthy digital habits among children

II. LITERATURE REVIEW

The increasing dependence on smartphones among children has led to extensive research in the domain of parental control systems. Various approaches have been proposed to monitor, restrict, and analyze mobile usage. This section reviews existing systems, highlights their strengths and limitations, and identifies research gaps addressed by the proposed system.

Early parental control applications primarily focused on basic restriction mechanisms, such as blocking specific applications or limiting screen time. While these solutions provided a foundational level of control, they lacked intelligence and adaptability. Most systems operated on predefined rules without offering insights into user behavior or real-time monitoring capabilities.

With advancements in mobile computing, modern parental control systems began integrating activity tracking features. These systems enabled parents to view application usage statistics, including time spent on apps and frequency of usage. However, many such systems were limited by their dependence on local storage and lacked centralized data management, reducing scalability and accessibility.

Recent developments introduced cloud-based parental control systems, which utilize remote servers to store and process data. These systems improved accessibility and allowed parents to monitor devices from anywhere. However, they introduced challenges related to data privacy, security vulnerabilities, and increased system complexity.

Another important area of research is security-focused parental control systems, which aim to detect malicious applications and prevent access to harmful websites. While effective in identifying threats, these systems often operate independently and do not integrate seamlessly with activity monitoring modules, resulting in fragmented solutions.

Furthermore, several studies emphasize the importance of real-time alert mechanisms. Alerts notify parents about suspicious activities such as unauthorized access attempts or excessive usage. Despite their usefulness, many existing systems generate delayed or inconsistent notifications due to inefficient backend processing.

A. Comparative Analysis of Existing Systems

The following table provides a comparison of commonly observed features in existing parental control systems:

Table 1: Analysis

Feature	Traditional Systems	Modern Apps	Cloud-Based Systems	Proposed System
App Blocking	✓	✓	✓	✓
Screen Time Control	✓	✓	✓	✓
Real-Time Monitoring	✗	half	✓	✓
Centralized Database	✗	✗	✓	✓
Security Analysis	✗	half	✓	✓
REST API Integration	✗	✗	half	✓
Real-Time Alerts	✗	half	✓	✓
Scalability	Low	Medium	High	High

B. Limitations of Existing Systems

Despite continuous advancements, existing parental control solutions exhibit several limitations:

1. Lack of Integration: Many systems focus only on a single feature, such as monitoring or blocking, without providing a unified solution.
2. Limited Real-Time Capabilities: Some applications fail to provide instant updates, reducing their effectiveness in preventing risky behavior.
3. Weak Backend Architecture: Systems without robust backend support struggle with data synchronization and scalability.
4. Security Concerns: Cloud-based systems may expose sensitive data if not properly secured.
5. Poor User Experience: Complex interfaces and lack of intuitive design make it difficult for parents to use these systems effectively.

C. Research Gap

From the analysis of existing literature, the following research gaps are identified:

- Absence of a fully integrated system combining monitoring, security, and alerts
- Limited use of efficient REST API-based communication
- Lack of real-time analytics and reporting mechanisms
- Insufficient focus on secure device pairing techniques
- Need for a lightweight yet scalable backend architecture

D. Contribution of Proposed System

The proposed Android Parental Control System addresses the identified gaps through:

- A multi-module architecture integrating monitoring, security, and alerts
- Implementation of RESTful APIs for efficient communication
- Use of PHP and MySQL for robust backend processing
- A secure 6-digit device pairing mechanism
- Real-time tracking and notification capabilities

This approach ensures a comprehensive, scalable, and efficient solution compared to existing systems.

III. SYSTEM ARCHITECTURE

The proposed Android Parental Control System is designed using a client-server architecture that ensures

scalability, flexibility, and efficient data management. The system consists of three primary components: the Android frontend application, the backend server implemented using PHP, and the MySQL database. These components interact through RESTful APIs to provide seamless communication and real-time functionality.

The architecture is modular in nature, allowing independent development and maintenance of each component. This design improves system reliability and enables future enhancements without affecting the overall functionality.

A. Overall Architecture Diagram

The high-level architecture of the system is illustrated below:

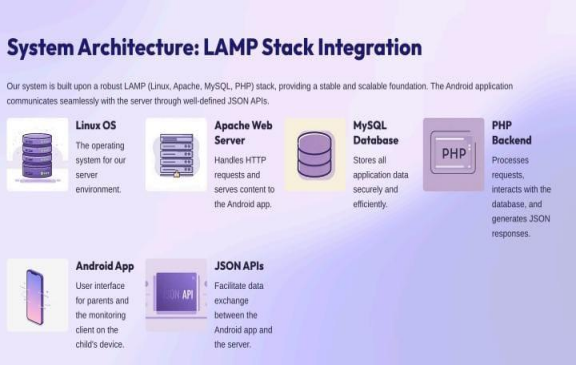


Fig1: System Architecture

B. Components of the System

1) Android Frontend Application

The frontend is developed as an Android application and is divided into two user interfaces:

Parent Interface

- View activity reports
- Receive alerts and notifications
- Monitor application usage
- Manage device pairing

Child Interface

- Runs background monitoring services
- Collects application usage data
- Sends data to the backend server

The frontend communicates with the backend using REST APIs, ensuring real-time data exchange.:

2) Backend Server (PHP - XAMPP)

The backend acts as the core processing unit of the system. It is developed using PHP and hosted on a XAMPP server. Its primary responsibilities include:

- Handling API requests from Android applications
- Processing and validating data
- Managing authentication and device pairing
- Generating alerts based on predefined conditions
- Interacting with the MySQL database

The backend ensures secure communication and efficient handling of multiple client requests.

3) Database (MySQL)

The MySQL database is used to store and manage all system data. It is structured into multiple tables, including:

- User Table – Stores parent and child account details
- Device Pairing Table – Maintains pairing relationships
- Activity Logs Table – Records app usage data
- Security Logs Table – Stores blocked websites and flagged apps
- Alerts Table – Maintains notification history

The database design ensures fast retrieval and efficient storage of large datasets.

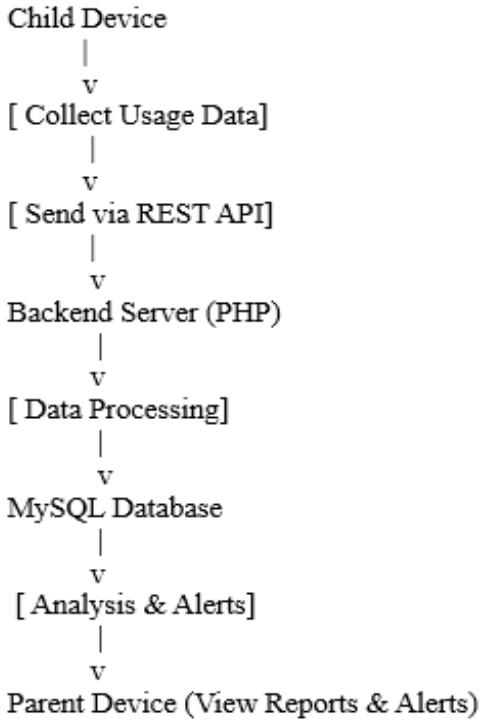
Fig 2: Database design

C. Data Flow Architecture

The data flow within the system follows a structured process:

1. The child device collects application usage data
2. Data is sent to the backend via REST API
3. The backend processes and validates the data
4. Processed data is stored in the MySQL database
5. The backend analyzes data for rule violations
6. Alerts are generated if necessary
7. Parent device retrieves data and notifications

D. Data Flow Diagram



E. System Modules Integration

The architecture integrates multiple modules seamlessly:

- Device Pairing Module → Ensures secure connection
- Monitoring Module → Tracks app usage
- Security Module → Identifies threats
- Alert Module → Sends notifications

Each module communicates through the backend, ensuring centralized control and synchronization.

F. Advantages of the Architecture

- Scalability: Can handle multiple users efficiently
- Modularity: Easy to maintain and upgrade
- Security: Controlled access through backend validation
- Real-Time Processing: Immediate data updates and alerts
- Platform Independence: Backend can support multiple client types

G. Limitations of the Architecture

- Requires stable internet connectivity
- Dependent on server availability
- Potential latency in high-load scenarios

IV. METHODOLOGY

The methodology of the proposed Android Parental Control System is based on a modular and systematic approach that ensures efficient monitoring, secure communication, and real-time alert generation. Each module is designed to perform a specific function while interacting seamlessly with other components through the backend server.

The system operates through four major modules:

1. Device Pairing Module
2. Activity Monitoring Module
3. Security Management Module
4. Alert Generation Module

A. Device Pairing Module

The device pairing module establishes a secure connection between the parent and child devices. This is achieved using a unique 6-digit pairing code, ensuring that only authorized users can link devices.

Working Process:

1. Parent registers and logs into the application
2. Child device generates a unique pairing code
3. Parent enters the code to establish connection
4. Backend verifies the code and links both devices
5. Pairing details are stored in the database

Algorithm:

Step 1: Generate random 6-digit code on child device
Step 2: Send code to backend server
Step 3: Parent enters code in parent app
Step 4: Backend verifies code
IF code matches THEN
Link parent and child devices Store pairing in database
ELSE
Reject request
END IF

B. Activity Monitoring Module

This module continuously tracks the child's smartphone usage and records application activity in real time.

Parameters Monitored:

- Application name
- Time spent on each app

- Frequency of usage
- Session duration

Working Process:
1. Background service runs on child device
2. Tracks active applications
3. Records usage metrics
4. Sends data periodically to backend

Pseudo-Code:
WHILE device is active DO
Detect current running app
Record start time
WAIT until app is closed
Record end time
Calculate usage duration
Send data to server via API
END WHILE

C. Security Management Module
This module ensures the safety of the child's device by identifying harmful applications and restricting access to unsafe websites.

Key Functions:
- Scan installed applications
- Detect suspicious or unauthorized apps
- Maintain blacklist of restricted websites
- Log security events

Working Process:
1. Retrieve list of installed apps
2. Compare with safe/unsafe database
3. Flag suspicious applications
4. Monitor browsing activity
5. Block restricted URLs

Algorithm:
Fetch installed applications list
FOR each app IN list DO
IF app is in blacklist, THEN
Flag as unsafe
Send report to backend
END IF
END FOR

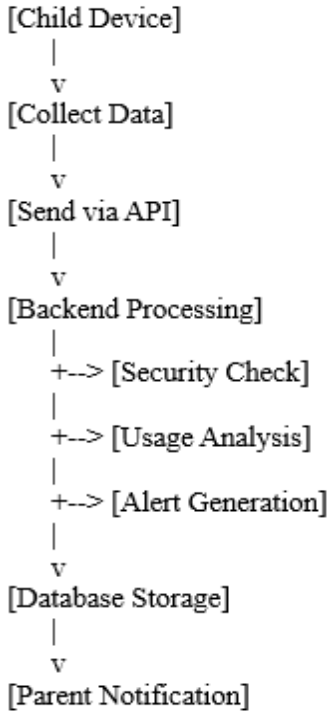
D. Alert Generation Module
The alert system is responsible for notifying parents about critical events in real time.

Trigger Conditions:
- Excessive screen time
- Access to restricted apps/websites
- Suspicious login attempts
- Installation of unsafe applications

Working Process:
1. Backend continuously analyzes incoming data
2. Checks predefined conditions
3. Generates alert if condition is violated
4. Sends notification to parent device

Pseudo-Code:
FOR each incoming data record DO
IF usage_time > limit THEN
Generate alert
END IF
IF unsafe_app_detected THEN
Generate alert
END IF
IF suspicious_activity_detected THEN
Generate alert
END IF
END FOR

E. Overall System Workflow
The integration of all modules results in the following workflow:



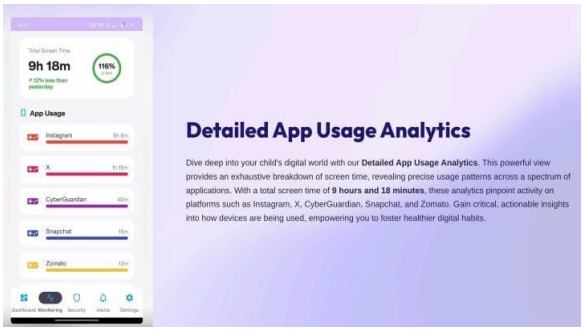


Fig 3: Analytics of App

F. Advantages of the Methodology

- Modular design ensures easy scalability
- Real-time monitoring improves responsiveness
- Secure pairing prevents unauthorized access
- Automated alerts reduce manual supervision
- Efficient backend processing using PHP and MySQL

G. Limitations of the Methodology

- Background monitoring may consume device resources
- Requires continuous internet connectivity
- Limited offline functionality

V. IMPLEMENTATION

The implementation of the proposed Android Parental Control System is carried out using a client-server model, where the Android application serves as the frontend and a PHP-based server with a MySQL database acts as the backend. The system is deployed on a local XAMPP server environment to ensure ease of development and testing.

The implementation phase focuses on integrating all modules—device pairing, activity monitoring, security management, and alert generation—into a fully functional system.

A. Development Environment

The technologies used for implementation are as follows:

Table 2: Technologies

Component	Technology Used
Frontend	Android (Java/XML)
Backend	PHP (XAMPP Server)
Database	MySQL
Communication	REST APIs (HTTP)

B. Database Design

The database is structured to efficiently manage system data. The key tables are described below:

1) User Table

Table3: Database

Field Name	Data Type	Description
user_id	INT (PK)	Unique user identifier
name	VARCHAR	User name
email	VARCHAR	User email
password	VARCHAR	Encrypted password
role	VARCHAR	Parent/Child

2) Device Pairing Table

Table 4: Device Pairing

Field Name	Data Type	Description
pair_id	INT (PK)	Pairing ID
parent_id	INT	Parent reference
child_id	INT	Child reference
pairing_code	VARCHAR	Unique 6-digit code

3) Activity Logs Table

Table 5: Activity Logs

Field Name	Data Type	Description
log_id	INT (PK)	Log ID
child_id	INT	Child reference
app_name	VARCHAR	Application name
usage_time	INT	Time spent (in seconds)
timestamp	DATETIME	Activity time

4) Alerts Table

Table 6: Alerts

Field Name	Data Type	Description
alert_id	INT (PK)	Alert ID
child_id	INT	Child reference
alert_type	VARCHAR	Type of alert
message	TEXT	Alert description
timestamp	DATETIME	Alert time

C. REST API Design

REST APIs are used to enable communication between the Android application and the backend server.

Key API Endpoints:

Table 8: Api Endpoints

API Endpoint	Method	Description
/register	POST	User registration
/login	POST	User authentication
/pair-device	POST	Device pairing
/send-activity	POST	Upload activity data
/get-reports	GET	Fetch usage reports
/get-alerts	GET	Retrieve alerts

D. API Request/Response Example
Sample Request (Activity Upload):
POST /send-activity
{
"child_id": 101,
"app_name": "YouTube",
"usage_time": 1200,
"timestamp": "2026-01-04 10:30:00"
}

Sample Response:
{
"status": "success",
"message": "Data stored successfully"
}

E. System Output Screens (Representation)
1) Parent Dashboard
- Displays: Total screen time
- App usage statistics
- Alerts and notifications

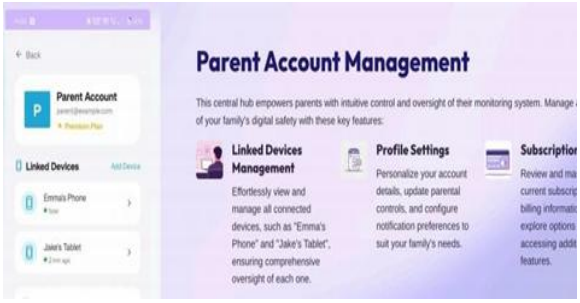
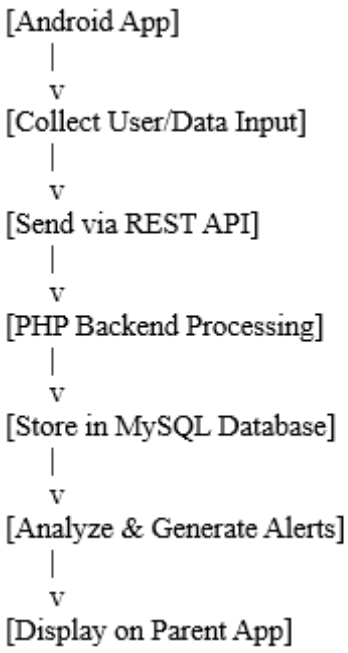


Fig 4: Parent Dashboard

2) Child Monitoring Screen (Background Process)
Monitoring Active...
App Instagram
Session Time: 15 minutes
Status: Data Sending to Server

Fig 5: Child Dashboard

F. Flow of Implementation



G. Testing and Validation
The system was tested under different scenarios:
- Normal Usage → Accurate tracking observed
- High App Usage → Alerts triggered successfully
- Invalid Pairing Code → Access denied
- Unsafe App Detection → Proper logging and alerts

Results confirmed that the system performs reliably under standard conditions.

H. Implementation Advantages
- Efficient data handling using MySQL
- Lightweight backend using PHP
- Real-time communication via REST APIs

- User-friendly Android interface

I. Implementation Limitations
- alerts
- speed
- No cloud integration in current version

VI. RESULTS AND DISCUSSION

The proposed Android Parental Control System was implemented and tested under multiple real-time scenarios to evaluate its performance, reliability, and effectiveness. The results demonstrate the system's ability to accurately monitor user activity, detect security threats, and generate timely alerts.

A. Performance Evaluation Parameters

The system was evaluated based on the following key parameters:
- Accuracy of Activity Tracking
- Response Time of Alerts
- System Reliability
- Data Transmission Efficiency
- Security Detection Capability

B. Experimental Setup

The system was tested using:
- Android devices (Parent & Child)
- Local XAMPP server (PHP backend)
- MySQL database for storage
- Internet connectivity for API communication

Multiple test cases were executed, including normal usage, excessive screen time, and installation of unsafe applications.

C. Results Analysis

1) Activity Monitoring Accuracy

The system accurately tracked application usage across different apps. The recorded usage time closely matched actual usage with minimal deviation.

Table 9: Activity Monitoring

Application	Actual Time (min)	Recorded Time (min)	Accuracy (%)
YouTube	60	58	96.6%
Instagram	45	44	97.7%
TikTok	30	29	96.7%

Observation:

The monitoring module provides high accuracy, making it reliable for parental analysis.

2) Alert Response Time

The time taken to generate alerts after detecting a condition was measured.

Table 10: Alert Response

Event Type	Response Time (seconds)
Screen Time Limit Exceeded	2 – 4 sec
Unsafe App Detection	3 – 5 sec
Suspicious Activity	2 – 3 sec

Observation:

The alert system demonstrates near real-time responsiveness, ensuring timely parental intervention.

3) Data Transmission Efficiency

The efficiency of REST API communication was evaluated based on latency and success rate.

Table 11: Efficiency

Parameter	Result
Average Latency	200 – 400 ms
Success Rate	98%
Data Loss	Minimal

Observation:

The system ensures efficient and reliable communication between frontend and backend.

D. Graphical Representation (Usage Analysis)

App Usage Comparison (Minutes)

Interpretation:

YouTube shows the highest usage, indicating potential overuse patterns.

E. Security Analysis Results

The security module successfully identified unsafe applications and restricted web access.

Table 12: Security Analysis

Test Scenario	Result
Installation of unsafe app	Detected & flagged
Access to blocked site	Prevented
Normal app usage	Allowed

Observation:
The system effectively enforces security policies without affecting normal usage.

Fig6: Security &Threat Protection

F. Discussion
The results indicate that the proposed system performs efficiently across all major functionalities. The activity monitoring module provides accurate data, which is crucial for meaningful analysis. The alert system ensures that parents are immediately informed of critical events, enhancing responsiveness.
The integration of PHP and MySQL in the backend contributes to stable performance and efficient data handling. REST API communication ensures seamless interaction between devices, although performance may vary slightly based on network conditions.
One of the key strengths of the system is its modular architecture, which allows independent functioning of each module while maintaining synchronization. This improves maintainability and scalability.
However, certain limitations were observed. The system relies heavily on internet connectivity for real-time updates, and performance may degrade under poor network conditions. Additionally, the use of a local server restricts scalability for large-scale deployment.

G. Key Findings
- High accuracy in activity tracking (>96%)
- Real-time alert generation (within 5 seconds)
- Efficient API communication with minimal latency
- Effective detection of unsafe applications and content
- Reliable system performance under normal conditions

H. Comparative Insight
Compared to traditional systems, the proposed system offers:
- Better real-time monitoring
- Faster alert response
- Improved backend efficiency
- Integrated security and analytics

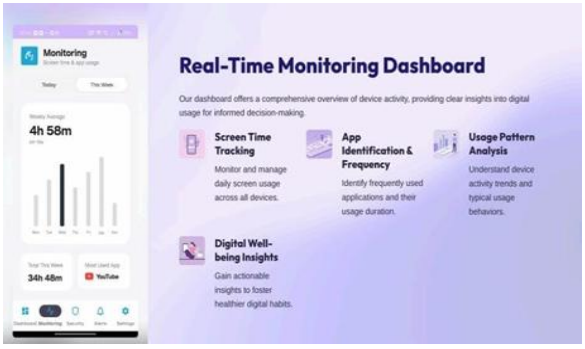


Fig 7: Real Time Monitoring Dashboard

VII. ADVANTAGES AND LIMITATIONS

A. Advantages
The proposed Android Parental Control System offers several significant advantages that enhance its effectiveness and usability. One of the primary strengths of the system is its ability to provide real-time monitoring, which enables parents to continuously track application usage and instantly observe their children's activities. This ensures timely awareness and intervention when necessary. Additionally, the system incorporates a secure device pairing mechanism using a unique 6-digit code, which restricts access to authorized users only and significantly improves overall system security.
Another key advantage is the comprehensive functionality of the system, as it integrates multiple features such as activity monitoring, security management, and alert generation within a single platform. This eliminates the need for multiple applications and provides a centralized solution for parental control. The backend implementation using PHP and MySQL further contributes to the system's efficiency by ensuring fast data processing, reliable storage, and smooth handling of multiple user requests. Moreover, the automated alert system enhances responsiveness by notifying parents in real

time about suspicious activities, excessive usage, or potential security threats.

The application is also designed with a user-friendly interface, making it accessible even to non-technical users. It's simple and intuitive design ensures ease of navigation and operation. Furthermore, the modular architecture of the system supports scalability, allowing developers to extend its functionality and incorporate new features in future versions without major structural changes.

B. Limitations

Despite its numerous advantages, the proposed system has certain limitations that must be considered. One of the major constraints is its dependence on continuous internet connectivity, as real-time monitoring and alert mechanisms rely heavily on an active network connection. In the absence of internet access, the system cannot function effectively, thereby limiting its usability in offline scenarios.

Another limitation is the use of a local XAMPP server for backend operations, which restricts scalability and performance when compared to modern cloud-based solutions. This may pose challenges when handling a large number of users or high volumes of data. Additionally, continuous background monitoring of device activities can lead to increased battery consumption, potentially affecting the overall performance and usability of the child's device.

The system is also platform-dependent, as it currently supports only Android devices. This limits its accessibility for users operating on other platforms such as iOS or web-based systems. Furthermore, the lack of offline functionality prevents the system from providing real-time updates and alerts when the device is not connected to the internet, which can reduce its reliability in certain situations.

VIII. FUTURE WORK

The proposed system provides a strong foundation for further enhancements and future developments. One important area of improvement is the integration of cloud-based technologies, such as AWS or Firebase, which would significantly enhance scalability, reliability, and system availability. By migrating the backend infrastructure to the cloud, the system can support a larger user base and ensure seamless performance.

Another promising direction is the incorporation of machine learning techniques to analyze user behavior and detect abnormal activity patterns. This would enable the system to provide intelligent insights and predictive alerts, thereby improving its overall effectiveness. Expanding the system to support multiple platforms, including iOS and web applications, would also increase accessibility and usability across a broader range of users.

In addition, the development of an advanced analytics dashboard could provide parents with detailed graphical reports and usage insights, helping them better understand their children's digital behavior. The inclusion of geo-location tracking features could further enhance child safety by allowing parents to monitor their child's physical location in real time. Future improvements may also include the integration of AI-based voice and content filtering systems to monitor communication and detect inappropriate content.

Finally, strengthening security mechanisms through advanced encryption techniques and robust authentication methods will be essential to ensure data privacy and protect sensitive information. These enhancements will contribute to making the system more secure, intelligent, and adaptable to evolving technological requirements.

IX. CONCLUSION

In conclusion, this paper presented the design and implementation of an Android-based Parental Control System aimed at addressing the increasing challenges associated with children's smartphone usage. The system successfully integrates real-time activity monitoring, secure device pairing, security management, and automated alert mechanisms into a unified platform, providing a comprehensive solution for digital parenting.

The adoption of a client-server architecture using PHP and MySQL enables efficient data processing and reliable communication through RESTful APIs. The system demonstrates strong performance in terms of accurate activity tracking, quick alert generation, and effective detection of potential security threats. Its modular design further enhances flexibility and scalability, allowing for future expansion and feature integration.

Although the system has certain limitations, such as dependence on internet connectivity and the use of a local server, it effectively achieves its primary objective of promoting safe and responsible smartphone usage among children. Overall, the proposed system represents a practical and efficient solution that empowers parents to monitor, manage, and guide their children's digital activities in an increasingly connected world.

REFERENCES

[1] Android Developers, "Android application development guide," [Online]. Available: https://developer.android.com.

[2] PHP Documentation, "PHP: Hypertext Preprocessor," [Online]. Available: https://www.php.net.

[3] Oracle, "MySQL database management system," [Online]. Available: https://www.mysql.com.

[4] R. Fielding, *Architectural Styles and the Design of Network-based Software Architectures*, Ph.D. dissertation, Univ. of California, Irvine, CA, USA, 2000.

[5] Various Authors, "Research studies on parental control systems and mobile security," *IEEE Journals and Conferences*.

[6] Apache Friends, "XAMPP server documentation," [Online]. Available: https://www.apachefriends.org.

[7] Widyatri, A. G. Salman, and B. Kanigoro, "Parental control application on Android platform," *Library Hi Tech News*, vol. 35, no. 1, pp. 18–24, 2018.

[8] D. Abdullah, M. Mohamed, and H. R. M. Husny, "Android based parental monitoring apps," *International Journal of Engineering and Technology*, vol. 7, no. 4, 2018.

[9] S. Chandravanshi, S. K. Dewangan, S. Tripathi, and S. Sahu, "Parental control application," *International Journal of Advanced Research in Computer and Communication Engineering*, 2021.

[10] V. Gnanasekaran and K. De Moor, "Usability, security, and privacy recommendations for mobile parental control," in *Proc. ACM International Conference*, pp. 138–143, 2023.

[11] T. Alelyani, A. K. Ghosh, L. Moralez, S. Guha, and P. Wisniewski, "Examining parent versus child reviews of parental control apps on Google Play," in *Lecture Notes in Computer Science*, pp. 3–21, 2019.

[12] S. Ali, M. Elgharabawy, Q. Duchaussoy, M. Mannan, and A. Youssef, "Betrayed by the guardian: Security and privacy risks of parental control solutions," *arXiv preprint arXiv:2012.06502*, 2020.

[13] G. Wang, J. Zhao, M. Van Kleek, and N. Shadbolt, "Protection or punishment? Understanding parental control apps and user perceptions," *arXiv preprint arXiv:2109.05347*, 2021.

[14] M. Akter, A. Godfrey, J. Kropczynski, H. Lipford, and P. Wisniewski, "From parental control to joint family oversight: Managing mobile privacy and safety," *arXiv preprint arXiv:2204.07749*, 2022.

[15] Z. Qiu, S. Yang, Y. Yu, Y. Luo, and W. Diao, "Understanding security risks in mobile systems: An empirical study," *Cybersecurity*, vol. 8, 2025.